POLYTOPE-BASED TOPOLOGY OPTIMIZATION USING A MIMETIC-INSPIRED METHOD

BY

ARUN L. GAIN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Professor Glaucio H. Paulino, Chair and Director of Research
Doctor Joseph Bishop, Sandia National Laboratories
Professor C. Armando Duarte
Professor Ahmed E. Elbanna
Professor Harrison M. Kim
Professor Emílio C. N. Silva, University of São Paulo
Professor Alok Sutradhar, Ohio State University

Abstract

Topology optimization refers to the optimum distribution of materials, so as to achieve certain prescribed design objectives while simultaneously satisfying constraints. Engineering applications often require unstructured meshes to capture the domain and boundary conditions accurately and to ensure reliable solutions. Hence, unstructured polyhedral elements are becoming increasingly popular. Since the pioneering work of Wachspress, many interpolants for polytopes have come forth; such as, mean value coordinates, natural neighbor-based coordinates, metric coordinate method and maximum entropy shape functions. The extension of the shape functions to three-dimensions, however, has been relatively slow partly due to the fact that these interpolants are subject to restrictions on the topology of admissible elements (e.g., convexity, maximum valence count) and can be sensitive to geometric degeneracies. More importantly, calculating these functions and their gradients are in general computationally expensive. Numerical evaluation of weak form integrals with sufficient accuracy poses yet another challenge due to the non-polynomial nature of these functions as well as the arbitrary domain of integration. Virtual Element Method (VEM), which has evolved from Mimetic Finite Difference methods, addresses both the issues of accuracy and efficiency. In this work, a VEM framework for three-dimensional elasticity is presented. Even though VEM is a conforming Galerkin formulation, it differs from tradition finite element methods in the fact that it does not require explicit computation of approximation spaces. In VEM, the deformation states of an element are kinematically decomposed into rigid body, linear and higher order modes. The discrete bilinear form is constructed to capture the linear deformations exactly which ensures that the displacement patch test is passed and optimum convergence is achieved. The present work focuses on first-order VEM with degrees of freedom associated with the vertices of the elements. Construction of the stiffness matrix reduces to the evaluation of surface integrals, in contrast to the volume integrals encountered in the conventional finite element method (FEM), thus reducing the overall computational cost.

By means of the aforementioned approach, a framework for three-dimensional topology optimization is developed for polyhedral meshes. In the literature, topology optimization problems are typically solved with either tetrahedral or brick meshes. Numerical anomalies, such as checkerboard patterns and one-node connections, are present in such formulations.

Constraints in the geometrical features of spatial discretization can also result in mesh dependent sub-optimal designs. In the current work, polyhedral meshes are proposed as a means to address the geometric features of the domain discretization. Polyhedral meshes not only provide greater flexibility in discretizing complicated domains but also alleviate the aforementioned numerical anomalies. For topology optimization problems, many approaches are available; which can mainly be classified as density-based methods and differential equation-driven methods (further subclassified as level-set and phase-field methods). Before choosing density-based methods for polyhedral topology optimization, a couple of differential equation-driven methods; which are representative of the literature, are exhaustively analyzed in two-dimensions. Finally, we also investigate aesthetics in topology optimization designs. In this work, two-dimensional topology optimization on tessellations is investigated as a means to coalesce art and engineering. M.C. Escher's tessellations using recognizable figures are mainly utilized. The aforementioned Mimetic Finite Difference-inspired approach (VEM) facilitates accurate numerical analysis on any non self-intersecting closed polygons such as tessellations.

To my parents

Acknowledgments

I would like to take this opportunity to acknowledge my Professors, colleagues, family and friends who have made this work possible. First and foremost, I would like to express my gratitude to my advisor, Professor Glaucio H. Paulino for providing me the opportunity to work with him on intellectually challenging topics. I am grateful for his continued guidance and support and for providing me with multitude of resources for my overall professional development.

Many thanks to Dr. Joseph Bishop, Professors Armando Duarte, Ahmed Elbanna, Harrison Kim, Emilio Silva and Alok Sutradhar for taking the time out of their busy schedule to participate in my PhD defense committee and providing me valuable feedback to improve my work. I would specially like to thank Prof. Duarte for his encouragement over the years and for having me as his teaching assistant for the courses - Structural Mechanics and Structural Analysis, which was a great learning experience. I am grateful to Cameron Talischi, Prof. Ivan Menezes, Leonardo Duarte for collaborating with me and for all the intellectually stimulating discussions that we had during my PhD.

I am thankful to my colleagues Kyoungsoo Park, Bin Shen, Tam Nguyen, Lauren Beghini, Tomas Zegard, Sofie Leon, Daniel Spring, Junho Chun, Evgueni Filipov, Heng Chi, Maryam Eidini, William Colletti, Adeildo Soares Ramos Jr., Ludimar Lima de Aguiar and Marco Alfano for their support and friendship. Special thanks to my friends Avinash Voruganti, Pablo Caiza, Jason Patrick, Rodrigo Espinha, Herisangela Espinha, Daiane Brisotto, Ying Yu and Piyush Chaunsali for their camaraderie and encouragement.

I gratefully acknowledge the financial support of the FMC Educational Fund Fellowship and the National Science Foundation (NSF) through projects CMMI 1321661 and CMMI 1335160. I also acknowledge partial support from the Donald B. and Elizabeth M. Willett endowment at the University of Illinois at Urbana-Champaign.

Finally, I wish to express my gratitude to me parents for their tremendous encouragement and unwavering selfless dedication throughout my life. They have always been there, motivating and inspiring me to set the highest standards in every aspect of my life. I dedicate this dissertation to them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Typical engineering applications often require unstructured meshes to capture the domain and boundary conditions accurately and to ensure reliable solutions. Hence, unstructured polyhedral elements are becoming increasingly popular. They are especially attractive in topology optimization; which optimally distributes material in a given domain, with variable connectivity, so as to satisfy certain design objectives. Polyhedral/polygonal elements not only provide convenience and flexibility in discretizing complicated design domains but also lead to optimal designs which are not biased by the domain discretization. Many finite element interpolants on polytopes are available; such as, Wachspress, mean value coordinates, natural neighbor-based coordinates, metric coordinate method, and maximum entropy shape functions. In two-dimensions, employing the aforementioned polygonal shape functions, numerical analysis can be efficiently conducted by mapping the physical element to a reference element [200], similar to that employed for typical triangle and quad elements. However, the three-dimensional extension of the shape functions has been relatively slow partly due to the fact that these interpolants are subject to restrictions on the topology of admissible elements (e.g., convexity, maximum valence count) and can be sensitive to geometric degeneracies. More importantly, calculating these functions and their gradients are in general computationally expensive. Numerical evaluation of weak form integrals with sufficient accuracy poses yet another challenge due to the non-polynomial nature of these functions as well as the arbitrary domain of integration. All these reasons suggest that there is a need for an accurate and numerically efficient scheme for polytopes. This thesis strives to develop such an approach.

1.1 Introduction to topology optimization

Topology optimization refers to the optimum distribution of a material in a given design space, under certain specified boundary conditions, so as to satisfy prescribed performance objectives. In manufacturing industries, topology optimization is often used as a tool to obtain preliminary conceptual designs. Manufacturing constraints applied to topology optimization can yield designs which can be fine-tuned for performance and manufacturability.

Hence, the traditional costly and time consuming design iterations can be minimized while still producing equipment of superior quality. Topology optimization is used in a wide range of industries such as the automotive and equipment manufacturing industries. Topology optimization approaches can primarily be classified as either density-based methods or differential equation-driven methods. In the next two sections, a brief background of both classes of optimization methods is provided.

1.1.1 Density-based methods

Density-based methods refer to approaches where the geometry is represented in terms of two or more phases, one of which typically represents a *no material* phase. In general, the material distribution can be considered to be either constant in each element or interpolated using shape functions with the densities defined at the element nodes. Some of the early work in the field of density-based methods includes that by Bendsøe and Kikuchi [37], and Suzuki and Kikuchi [204], which use the homogenization method to determine macroscopic structure properties, such as elastic modulus, based on the microstructure configuration. Subsequently, its variants, such as the Solid Isotropic Material with Penalization (SIMP) method, were proposed [36, 172, 38]; which provided a simple approach to determine intermediate material densities. Mathematical programming algorithms; such as, Method of Moving Asymptotes (MMA) [205], Optimality Criteria (OC) [39], Sequential Linear Programming (SLP), Sequential Quadratic Programming (SQP) and CONvex LINearization approximations (CONLIN) [83] are typically used to update the designs in density-based methods.

Density-based methods have been used to explore a wide variety of problems; such as, dynamics [75, 190], multi-physics [183, 187], photonic crystal structures [113], phononic band gap materials [185], functionally graded materials [162, 194], piezoelectric actuators [56], reliability-based optimization [118, 150], large-scale optimization [230, 148, 15] and biomedical engineering applications [203]. Commercial software; such as, OptiStruct, Ansys and MSC/Nastran are based on density methods.

Evolutionary Structural Optimization (ESO) is another class of density-based methods used for topology optimization [238]. In ESO methods, certain parameters, such as Mises stress, are evaluated for each element and, at each iteration, elements with the lowest parameter value are eliminated. In Bi-directional ESO (BESO), along with elimination, new elements are added at locations adjacent to the elements with high parameter value. Primary drawbacks of ESO methods are that they are fully heuristic, they require more iterations to converge than gradient-based methods, and they have difficulty controlling the final volume of the design. Genetic Algorithms (GAs) related to Charles Darwin's survival of the fittest principle; which are non-gradient methods, have also been proposed for density-based topology optimization. In the literature, authors have often argued that such methods provide globally optimal solutions. Since the early work of Holland [107] and Goldberg [97], many GAs have been proposed for topology optimization [64, 207]. The main criticism of such methods is that they become prohibitively expensive for large scale problems.

1.1.2 Differential equation-driven methods

Recently, a new set of approaches for topology optimization have come forth in which the design is driven by differential equations. One of the differential equation driven approaches uses the level-set method [154, 179, 152], c.f. Fig. 1.1, in which the fronts and free boundaries evolve using the Hamilton-Jacobi equation. The level-set method utilizes implicit level-set functions to represent the geometry; which allows for the determination of structural boundaries. The level-set method is a computationally tractable and versatile method, which has been adapted in a variety of fields; such as, fluid mechanics [202, 63, 248, 78, 60], optics [115, 102], image processing, solids modeling, and computer animation [242, 214, 152]. In addition, the level-set method has been used to solve thermal problems [100, 249, 237, 119, 112, 142, 239].

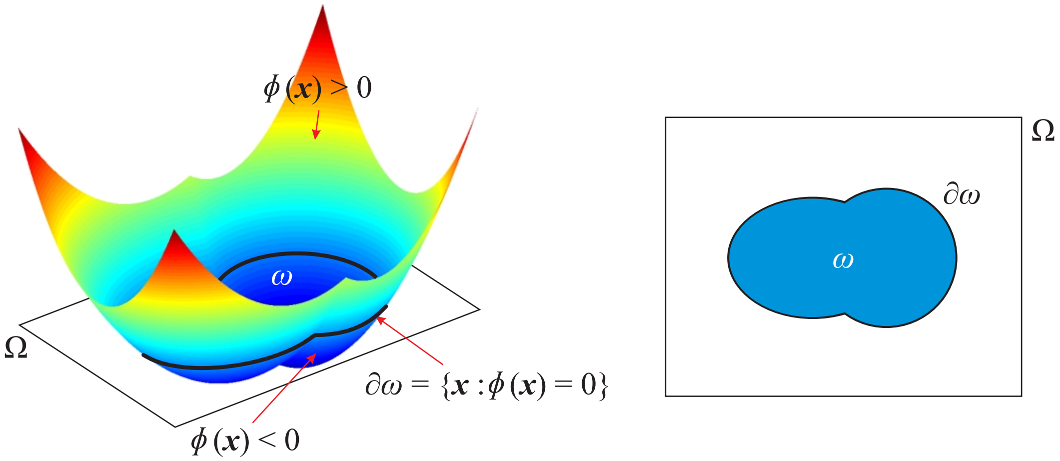


Figure 1.1: Representation of design domain Ω using implicit function (level-sets). This is an alternative approach to explicit parameterization of the geometry.

Phase-field based methods are another category of approaches in which designs evolve using specialized differential equations. Basically, the phase-field method is a diffuse interface model where boundaries between phases are not sharp, but are considered to have a finite thickness, hence providing a smooth transition for the physical quantities between the phases. It has been used in materials science to study the phase transition phenomenon. Cahn and Hillard [54], and Allen and Cahn [11] used the theory of phase transition to study liquid phases with variable densities. Phase-field methods have been applied in a wide variety of fields; such as, fracture mechanics [17], visual reconstruction [139], and crystal growth simulations [121]. In Chapter 2, four representative level-set methods and one phase-field method are critically assessed [89].

1.2 Topology optimization problem statement

Current research concentrates on optimization of linearized elastic system under small deformations subjected to surface tractions $\boldsymbol{t}$. The elasticity problem is expressed as: find $\boldsymbol{u}$ such that,

$$a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}), \qquad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}} \tag{1.1}$$

where

$$\begin{gathered} a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, d\boldsymbol{x}, \qquad \ell(\boldsymbol{v}) = \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{v} \, d\boldsymbol{s} \\ \boldsymbol{\mathcal{V}} = \left\{ \boldsymbol{v} \in H^1(\Omega)^3 : \boldsymbol{v}|_{\Gamma_u} = \mathbf{0} \right\} \end{gathered} \tag{1.2}$$

Here, $\boldsymbol{\varepsilon}(\boldsymbol{u}) = {}^1/_2(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ is the second order linearized strain tensor and $\mathbf{C}$ is the material elasticity tensor. The working domain, Ω, contains all admissible shapes ω, i.e., $\omega \subseteq \Omega$. Its boundary $\partial\Omega$ consists of three disjoint segments, $\partial\Omega = \Gamma_u \cup \Gamma_{t0} \cup \Gamma_t$. Here, Γ_u, Γ_{t0}, and Γ_t represent displacement, homogeneous traction, and non-homogeneous traction boundary conditions ($\boldsymbol{t} \neq \mathbf{0}$), respectively. Also, the design ω, with boundary $\partial\omega = \gamma_u \cup \gamma_{t0} \cup \gamma_t$, is constrained to satisfy $\gamma_u \subseteq \Gamma_u$ and $\gamma_t = \Gamma_t$. Here, γ_u, γ_{t0}, and γ_t correspond to the boundaries of ω with displacement, homogeneous traction and non-homogeneous traction boundary conditions, respectively (c.f. Fig 1.2).

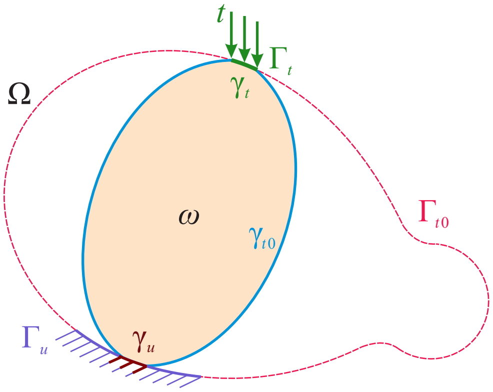


Figure 1.2: Domain description for the topology optimization problem. The boundary, $\partial\Omega$, of the working domain, Ω, consists of Γ_u (displacement boundary), Γ_{t0} (homogeneous traction boundary) and Γ_t (non-homogeneous traction boundary). The design ω, with boundary $\partial\omega = \gamma_u \cup \gamma_{t0} \cup \gamma_t$, is constrained to satisfy $\gamma_u \subseteq \Gamma_u$ and $\gamma_t = \Gamma_t$. Boundaries γ_u, γ_{t0}, and γ_t correspond to displacement, homogeneous traction, and non-homogeneous traction boundary conditions on $\partial\omega$, respectively.

The thesis focuses on two categories of problems, compliance minimization and linear compliant mechanisms. The topology optimization problem of compliance minimization refers to finding the stiffest configuration under applied loads and boundary conditions. The work done by the loads, i.e. compliance, is defined as follows:

$$J_1\left(\rho\right) = \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{u} \, d\boldsymbol{s} = \int_{\Omega} \mathbf{C}\left(\rho\right) \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) \, d\boldsymbol{x} \tag{1.3}$$

The effective elasticity tensor $\mathbf{C}$ is a function of density function, $\rho(\boldsymbol{x})$. The choice of $\mathbf{C}$ will be discussed in the subsequent chapters.

The other category of problems we study are the linear compliant mechanisms, specifically the displacement inverter and the gripper problem. The objective is to maximize the displacement in a predefined direction, u_{out}, in response to the force, f_{in}, exerted by the actuator, modeled by a spring of stiffness k_{in}. So, the quantity we aim to minimize is:

$$J_2\left(\rho\right) = -u_{out} \tag{1.4}$$

For non-trivial solutions, a volume constraint, $\int_{\Omega} \rho(\boldsymbol{x}) \, d\boldsymbol{x} \leq V_f|\Omega|$, is imposed. Thus, the combined optimization problem can be expressed as:

$$\begin{aligned} &\inf_{\rho} J_i\left(\rho\right) \qquad \text{for } i = 1 \text{ or } 2 \\ \text{subject to:} \quad &\int_{\Omega} \mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, d\boldsymbol{x} = \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{v} \, d\boldsymbol{s}, \qquad \int_{\Omega} \rho(\boldsymbol{x}) \, d\boldsymbol{x} \leq V_f|\Omega| \end{aligned} \tag{1.5}$$

where V_f is the prescribed maximum volume fraction and $|\Omega|$ is the volume of the working domain Ω.

1.3 Finite element analysis using polytopes

Finite element analysis (FEA), typically utilizes triangle/tetrahedron and quadrilateral/brick elements [70]. In the past couple of decades, polygonal elements; which are inspired from nature, have grown in stature. Carbon allotropes, salt and basalt crystals are all polygonal. Also, polygonal geometry is evident in botany (certain organic cells, chloroplasts). Beehives, pattern on the skins of tropical fish and giraffe are also inherently polygonal in shape (Fig. 1.3). Polygonal elements have been used in a wide variety of fields; such as, fracture mechanics [42, 43, 41], topology optimization [209, 211, 87], micromechanical analysis [94], computer graphics and image processing [84, 109, 108].

In topology optimization, polytopes have inherent advantages. In their work, Talischi at al. [209] have shown that mesh attributes can influence the material distribution and the orientation of members. Thus, if the mesh geometry is too restrictive, expected features, e.g. the orthogonality of members in the Michell truss problem, are eliminated from the final design. Also, the choice of lower order finite elements, such as triangles and quads, introduces

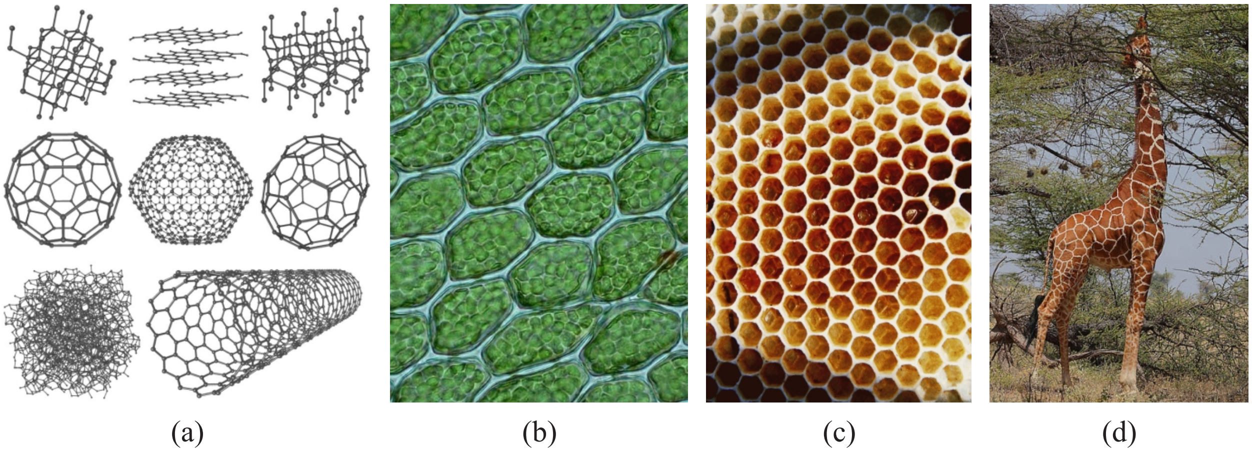


Figure 1.3: Motivation behind polygonal elements. (a) Carbon allotropes (www.chemicool. com). (b) Plagiomnium affine (http://forum.mikroscopia.com). (c) Beehive (http:// openbydesign.wpmued.org). (d) Giraffe (www.wikipedia.org).

artificial features such as single node connections and checkerboard pattern, which spuriously demonstrates superior performance [76, 186]. Polygonal/polyhedral meshes naturally alleviate single node connections and prevent the occurrence of checkerboard patterns [209]. As previously mentioned, they also provide flexibility in discretizing complicated design domains which arise in real world applications. There are couple of approaches available in the literature to deal with polytopes. They can mainly be categorized as polygonal/polyhedral shape function-based methods and Mimetic Finite Difference (MFD) methods. Both methods are reviewed briefly in the subsequent sections.

1.3.1 Polygonal/Polyhedral shape function-based methods

Polygonal/Polyhedral shape function-based methods are one approach for the numerical solution of the discrete state equations on arbitrary meshes. The rational polygonal interpolants developed by Wachspress [222] laid the foundation for future research in the field of polygonal/polyhedral interpolants [234, 199, 84, 85, 200, 196]. Later, Warren [234] extended the Wachspress shape functions to three-dimensional convex polytopes. Natural neighbor shape functions are another class of polygonal interpolants which can be further subdivided into the Sibson coordinates [182, 199] and Non-Sibson coordinates or Laplace shape functions [32, 105, 68, 199, 200]. Wachspress and natural neighbor shape functions can only handle convex polygons. The mean value coordinates developed by Floater [84], subsequently extended to three-dimensions [86], are well-defined for concave polygons, but can result in negative values [108]. The metric coordinate method [138] is another approach that is applicable to concave polygons. Another class of polygonal/polyhedral shape functions, known as maximum entropy shape functions, was developed by Sukumar [196]; Arroyo and Ortiz [19]

Interpolants	References	Concave elements	Remarks
Wachspress	[222, 234, 223]	No	Earliest interpolant based on rational polynomials
Sibson	[182, 199]	No	Utilizes Voronoi tessellations to construct interpolant which reduces to the ratios of areas of Voronoi cells
Non-Sibson	[32, 105, 68, 199]	No	Also based on Voronoi tessellations. The interpolant is a function of Lebesgue measure of Voronoi edge and L^2 distance norm
Mean value coordinates	[84, 86]	Yes	Interpolant is a function of geometric quantities - L^2 distance norm and area
Metric coordinate method	[138]	Yes	Uses measures such as edge length, signed area of triangle, and trigonometric functions of sine and cosine to construct the shape functions
Maximum entropy	[196, 19, 108]	Yes	Shape functions and their derivatives are obtained by maximizing the Shannon's entropy function under prescribed boundary conditions
Harmonic coordinates	[114, 140, 40]	Yes	Shape functions and their derivatives are obtained by solving the Laplace equation hierarchically

Table 1.1: Summary of the polygonal/polyhedral interpolants.

which is based on the Jaynes's principle of maximum entropy for convex polytopes. Later, Hormann and Sukumar [108] extended the maximum entropy shape functions to arbitrary polytopes using prior functions. For a detailed overview of the main developments in the field of conforming polygonal interpolants refer to [198]. A summary of the polygonal/polyhedral interpolants is provided in Table. 1.1.

1.3.2 Mimetic Finite Difference (MFD) methods

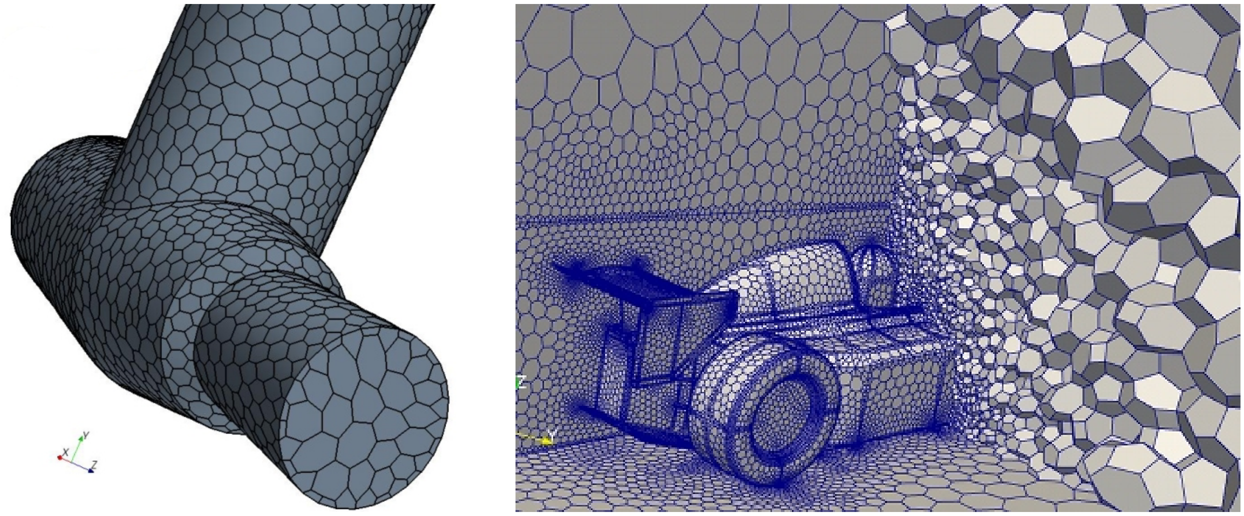
As briefly mentioned before, in two-dimensions, polygonal shape function-based methods can be efficiently implemented using the iso-parametric mapping scheme for numerical integration [200]. However, in three-dimensions, numerical integration can only be performed in physical coordinates and, for numerically accurate results, quadrature rules of very high order need to be used which increases the computational cost. Recently, a new set of numerical methods, known as Mimetic Finite Difference (MFD) methods, have emerged. The main characteristic which differentiates the MFD methods and the standard finite element (FE) approaches is that in the MFD methods, there are no explicitly defined shape functions associated with the discrete degrees of freedom. Thus, the continuous differential operators; such as, the div, grad, curl and trace, are approximated or *mimicked* by their discrete counterparts; which utilize the discrete quantities defined only at the degrees of freedom.

This provides greater flexibility in the geometric shapes of the admissible elements. High quality, skewed, degenerate and even non-convex polyhedra are all admissible. The MFD methods have been used to solve problems such as linear diffusion [110, 144, 129, 48, 99], electromagnetics [111], gas dynamics [55], linear elasticity in mixed weakly symmetric form using Hellinger-Reissner formulation [25] and Stoke's flow problem [28, 30]. Higher order MFD methods have also been developed [195, 31, 99, 29] which make use of nodal as well as additional degrees of freedom; namely moments on edges, faces and elements. The MFD methods have been explored in three-dimensional problems using polyhedral meshes [49, 48, 47, 28, 30].

Since their inception, MFD methods have evolved from a finite difference/finite volume framework to more of a finite element-type framework, known as the Virtual Element Method (VEM). Unlike MFD methods, in VEM the degrees of freedom are attached to trial functions defined inside the elements, similar to finite element-type methods (FEM). But unlike FEM, the approximation space is not computed explicitly. At the heart of VEM is the kinematic decomposition of element deformation states into rigid body, constant strain and higher order modes. In VEM, the basis which span the rigid body and constant strain spaces are defined along with the projection maps to extract the corresponding deformation states. Using these projection maps, the discrete bilinear forms and load linear forms are constructed without the need for computationally expensive evaluation of canonical basis functions and their derivatives inside the element. The construction ensures that the linear deformation states are captured exactly, thus guaranteeing the satisfaction of engineering patch test and optimal convergence of numerical solutions. The discrete bilinear form is constructed to have two components; a consistency component and a stability component. The degrees of freedoms (DOFs) are carefully chosen so as to allow accurate DOF-based computations when only the polynomials are involved, i.e. the consistency term calculation. The Virtual Element Method is relatively new and has been used to solve two-dimensional Laplace [26] and elasticity problems [27, 91, 90, 88]. It has also been explored for plate bending [50]. One goal of the current work is to explore VEM for three-dimensional linear elastic problems.

1.4 Research motivation

The main focus of the current work is to explore the benefits of polytopes in numerical analysis and to develop an efficient technique for three-dimensional topology optimization. The reason polytopes are attractive is because they facilitate automatic discretization of complicated domains (Fig. 1.4). Tetrahedrons offer similar meshing benefits but numerically inaccurate results are obtained for problems such as bending, if lower order tetrahedrons are used. For computational fluid dynamics simulations, researchers have shown that simulations on polyhedral meshes converge faster than tetrahedral meshes and provide higher accuracy [167].

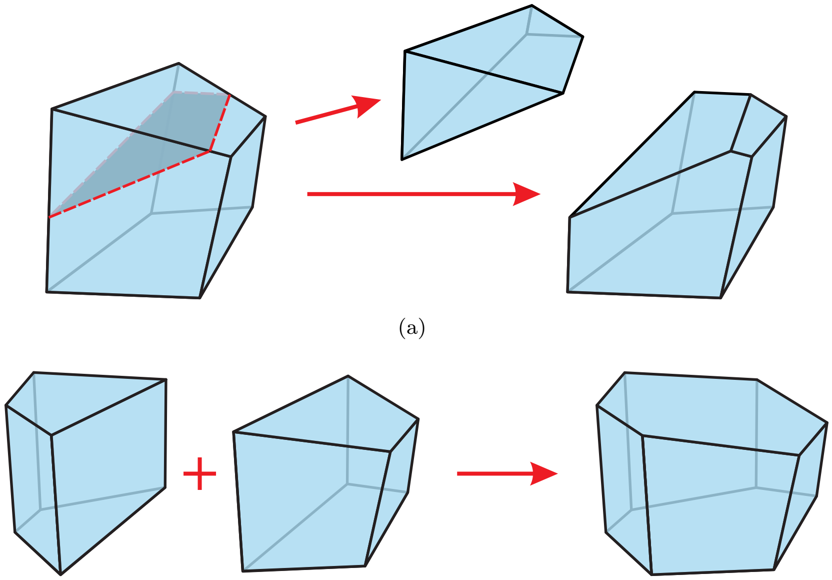


(a) (b)

Figure 1.4: Complicated domains discretized using polyhedral meshes in computational fluid dynamics simulations. (a) Valve (www.cd-adapco.com), (b) Formula 1 racecar (www.symscape.com).

Numerical simulations of phenomena, such as fracture, often require refinement and coarsening of typical finite elements. Such techniques generate elements which are inherently polyhedral (Fig. 1.5). Gradient calculation is an integral part of finite volume schemes. The fact that polyhedrons have many faces leads to a superior gradient approximation.

As discussed in Section 1.3.1, many polygonal/polyhedral shape functions-based methods; such as, Wachspress, and maximum entropy, are available in the literature for numerical analysis. In three-dimensions, numerical computations need to be performed in physical coordinates, as iso-parametric mapping is infeasible. In addition, numerical accuracy is dictated by quadrature order used. A Mimetic Finite Difference-inspired Virtual Element Method (VEM), discussed in Chapter 4, addresses both the issues of accuracy and efficiency. To support the arguments presented here, a comparative study is conducted between the maximum entropy shape function-based analysis (MAXENT) (refer to [196]) and the VEM (discussed in Chapter 4) using the displacement patch test. The domain is a cube of dimensions $1 \times 1 \times 1$ discretized using an increasing number of polyhedrons (e.g. 50, 100, 200). An arbitrary linear displacement of the form $\mathbf{U} = \mathbf{A}\mathbf{X}+\mathbf{B}$, where $\mathbf{A} = 1/100\,[2\ 1\ 3;\ 3\ 4\ 2;\ 4\ 3\ 1]$ and $\mathbf{B} = 1/100\,[1\ 2\ 3]^T$, is applied to the nodes on the boundaries of the cube. No forces are applied to the system. Accuracy is measured in terms of relative L^2 displacement errors. In the MAXENT approach, to numerically integrate the stiffness matrix, polyhedrons are partitioned into tetrahedrons and corresponding quadrature rules for tetrahedrons are used. On a CVT mesh of 50 elements, the observed relative L^2 errors in displacement for the



(a)

(b)

Figure 1.5: Mesh refinement and coarsening techniques applied to traditional finite elements produces elements which are inherently polyhedral. (a) Refinement. (b) Coarsening.

MAXENT approach are 1.84×10^{-1}, 8.14×10^{-2} and 4.93×10^{-2} when first, second and forth order quadrature rules (consisting of 1, 4 and 11 points) are used for each tetrahedron indicating that the displacement patch test is not passed. Moreover, the displacement error decreases at a low rate, with respect to order of integration. Next, performance of the MAXENT approach is compared against VEM under mesh refinement. Second order scheme (four Gauss points) is used for integration of the stiffness matrix in the MAXENT approach. The relative L^2 displacement error for the MAXENT approach, corresponding to different mesh discretizations, continues to be large; where as for VEM, the errors are of the order of machine precision. These VEM results serve as source of encouragement for the current research using polyhedrons.

Number of polyhedrons	MAXENT	Present
50	8.14×10^{-2}	5.72×10^{-15}
100	9.24×10^{-2}	1.91×10^{-14}
200	6.63×10^{-2}	2.66×10^{-14}

Table 1.2: Comparative study between the MAXENT approach and the VEM approach discussed in Chapter 4 using the displacement patch test under mesh refinement. Columns 2 and 3 corresponding to relative L^2 displacement errors.

1.5 Thesis outline

The remainder of this thesis is organized as follows. Chapter 2 presents a critical assessment of representative differential equation-driven methods for topology optimization. Four level-set methods and one phase-field method are exhaustively analyzed to identify their strengths and weaknesses. The goal of this study is to provide some guidelines for future developments in the field of differential equation-driven methods. Before embarking on the development of a numerical approach for three-dimensional topology optimization using polyhedrons, leveraging the knowledge of differential equation-driven methods obtained from Chapter 2, a two-dimensional phase-field based optimization scheme for polygons is provided in Chapter 3. A centroidal Voronoi tessellation (CVT) based finite volume scheme is used to solve the governing Allen-Cahn equation on unstructured polygonal meshes. Chapter 4 discusses a Mimetic Finite Difference-inspired approach for three-dimensional linear elasticity. Detailed steps for the construction of the stiffness matrix and force vector are shown. To illustrate the accuracy of the present approach, numerical verification studies are presented which include the displacement patch test and shear loaded cantilever beam bending problem. Using the numerical scheme discussed in Chapter 4 to solve the elasticity equation, a three-dimensional topology optimization scheme is developed in Chapter 5. The effectiveness of the optimization scheme is shown using numerical examples on various complicated design domains. Since the Mimetic Finite Difference method presented in this work provides greater flexibility in the geometric shapes of the admissible elements, Chapter 6 explores topology optimization using tessellations as a means to coalesce engineering and art. M.C. Escher's tessellations, created using recognizable figures such as birds and animals, provides a tool to add aesthetics to otherwise bland engineering designs. Finally, some concluding remarks are provided in Chapter 7 along with suggestions for potential extensions of the current work.

Chapter 2

A critical comparative assessment of differential equation-driven methods for structural topology optimization

Density-based methods have been predominantly used in structural optimization [39]. Recently, differential equation-driven methods have emerged as an alternate approach for structural topology optimization. In such methods, the design is evolved using special differential equations. Implicit level-set methods are one such set of approaches in which the design domain is represented in terms of implicit functions and generally (but not necessarily) use the Hamilton-Jacobi equation as the evolution equation. Another set of approaches are referred to as phase-field methods; which generally use a reaction-diffusion equation, such as the Allen-Cahn equation, for topology evolution. Current work analyzes four level-set methods and one phase-field method, which are representative of the literature. In order to evaluate performance, all the methods are implemented in MATLAB and studied using two-dimensional compliance minimization problems. The goal of this work is to serve as a guide for future developments in the field of topology optimization.

2.1 Introduction

Shape and topology optimization using level-sets have been explored by many researchers [180, 153, 7, 226, 8]. Wang et al. [226] presented a structural topology optimization method for bi-material systems and studied the compliance minimization problem for linear elastic materials. Wang and Wang [227] extended Wang et al. [226] method to multi-material systems. During the same time, Allaire et al. [7, 8] proposed a structural optimization technique combining the classical shape derivative and the level-set method. They solved the compliance minimization, compliant mechanism design, and design dependent load problems for linear elastic systems and also investigated nonlinear elasticity problems. Both Wang et al. [226] and Allaire et al. [8] used an upwind scheme [179] for the discrete solution of the Hamilton-Jacobi equation. Later, Allaire and Jouve extended their level-set method for eigenvalue, multiple load [5] and minimum stress design problems [6]. Also, Yamada et al. [240] and Yamasaki et al. [241] solved the free vibration eigenvalue topology problem using the level-set method. Other researchers (Wei and Wang [236]; Luo et al. [132, 133]; Challis and Guest [60]) have also used shape sensitivities for the evolution of design using the level-set methods.

There have been some attempts to develop level-set methods which do not involve solving the Hamilton-Jacobi equation for the evolution of shapes. For example, Belytschko et al. [34] proposed a topology optimization method where the weak form of the equilibrium equation is expressed as a Heaviside step function of the level-set function. The Heaviside function is subsequently regularized to enable evaluation of sensitivities of the objective functions and constraints. Van Dijk [216, 215] developed a method where the design domain is implicitly represented by a level-set function and the design is evolved using a steepest-descent type update scheme which utilizes the discrete sensitivities of the objective function. Non-Hamilton-Jacobi based level-set methods can also be seen in the fluid topology optimization literature. Cunha [72] presented an Eulerian-type parametric level-set based shape optimization method where the design domain is expressed in terms of level-sets and the design variables are defined at the mesh vertices. They used it to obtain shapes which reproduce a particular velocity field for incompressible, viscous fluid using Navier-Stokes and Stokes flow models. Pingen et al. [169] examined a parametric level-set method for fluid topology optimization using a hydrodynamic Lattice Boltzmann method. Kreissl et al. [125] find the optimal layout of fluidic devices employing an explicit level-set method along with a Lattice Boltzmann solver. No-slip boundary conditions are enforced along the solid-fluid interface using second-order accurate interpolation schemes. Kriessl and Maute [124] use an approach similar to Kreissl et al. [125] and model the flow field by the incompressible Navier-Stokes equations discretized by the extended finite element method (XFEM). Also, they enforce a no-slip condition along the solid-fluid interface by applying the stabilized Lagrange multiplier method.

Hamilton-Jacobi based level-set methods generally have a tendency to become too steep near the boundaries (hence have high spatial gradients) or too flat during the course of evolution, thus affecting the accuracy and rate of convergence of the level-set method, see Fig. 2.1. Without any control over the gradients near the boundaries, the evolution algorithm tends to become unstable (indicated by sharp rise in the level-set values), leading to an inaccurate estimation of the boundary normal. One option to control the gradients is to periodically reinitialize the level-set function; for example, to a signed-distance function, to maintain the numerical accuracy. In one reinitialization approach, the zero level-set function isocontour; which represents the shape boundary, is approximated using the same shape functions as in FEM and then the distances from the discretized isocontour are computed

[67, 241]. Another popular approach is to solve a specifically tailored partial differential equation [202, 8]. Alternatively, Sethian [178, 179] proposed a reinitialization scheme, known as Fast Marching Method (FMM), which allows one to solve the boundary value problem, without any iteration, using an optimal ordering of the grid points.

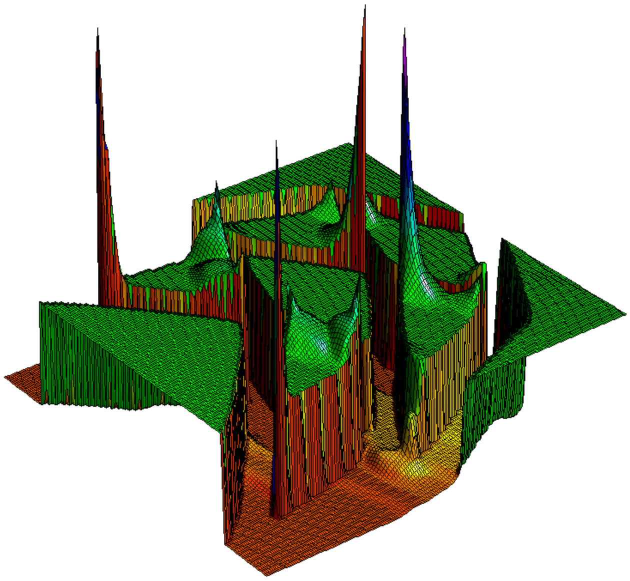


Figure 2.1: Level-set function without any slope control near the design boundaries.

The works of Sethian and Wiegmann [180], Wang et al. [226] and Allaire et al. [8] illustrate that level-set methods allow for drastic change in topology during evolution, but can have final configurations that are very sensitive to the chosen initial configuration. This problem can be attributed to the fact that there are no inherent hole nucleation mechanisms in the level-set based topology optimization methods for two-dimensional problems. Pre-existing holes can only merge or cancel. Several attempts have been made to alleviate this issue. One set of popular approaches is the use of topological derivatives. Topological derivative approaches can be further sub-categorized into two strategies. In the first strategy, holes are nucleated by removing material from the locations where the topological derivative takes the least value [4, 225]. A second strategy modifies the Hamilton-Jacobi equation to include topological sensitivity information [52, 16, 102, 59]. Amstutz and Andrä [16] only use topological derivatives and no shape derivatives in their evolution equation. Another hole nucleation approach is based on radial basis functions [231, 232, 233, 106]. This approach does not make use of topological derivative information; instead the Hamilton-Jacobi equation is reduced to a set of ordinary differential equations using multi-quadratic splines and solved using Euler's method. The elimination of reinitialization and adoption of smoothed naturally extended velocities aids the creation of new holes. Use of radial basis functions helps maintain the smoothness of the level-set function. Other researchers have also used radial basis functions to parameterize the level-set function [74, 236, 134, 125]. In case of three-dimensional optimization, Allaire et al. [8] have shown that the traditional level-set approach using the Hamilton-Jacobi equation with shape derivatives is able to nucleate new holes in the domain due to the pinching of thin walls. Other researchers (e.g. Challis et al. [62, 61]) have arrived at similar conclusions for three-dimensional optimization with the level-set method. Recently, Van Dijk et al. [218] published a review article which provides a detailed overview of the different level-set methods for structural topology optimization.

Phase-field based methods are another category of approaches rapidly gaining popularity. Essentially, the phase-field method is a diffuse interface model where boundaries between phases are not sharp, but considered to have a finite thickness, hence providing a smooth transition for the physical quantities between the phases. It has been used in materials science to study the phase transition phenomenon. Cahn and Hillard [54], and Allen and Cahn [11] used the theory of phase transition to study the liquid phases with variable densities. Phase-field methods have been applied in a wide variety of fields such as fracture mechanics [17]), visual reconstruction [139], and crystal growth simulations [121]. Bourdin and Chambolle [46] used the phase-field method to study compliance minimization problems subjected to design dependent loads such as pressure and gravity. Wang and Zhou [228] used the van der Waals-Cahn-Hillard phase transition theory to propose a phase-field method for topology optimization of a design domain consisting of bi-phase systems. Later, Wang and Zhou [229] extended the method to tri-phase systems. In addition, Burger and Stainko [53] proposed a phase-field method based relaxation scheme for structural topology optimization problems with local stress constraints. Takezawa et al. [208] utilized the Allen-Cahn equation [11], a time dependent reaction-diffusion equation, for the evolution of topologies in structural optimization problems. The uniqueness of their approach was the utilization of the objective function sensitivity to construct the double well potential function. Also, Wallin et al. [224] presented a topology optimization procedure which uses a volume preserving Cahn-Hillard model and an adaptive finite element formulation.

The goal of the current work is to study some of the prominent, and characteristically different, level-set and phase-field methods which are representative of the literature. Our efforts focus on critically understanding the following five methods: the AJT level-set method [8], the DLK level-set method [216], the WW level-set method [231], Challis' level-set method [59] and the TNK phase-field method [208].[1] *We acknowledge that the cited authors may have substantially improved their methods after the aforementioned papers were published. We would like to clarify that our goal is not to address the latest contributions of each author, but to investigate what has been reported in the five specific papers cited above, namely Allaire et al. [8], Van Dijk et al. [216], Wang and Wang [231], Challis [59], and Takezawa et al.*

---

[1] The acronyms, AJT, DLK, WW and TNK are used to abbreviate the last name of the authors in each corresponding paper.

[208]. Although, the level-set and phase-field methods have been used to study a wide variety of problems subjected to different constraints, we will focus on *two-dimensional compliance minimization problems* only. It should be noted that the methods covered in this work are representative and not exhaustive of the differential equation driven methods for topology optimization. We hope this work will be a guide for future additional developments in the field of topology optimization.

The remainder of the chapter is organized as follows. Section 2.2 discusses the formulation of the topology optimization problem. Section 2.3 briefly reviews the differential equation-driven methods analyzed in this work. Section 2.4 discusses the results of our analysis of these methods, using several numerical examples. Finally, Section 2.5 provides some concluding remarks.

2.2 Problem statement for differential equation driven methods

For the current comparative study, the topology optimization problem (1.5) is written as unconstrained optimization problem using Lagrange multiplier method as:

$$\inf_{\phi} \bar{J}\left(\phi\right) = J\left(\phi\right) + \lambda P\left(\phi\right) \tag{2.1}$$

Here, λ is a positive Lagrange multiplier and volume constraint, $P(\phi)$, is given by

$$P(\phi) = \left(\int_{\Omega} \rho(\phi) d\boldsymbol{x}/|\Omega| - V_f\right) \tag{2.2}$$

where V_f is the prescribed volume fraction. In discrete form (2.1) can be rewritten as:

$$\inf_{\phi} \bar{J}\left(\phi\right) = \mathbf{F}^T\mathbf{U} + \lambda\left(\boldsymbol{V}^T\boldsymbol{\rho}(\phi) - V_f\right) \tag{2.3}$$

where $\mathbf{F}$ is the discretized global force vector, $\mathbf{U}$ is the global nodal displacement vector, $\boldsymbol{V}$ is an array of the fractional areas of elements, $\boldsymbol{V} = [A_1, A_2, ..., A_n]^T/\sum_i A_i$, the A_i's are element areas and $\boldsymbol{\rho}(\phi)$ is the element density array. In the case of level-set methods, the design function, ϕ, is defined as:

$$\begin{cases} \phi = 0 & \boldsymbol{x} \in \partial\omega \cap \Omega, \\ \phi < 0 & \boldsymbol{x} \in \omega, \\ \phi > 0 & \boldsymbol{x} \in \left(\Omega \backslash \left(\omega \cup \partial\omega\right)\right). \end{cases} \tag{2.4}$$

The phase-field method starts with the boundary conditions specified in Fig. 1.2. At any time during the optimization process, the phase-field domain can be illustrated by Fig.

2.2. The working domain, Ω, is considered to be composed of two phases (ω_0, ω_1) and the interfacial boundary between the phases, ξ, which is called the diffuse interface (Fig. 2.2). The diffuse interface acts as the transition zone between the two phases. The working domain Ω contains all admissible shapes ω, i.e., $\omega \subseteq \Omega$. Here $\omega \subset (\omega_1 \cup \xi)$. Accordingly, the design function, ϕ, for the phase-field method is defined as:

$$\begin{cases} \phi = 1 & \boldsymbol{x} \in \omega_1, \\ 0 < \phi < 1 & \boldsymbol{x} \in \xi, \qquad \text{Diffuse interface} \\ \phi = 0 & \boldsymbol{x} \in \omega_0. \end{cases} \tag{2.5}$$

For uniformity of notation, we represent the design variable as ϕ for both the level-set and phase-field methods. However, for the phase-field method, the design function, ϕ, is the same as the density function, ρ.

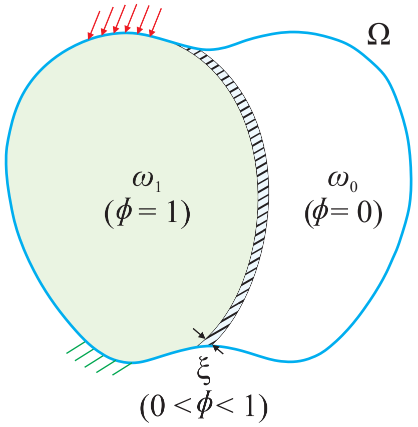


Figure 2.2: Phase-field working domain Ω. Domains ω_1, ω_0 and ξ represent solid phase, void phase and diffuse interface, respectively. Here, ϕ represents the design function.

2.3 A brief review of differential equation-based methods

Two general categories of differential equation-driven approaches for topology optimization can be found in the literature - level-set methods and phase-field methods. We specifically selected four representative level-set approaches and one representative phase-field approach to review in this work. We first present a brief summary of the five methods. Note that all the discussions about the AJT, DLK, WW and TNK methods in sections 2.4 and 2.5 are based on our own implementation and the discussion on Challis' method are based on the code provided in [59] with few modifications, to maintain uniformity amongst methods investigated here, which are pointed out in the section below.

2.3.1 AJT level-set method (Allaire et al. 2004)

In the level-set method proposed by Allaire et al. [8], the design front propagation is governed by the Hamilton-Jacobi equation and the advection velocity is derived from a shape sensitivity analysis. The solid phase, $\phi < 0$, is assumed to be filled with material of elasticity tensor $\mathbf{C}^0$. In order to avoid singularities in the global stiffness matrix, the void region, $\phi > 0$, is filled with a weak phase with elasticity tensor $k_{min}\mathbf{C}^0$. Here, k_{min} is chosen as 10^{-3}. This is called the Ersatz material approach. Thus, the effective elasticity tensor $\mathbf{C}$ for the entire design domain Ω is defined as:

$$\mathbf{C}(\phi) = \rho_e(\phi)\mathbf{C}^0 \qquad \text{with} \qquad \rho_e(\phi) = \begin{cases} 1 & \phi < 0, \\ k_{min} & \phi > 0. \end{cases} \tag{2.6}$$

Here, density ρ_e is taken as piecewise constant for each element. The procedure to calculate element densities for the elements which are cut by the zero level-set function is not clearly described in [8]. One possible approach is provided by Allaire et al. [9]. In this approach, first the rectangular element is split into four triangles and the central node is assigned the average of the level-set function values at the rectangular vertices. Then, linear interpolation is used to obtain densities corresponding to each triangle. The element density is the average of the densities of the constituent triangles.

The topology is evolved over fictitious time using the Hamilton-Jacobi equation:

$$\frac{\partial \phi}{\partial t} + v\,|\nabla \phi| = 0, \qquad \frac{\partial \phi}{\partial \boldsymbol{n}} = 0 \text{ on } \partial\Omega \tag{2.7}$$

where $\boldsymbol{n}$ is the normal vector and the advection velocity, v, is obtained from the shape sensitivity analysis. For objective (2.1), v is given as:

$$v = \boldsymbol{\varepsilon}(\boldsymbol{u})^T \mathbf{C} \boldsymbol{\varepsilon}(\boldsymbol{u}) - \lambda \tag{2.8}$$

In our implementation, design function, ϕ, is nodal based. Hence, the velocities need to be calculated at the nodes as well. First, (2.8) is integrated over each finite element to obtain elemental velocities. Velocities at the nodes are obtained by taking the average of the elemental velocities surrounding each node. We use a second order upwind scheme [179] to solve (2.7).

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n} - \Delta t \left(\max\left(v_{i,j}, 0\right) \nabla^{+} + \min\left(v_{i,j}, 0\right) \nabla^{-} \right) \tag{2.9}$$

Here, $\phi_{i,j}^n$, $v_{i,j}$ are values of ϕ, v for the nth iteration at the node located at $\boldsymbol{x}_{i,j}$. The parameters ∇^+ and ∇^- in (2.9) are given by:

$$\begin{aligned}\nabla^+ &= \left[\max(A_1,0)^2 + \min(A_2,0)^2 + \max(A_3,0)^2 + \min(A_4,0)^2\right]^{1/2} \\ \nabla^- &= \left[\max(A_2,0)^2 + \min(A_1,0)^2 + \max(A_4,0)^2 + \min(A_3,0)^2\right]^{1/2}\end{aligned} \tag{2.10}$$

where the terms A_1, A_2, A_3, A_4 are:

$$\begin{aligned} A_1 &= D^{-x}_{i,j} + \frac{\Delta x}{2} m\left(D^{-x-x}_{i,j}, D^{+x-x}_{i,j}\right), & A_2 &= D^{+x}_{i,j} - \frac{\Delta x}{2} m\left(D^{+x+x}_{i,j}, D^{+x-x}_{i,j}\right), \\ A_3 &= D^{-y}_{i,j} + \frac{\Delta y}{2} m\left(D^{-y-y}_{i,j}, D^{+y-y}_{i,j}\right), & A_4 &= D^{+y}_{i,j} - \frac{\Delta y}{2} m\left(D^{+y+y}_{i,j}, D^{+y-y}_{i,j}\right). \end{aligned} \tag{2.11}$$

The function m and derivatives $D^{+x+x}_{i,j}$, $D^{+x-x}_{i,j}$, D^{+x} and D^{-x} are defined as:

$$m(x,y) = \begin{cases} \begin{cases} x & \text{if } |x| \le |y| \\ y & \text{if } |x| > |y| \end{cases} & \text{for } xy \ge 0, \\ \quad 0 & \text{for } xy < 0. \end{cases} \tag{2.12}$$

$$D^{+x+x} = \frac{\phi^n_{i+2,j} - 2\phi^n_{i+1,j} + \phi^n_{i,j}}{(\Delta x)^2}, \qquad D^{+x-x} = \frac{\phi^n_{i+1,j} - 2\phi^n_{i,j} + \phi^n_{i-1,j}}{(\Delta x)^2}, \tag{2.13}$$

$$D^{+x} = \frac{\phi^n_{i+1,j} - \phi^n_{i,j}}{\Delta x}, \qquad D^{-x} = \frac{\phi^n_{i,j} - \phi^n_{i-1,j}}{\Delta x}. \tag{2.14}$$

Other derivatives, $D^{-x-x}_{i,j}$, $D^{+y+y}_{i,j}$, $D^{+y-y}_{i,j}$, $D^{-y-y}_{i,j}$, D^{+y} and D^{-y}, can be calculated in a similar way. Also, Δx and Δy are the distances between the nodes in the x and y direction, respectively. Over the course of evolution, the level-set function may become too steep or too flat which may result in an inaccurate approximation of the normal, $\boldsymbol{n}$. For numerical accuracy, the level-set function needs to be reinitialized/smoothed periodically. Reinitialization to a signed distance function is one option; which can be achieved by solving the equation:

$$\begin{gathered} \frac{\partial \phi}{\partial t} + sign\left(\phi_0\right)\left(|\nabla\phi| - 1\right) = 0, \qquad \frac{\partial \phi}{\partial \boldsymbol{n}} = 0 \text{ on } \partial\Omega \\ \text{with} \quad \phi(x, t=0) = \phi_0(x). \end{gathered} \tag{2.15}$$

At steady state, the above equation reduces to $|\nabla\phi| = 1$, the solution of which is a signed distance function. A second order upwind scheme, discussed previously, is used to solve the reinitialization equation (2.15).

2.3.2 DLK level-set method (Van Dijk et al. 2009)

Traditionally, level-set methods use shape derivatives to estimate the advection velocity for the Hamilton-Jacobi equation. Van Dijk et al. [216] proposed a modified version of level-set method, which uses discrete sensitivity of the objective function in a steepest-descent type evolution scheme. Their work was motivated by the argument that the use of shape sensitivities to derive the velocity field may cause inconsistencies in the discretization of the velocity field and governing equations, which may lead to poor performance. They also dealt with multiple constraints, which is not very common in the literature of differential equation-driven methods for topology optimization. The authors use an adjoint approach to calculate the discrete sensitivities, modifying the objective function (2.3) by adding the zero function $\boldsymbol{Q}^T\boldsymbol{R}\left(\boldsymbol{\rho}(\phi),\mathbf{U}\right)$ as follows:

$$\inf_{\phi} \bar{J}\left(\phi\right) = \mathbf{F}^T\mathbf{U} + \lambda\left(\boldsymbol{V}^T\boldsymbol{\rho}(\phi) - V_f\right) + \boldsymbol{Q}^T\boldsymbol{R}\left(\boldsymbol{\rho}(\phi),\mathbf{U}\right) \tag{2.16}$$

where $\boldsymbol{Q}$ is a Lagrange multiplier vector and $\boldsymbol{R}\left(\boldsymbol{\rho}(\phi),\mathbf{U}\right)$ is the residual obtained from the finite element discretization of the equilibrium equation. Apart from the volume constraint, the authors also impose displacement constraint on the optimization problem. As mentioned earlier, for simplicity, we will only investigate the compliance minimization problem with volume constraint. The sensitivity of the objective function (2.3) is calculated by differentiating (2.16) as follows:

$$\frac{\partial \bar{J}}{\partial \boldsymbol{\phi}} = \frac{\partial \bar{J}}{\partial \boldsymbol{\rho}}\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\phi}} = \left(\boldsymbol{Q}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\rho}} + \lambda\boldsymbol{V}^T\right)\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\phi}} \tag{2.17}$$

where $\boldsymbol{Q}$ is computed by solving the adjoint system:

$$\frac{\partial \bar{J}}{\partial \mathbf{U}} = \mathbf{F}^T + \boldsymbol{Q}^T\frac{\partial \boldsymbol{R}}{\partial \mathbf{U}} = \mathbf{0} \tag{2.18}$$

In the level-set literature, it is often not clear how the level-set function is mapped to the density domain in the design interface region. The authors (Van Dijk et al. [216]), however, present a clear mapping scheme using an approximate Heaviside function as shown below:

$$\rho_e(\phi) = \frac{\int_{\Omega_e} H\left(\phi\right) d\boldsymbol{x}}{\int_{\Omega_e} d\boldsymbol{x}} \tag{2.19}$$

where Ω_e represents the element domain and $H(\phi)$ is an approximate Heaviside function given by:

$$H(\phi) = \begin{cases} 1 & \phi < -h, \\ \frac{1}{4}(1-\epsilon)\left(\left(\frac{\phi}{h}\right)^3 - 3\left(\frac{\phi}{h}\right) - 2\right) + 1 & -h \le \phi \le h, \\ \epsilon & \phi > h. \end{cases} \tag{2.20}$$

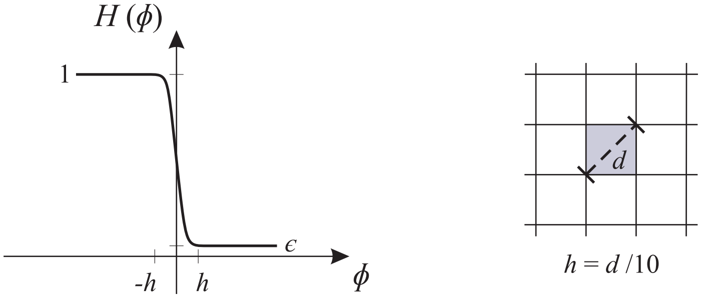


Figure 2.3: Approximate Heaviside function to evaluate element densities in the elements cut by the zero level-set, $\phi = 0$.

Here, $h = d/10$, where d is the length of the diagonal of the finite element. Thus, $\partial\boldsymbol{\rho}/\partial\boldsymbol{\phi}$, needed to evaluate the objective function sensitivity, can be calculated as:

$$\frac{\partial \rho_e}{\partial \phi_i} = \frac{\int_{\Omega_e} \delta(\phi) N_i d\boldsymbol{x}}{\int_{\Omega_e} d\boldsymbol{x}} \tag{2.21}$$

where $\delta(\phi)$ is the derivative (Dirac delta function) of the approximate Heaviside function (2.20):

$$\delta(\phi) = \begin{cases} 0 & \phi < -h, \\ \frac{1}{4}(1-\epsilon)\left(\frac{3\phi^2}{h^3} - \frac{3}{h}\right) & -h \le \phi \le h, \\ 0 & \phi > h. \end{cases} \tag{2.22}$$

The integral in the numerator of (2.19) and (2.21) is approximated by sampling ϕ at 10×10 points over each finite element. In the literature, other approximate Heaviside based mapping schemes can be found which utilize polynomial functions [226, 131, 116, 215] and trigonometric functions [34, 101, 132, 169]. Finally, the level-set is evolved according to the following direct update scheme:

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n - t\frac{\partial \bar{J}/\partial \boldsymbol{\phi}}{|\partial \bar{J}/\partial \boldsymbol{\phi}|} \tag{2.23}$$

where $\boldsymbol{\phi}^n$ and t represent the design function array at the nth iteration and scaling constant, respectively. As mentioned before, the authors' update scheme (2.23) is different from the typical advection equation updates (2.7) used for the level-set methods. The DLK approach requires very frequent reinitialization to achieve convergence and this is because the direct update (2.23) does not preserve the magnitude of the gradient of the level-set function. Reinitialization is performed using (2.15).

2.3.3 WW level-set method (Wang and Wang 2006)

We next look at the radial basis function-based level-set method proposed by Wang and Wang [231]. Similar approaches can be seen in [232, 233]. Radial basis functions are radially symmetric functions centered at a specific point. Wang and Wang [231] used a particular form of radial basis functions, known as multi-quadric splines, and method of lines to transform the Hamilton-Jacobi partial differential equation into a system of ordinary differential equations. The authors claim that their method does not require reinitialization (which is expensive) and is insensitive to initial designs. The general form of a radial basis function, centered around $\boldsymbol{x}_i$, can be written as $N_i(\boldsymbol{x}) = N(|\boldsymbol{x} - \boldsymbol{x}_i|)$. The multi-quadratic splines used in their work can be expressed as:

$$N_i(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x}_i)^2 + c_i^2} \tag{2.24}$$

where c_i is assumed to be a constant shape parameter which affects the flatness of the splines. The design function, $\phi(\boldsymbol{x})$, can be written in terms of multi-quadratic splines with m nodes or knots as:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i N_i(\boldsymbol{x}) + p(\boldsymbol{x}) \tag{2.25}$$

where α_i is the weight of the radial basis function at the ith knot and $p(\boldsymbol{x}) = p_0 + p_1 x + p_2 y$ is a first degree polynomial to account for the linear and constant portions of the function. Using the orthogonality conditions ($\sum_i^m \alpha_i = 0,\ \sum_i^m \alpha_i x_i = 0,\ \sum_i^m \alpha_i y_i = 0$) and the given function values at m knots ($\phi(\boldsymbol{x}_i) = f_i,\ i = 1, 2, ..., m$), we get a system of $m + 3$ linear equations to solve for $m + 3$ unknown coefficients. In matrix notation, the above equations can be written as:

$$\boldsymbol{H\alpha} = \boldsymbol{f} \tag{2.26}$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{P}_N & \boldsymbol{P}_X \\ \boldsymbol{P}_X^T & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{P}_N = \begin{bmatrix} N_1(\boldsymbol{x}_1) & \cdots & N_m(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ N_1(\boldsymbol{x}_m) & \cdots & N_m(\boldsymbol{x}_m) \end{bmatrix}, \qquad \boldsymbol{P}_X = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_m & y_m \end{bmatrix} \tag{2.27}$$

$$\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_m \ p_0 \ p_1 \ p_2]^T , \qquad \boldsymbol{f} = [f_1 \ \cdots \ f_m \ 0 \ 0 \ 0]^T \tag{2.28}$$

Thus, (2.25) can be written as, $\phi(\boldsymbol{x}) = \boldsymbol{N}^T(\boldsymbol{x})\,\boldsymbol{\alpha}$, where $\boldsymbol{N}^T(\boldsymbol{x}) = [N_1(\boldsymbol{x}) \ \cdots \ N_m(\boldsymbol{x}) \ 1 \ x \ y]^T$. The Hamilton-Jacobi equation, used to update the level-set function, is both space and pseudo-time dependent. In the WW approach [231], space and time are assumed to be separable and the time dependency is lumped into the coefficients $\boldsymbol{\alpha}$. So we have

$$\phi(\boldsymbol{x}, t) = \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{\alpha}(t) \tag{2.29}$$

Using (2.29), the Hamilton-Jacobi equation (2.7) can be simplified into following ODE:

$$\boldsymbol{H}\frac{d\boldsymbol{\alpha}}{dt} + \boldsymbol{B}(\boldsymbol{\alpha}) = 0 \tag{2.30}$$

where

$$\boldsymbol{B}(\boldsymbol{\alpha}) = \begin{bmatrix} v(\boldsymbol{x}_1)|\nabla \boldsymbol{N}^T(\boldsymbol{x}_1)\boldsymbol{\alpha}| \\ \vdots \\ v(\boldsymbol{x}_m)|\nabla \boldsymbol{N}^T(\boldsymbol{x}_m)\boldsymbol{\alpha}| \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.31}$$

The authors chose Euler's method to solve the ODE (2.30). So, coefficients $\boldsymbol{\alpha}$ are updated as:

$$\boldsymbol{\alpha}(t^{n+1}) = \boldsymbol{\alpha}(t^n) - \Delta t \boldsymbol{H}^{-1}\boldsymbol{B}(\boldsymbol{\alpha}\,(t^n)) \tag{2.32}$$

Over the course of the evolution, the level-set function may become either too steep at the design interface or too flat. Typically, in the level-set method literature, in order to maintain its regularity, the level-set function is reinitialized periodically to a signed distance function. Wang and Wang [231] argued that due to infinite smoothness of the radial basis functions, accuracy of the normal vector can be maintained and thus operations such as reinitialization are not required. In our implementation of the WW method, we too have not used any reinitialization. We will discuss this issue in detail in Section 2.4. They use the shape derivatives to define the advection velocity at the design front; which is extended to the entire domain using the assumption that the strain is zero in the void region, $\boldsymbol{\varepsilon} = \boldsymbol{0}$. To aid in the smooth progress of the front, they introduce the following scheme for the advection velocity:

$$v(\boldsymbol{x}) = \begin{cases} \boldsymbol{\varepsilon}(\boldsymbol{u})^T \mathbf{C} \boldsymbol{\varepsilon}(\boldsymbol{u}) - \lambda & \phi(\boldsymbol{x}) < -h, \\ \widehat{v}(\boldsymbol{x}) & -h \le \phi(\boldsymbol{x}) \le h, \\ -\lambda & \phi(\boldsymbol{x}) > h. \end{cases} \tag{2.33}$$

Here, $\widehat{v}(\boldsymbol{x})$ is smoothed advection velocity around the zero level-set (design boundary) which can be written as:

$$\widehat{v}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{p}\in \boldsymbol{Z}} W(|\boldsymbol{p}-\boldsymbol{x}|)\; v(\boldsymbol{x})}{\sum_{\boldsymbol{p}\in \boldsymbol{Z}} W(|\boldsymbol{p}-\boldsymbol{x}|)} \tag{2.34}$$

where

$$W(|\boldsymbol{p}-\boldsymbol{x}|) = r_{min} - |\boldsymbol{p}-\boldsymbol{x}| \tag{2.35}$$

Also, the parameter $h = 1$, and $\boldsymbol{Z}$ is the neighborhood of $\boldsymbol{x}$, such that $-h \le \phi(\boldsymbol{x}) \le h$, inside the filter window of radius r_{min}. The effect of the smoothed velocity field is discussed in Section 2.4.6. We use (2.19) to map the level-set function to the density domain. Note that, in the WW approach [231], no mapping scheme is specified.

2.3.4 Challis' level-set method (Challis 2010)

Recently, Challis [59] published an educational article on the level-set method. In her work, Challis utilized Burger et al. [52] approach of modifying the traditional Hamilton-Jacobi equation (2.7) to include topological derivatives to generate holes and applied it to topology optimization problems. The modified Hamilton-Jacobi equation is given as:

$$\frac{\partial \phi}{\partial t} + v\,|\nabla \phi| = -wg \tag{2.36}$$

where $g(\boldsymbol{x})$ is a scalar field that is based on the topological sensitivities of the objective function. And w is a positive parameter which determines the influence of $g(\boldsymbol{x})$. The level-set function value is evaluated at the center of each element and is considered constant within each element. In Challis' approach, the design does not have any intermediate densities. The level-set is mapped to the density field as follows:

$$\rho_e = \begin{cases} 1 & \text{if } \phi_e < 0, \\ 0 & \text{if } \phi_e \ge 0. \end{cases} \tag{2.37}$$

Here, ρ_e and ϕ_e represent the element density and the level-set function value at the center of the element, respectively. We would like to point out that in Challis' approach [59] the objective function (2.38) is used, which we have modified to (2.3) to maintain uniformity amongst the methods investigated in this work. The corresponding normal velocity, v, which is calculated using the shape sensitivities, is also updated to (2.8).

$$\inf_{\phi} \bar{J}\left(\phi\right) = \mathbf{F}^T\mathbf{U} + \lambda\left(\boldsymbol{V}^T\boldsymbol{\rho}(\phi) - V_f\right) + \frac{1}{2\Lambda}\left(\boldsymbol{V}^T\boldsymbol{\rho}(\phi) - V_f\right)^2 \tag{2.38}$$

Subsequently, the forcing term, g, evaluated using topological derivatives of the objective function (2.3), follows as:

$$g = \begin{cases} \frac{\pi(\lambda+2\mu)}{2\mu(\lambda+\mu)}\left(4\mu\boldsymbol{\varepsilon}(\boldsymbol{u})^T\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) + (\lambda-\mu)\boldsymbol{u}^T\left(\boldsymbol{k}_{\mathrm{Tr}}\right)\boldsymbol{u}\right) - \pi\lambda^n & \text{if } \phi < 0, \\ 0 & \text{if } \phi \geq 0. \end{cases} \tag{2.39}$$

The term $\boldsymbol{u}^T\left(\boldsymbol{k}_{\mathrm{Tr}}\right)\boldsymbol{u}$ is the finite element approximation of $\mathrm{tr}(\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}))\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}))$ [59]. After substituting v and g back into the modified Hamilton-Jacobi equation (2.36), it is solved using the upwind finite difference scheme. Frequent reinitialization of the level-set function to a signed distance function is performed using MATLAB's `bwdist` function. For further details, readers are referred to Challis' paper [59]. We also modified the global stiffness matrix assembly in Challis' code to make it more efficient by using the sparse matrix assembly function available in MATLAB.

2.3.5 TNK phase-field method (Takezawa et al. 2010)

In the current work, apart from the level-set methods discussed earlier, we also look at the recently proposed phase-field method for structural topology optimization [208]. In the phase-field method, the solid phase, ω_1, is filled with material having elasticity tensor $\mathbf{C}^0$ and the region, ω_0, mimics a void with elasticity tensor $k_{min}\mathbf{C}^0$. Here, k_{min} is chosen to be 10^{-3}. The effective elasticity tensor $\mathbf{C}$ for the entire design domains (c.f. Fig. 2.2) can be written as:

$$\mathbf{C}\left(\phi\right) = \begin{cases} \mathbf{C}^0 & \boldsymbol{x} \in \omega_1, \\ k\left(\phi\right)\mathbf{C}^0 & \boldsymbol{x} \in \xi, \\ k_{min}\mathbf{C}^0 & \boldsymbol{x} \in \omega_0. \end{cases} \tag{2.40}$$

where $k_{min} \leq k\left(\phi\right) \leq 1, k\left(\phi\right) = \phi^p$, $p = 3$. The scheme is similar to the SIMP method. The evolution of the phases is governed by the Allen-Cahn equation (a reaction-diffusion equation):

$$\frac{\partial\phi}{\partial t} = \kappa\nabla^2\phi - f'\left(\phi\right), \qquad \frac{\partial\phi}{\partial\boldsymbol{n}} = 0 \text{ on } \partial\Omega \tag{2.41}$$

where κ is the diffusion coefficient and $f\left(\phi\right)$ is a double well potential function. If $f\left(\phi\right)$ is chosen to satisfy the conditions $f(0) = 0$, $f(1) = \eta\frac{\bar{J}'(\phi_t)}{\left|\bar{J}'(\phi_t)\right|}$, and $f'(0) = f'(1) = 0$, then the optimization proceeds in the direction which minimizes the design objective. Here, η is a scaling constant, chosen as 10. One such $f(\phi)$ is given by (Fig. 2.4):

$$f(\phi) = \frac{1}{4}\phi^2 (1-\phi)^2 + \eta \frac{\bar{J}'(\phi_t)}{\left|\bar{J}'(\phi_t)\right|} \left(6\phi^5 - 15\phi^4 + 10\phi^3\right) \tag{2.42}$$

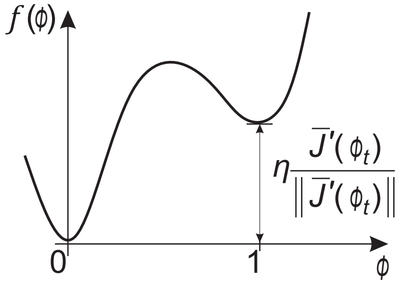


Figure 2.4: Illustration of double well potential function. Here $f(\phi)$ is the double well potential function, $\bar{J}'(\phi_t)$ represents the sensitivity of the reformulated objective function with respect to ϕ at time t and η is a scaling constant.

where $\bar{J}'(\phi_t)$ represents the sensitivity of objective function $\bar{J}$ with respect to ϕ at time t. Sensitivity analysis of objective functions, such as (1.3) and (1.4), is available in the book by Bendsøe and Sigmund [39]. Thus, the Allen-Cahn equation (2.41) reduces to:

$$\frac{\partial \phi}{\partial t} = \kappa \nabla^2 \phi + \phi(1-\phi)\left[\phi - \frac{1}{2} - 30\eta \frac{\bar{J}'(\phi_t)}{\left|\bar{J}'(\phi_t)\right|}\phi(1-\phi)\right] \tag{2.43}$$

One way to solve the Allen-Cahn equation is by using the finite difference scheme. Since an explicit finite difference scheme forces the function ϕ to diverge when $\phi \notin [0,1]$, a semi-implicit scheme is used to discretize the reaction term. Thus, the scheme to update ϕ can be written as:

$$\phi_{i,j}^{n+1} = \begin{cases} \frac{\phi_{i,j}^n + \Delta t (P_1+P_2)\kappa}{1-\left(1-\phi_{i,j}^n\right) r\left(\phi_{i,j}^n\right)\Delta t} & \text{for } r\left(\phi_{i,j}^n\right) \leq 0, \\ \frac{\phi_{i,j}^n\left(1+r\left(\phi_{i,j}^n\right)\Delta t\right) + \Delta t(P_1+P_2)\kappa}{1+\phi_{i,j}^n r\left(\phi_{i,j}^n\right)\Delta t} & \text{for } r\left(\phi_{i,j}^n\right) > 0. \end{cases} \tag{2.44}$$

where

$$r\left(\phi_{i,j}^n\right) = \phi_{i,j}^n - \frac{1}{2} - 30\eta \frac{\bar{J}'(\phi_{t1})}{\left|\bar{J}'(\phi_{t1})\right|}\phi_{i,j}^n\left(1-\phi_{i,j}^n\right)$$

$$P_1 = \frac{\phi_{i-1,j}^n - 2\phi_{i,j}^n + \phi_{i+1,j}^n}{(\Delta x)^2}, \qquad P_2 = \frac{\phi_{i,j-1}^n - 2\phi_{i,j}^n + \phi_{i,j+1}^n}{(\Delta y)^2} \tag{2.45}$$

The time step Δt satisfies Courant-Friedrichs-Lewy (CFL) condition.

2.4 Comparison among methods

Now, we discuss the results of our implementation of all the above mentioned methods. First the performance of the methods is evaluated for the two-dimensional compliance minimization test problems, followed by detailed discussions on the merits and limitations of each method. We start by looking at the implementation details of our computer codes below. It should be noted that for Challis' approach we used the code provided in her paper with a slight modification in her global stiffness matrix assembly, objective function, shape sensitivities and topological sensitivities evaluation (details are provided in section 2.3.4).

2.4.1 Implementation details

All the methods have been implemented in MATLAB and have a similar structure. The optimization algorithm consists of the following steps (c.f Fig. 2.5):

1. Initialize the design function based on the initial guess and set the value of various algorithm parameters.

2. Perform the following steps until convergence:

   (a) Compute the state variable $\boldsymbol{u}$ by solving the state equation (1.1, 1.2).

   (b) Calculate sensitivity of the objective function for the DLK level-set method and the TNK phase-field method. In the case of the AJT, WW and Challis' level-set methods, calculate the shape derivatives.

   (c) Update the topology using the respective evolution equations.

3. The level-set function needs to be reinitialized from time to time to maintain the signed distance characteristic. For this purpose, (2.15) is made use of for the AJT and DLK methods. For Challis' method, MATLAB's `bwdist` function is used. No reinitialization is performed for the WW level-set method and the TNK phase-field method.

4. Map the current level-set function into the density domain. Phase-field function, ϕ, is taken to be the same as the density function, ρ.

2.4.2 Cantilever beam optimization

We compare the performance of the methods first using the cantilever beam compliance minimization problem; which is a common benchmark problem in the literature. A Young's modulus of E = 1, and Poisson's ratio of $\nu = 0.3$, are used for all of the examples in this work and consistent units are employed. The design domain is rectangular with dimensions 2×1 and discretized using 120×60 Q4 elements. The cantilever beam is fixed on the left

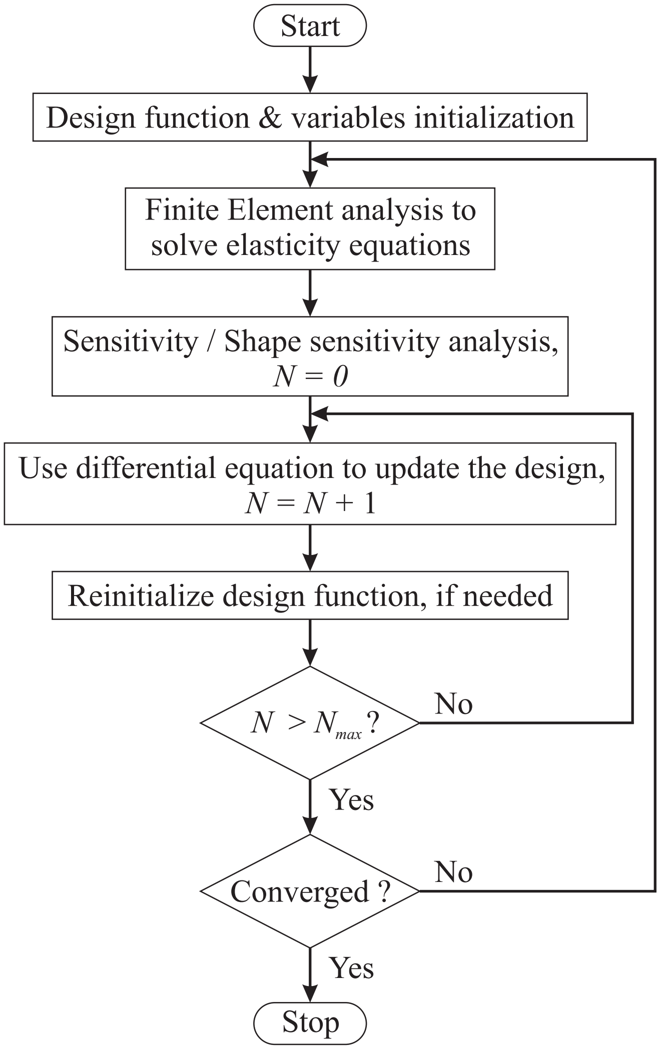


Figure 2.5: Flow chart for the differential equation-driven topology optimization. Here, N_{max} represents user defined maximum number of design updates for each finite element analysis.

side and a unit vertical load is applied at the midpoint of the right side, as shown in Fig. 2.6. The volume fraction, V_f, is fixed at 0.45 by updating the Lagrange multiplier, λ, using a scheme similar to Allaire and Pantz [10] and Takezawa et al. [208]. Optimization terminates when either the change in element densities is less than 0.1% or the change in the objective function is less than 0.01%. The densities are uniform inside each element. The time step, Δt, satisfies the CFL condition for the AJT, WW, Challis' level-set methods and the TNK phase-field method.

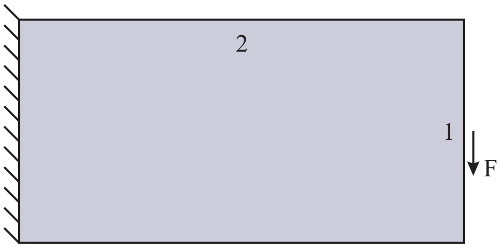


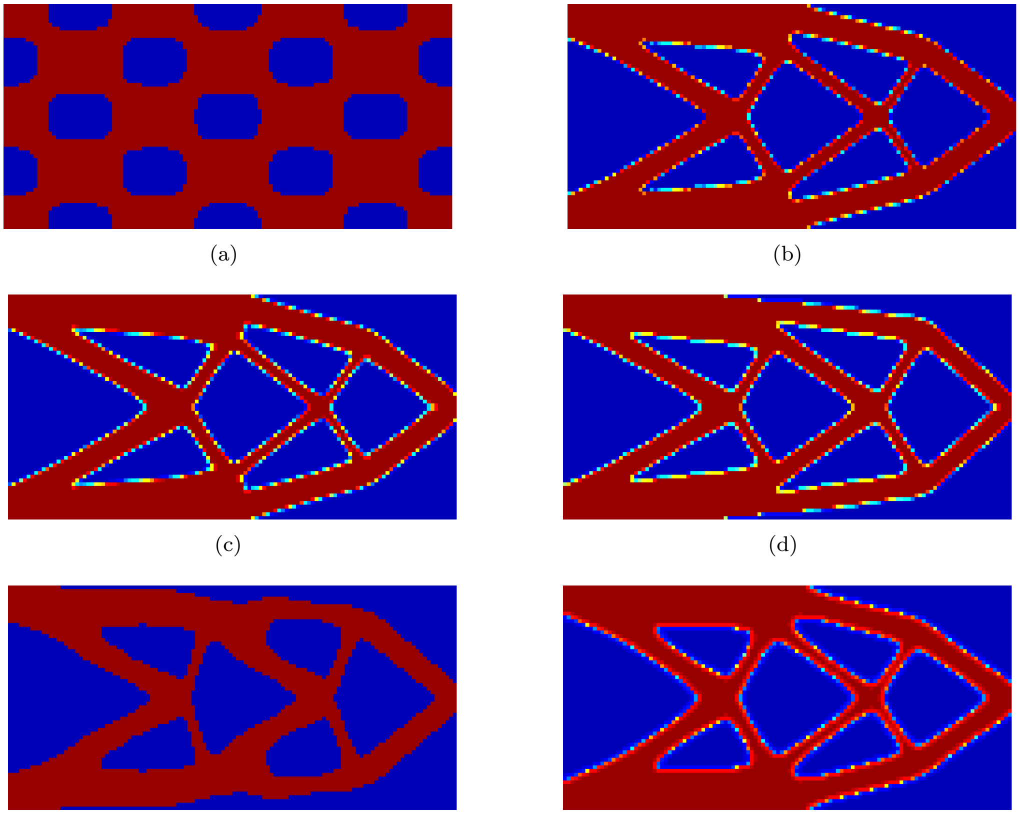
Figure 2.6: The cantilever beam with the prescribed load and boundary conditions is the first test problem.

Model parameters

For the AJT method, the level-set function is updated 20 times, using a second order upwind scheme for the Hamilton-Jacobi equation (2.9), for each elasticity analysis (1.1, 1.2). The level-set function is reinitialized after every 5 update steps of the Hamilton-Jacobi equation, by conducting five explicit time steps of the second order upwind scheme of (2.15). In our DLK level-set method implementation, the scaling parameter is chosen as $t = h_x$, where h_x is the length of one side of the finite element. Parameter t is divided into small update steps of $t_i = h_x/4$. After every update step of t_i, using (2.23), the level-set is reinitialized by conducting five explicit time steps of the second order upwind scheme of (2.15). For the WW method implementation, the shape parameter, c, is chosen as 10^{-4}, $r_{min} = 1.2$ and $\delta = 1$. We take 2 explicit time steps of the set of ODEs obtained from the Hamilton-Jacobi equation using Euler's method (2.32) for each elastic finite element analysis (1.1, 1.2). In Challis' code, the parameters `stepLength`, `numReinit` and `topWeight` [59] are chosen as 2, 4, and 2, respectively, and the design is updated using (2.36). The phase-field diffusion coefficient, κ, is taken as 2×10^{-5}. We perform 20 update step evolution equation for the phase-field function, by utilizing its semi-implicit finite difference approximate equation (2.44), for every solution of state equation (1.1, 1.2).

Results

The converged topologies for the cantilever beam problem, with starting topology Fig. 2.7(a), are shown in Fig. 2.7 and the summary of the results is given in Table 2.1. Converged configurations from the AJT level-set method (Fig. 2.7(b)), the DLK level-set method (Fig. 2.7(c)) and the TNK phase-field method (Fig. 2.7(f)) are almost identical. The DLK method produces the lowest compliance of 65.2, and all other compliances are in the range 65 to 74.



(a) (b) (c) (d) (e) (f)

Figure 2.7: Converged topologies for cantilever beam problem on a domain discretized using 120×60 mesh. (a) Initial configuration. (b) AJT level-set method. (c) DLK level-set method. (d) WW level-set method. (e) Challis' level-set code. (f) TNK phase-field method.

Method	Acronym	Update equation	Figure	Compliance (J)	Converged volume fraction
Allaire et al. [8]	AJT	(2.9)	2.7(b)	65.9	0.45
Van Dijk et al. [216]	DLK	(2.23)	2.7(c)	65.2	0.45
Wang and Wang [231]	WW	(2.32)	2.7(d)	66.8	0.45
Challis [59]	-	(2.36)	2.7(e)	73.8	0.45
Takezawa et al. [208]	TNK	(2.44)	2.7(f)	71.6	0.45

Table 2.1: Summary of results shown in Fig. 2.7 for the cantilever beam problem. Volume fraction, $V_f = 0.45$, is imposed on the optimization problem.

One of the drawbacks of the standard differential equation-driven methods is that new holes cannot nucleate within a structure because there is no built-in hole nucleation mechanism. However, the WW approach [231] and Challis' approach [59] do claim to generate holes. Hole nucleation capability may alleviate the dependence on the initial guess which we shall examine next.

Influence of initial guess

Typically, in differential equation driven methods, the optimal design is mesh dependent, i.e. it depends on the resolution of the design parameterization. If the design parameterization allows smaller details, the optimal design will be topologically more complex and finer members will be formed. To obtain mesh independent results, some type of length scale needs to be enforced (similar to density-based topology optimization methods [39]). None of the five methods studied in this work enforce any length scale control. Thus, the optimal designs will have topological complexity dependent on the initial guess, unless some form of hole nucleation mechanism is introduced. This can be easily tested by evaluating the performance of the five methods for a different initial configuration. For all the methods, the design parameters are kept the same as before. The results are shown in Fig. 2.8.

In spite of the inclusion of hole nucleation capabilities, we observe that the WW and Challis' methods do not alleviate the dependence on the initial guess. The converged topology from Challis' approach (Fig. 2.8(e)) is still similar to other methods which can not generate holes. Although the final configuration obtained from the WW approach (Fig. 2.8(d)) is topologically more complex than the others (has more holes than the initial starting configuration), it is still not the same as the result obtained in the last problem (Fig. 2.7(d)). Further discussion on hole nucleation capabilities and other features of the WW and Challis' methods will follow in subsequent sections.

2.4.3 Bridge with holes optimization

Next, we explore the problem of a bridge with holes. The design domain is rectangular in shape, discretized using 120 × 60 Q4 elements and is simply supported, as shown in Fig. 2.9. Two fixed holes are introduced into the design domain and taken care of during optimization using passive elements [39]. All other parameters are kept the same as in the previous cantilever beam problem.

Figure 2.10(a) is chosen as the starting topology. The converged topologies are shown in Fig. 2.10. Converged configurations from Challis' level-set method (Fig. 2.10(e)) and the TNK phase-field method (Fig. 2.10(f)) are visually similar, with the same number of members and similar member orientations. The AJT level-set method topology (Fig. 2.10(b)) also has the same number of members as the TNK method but with a different

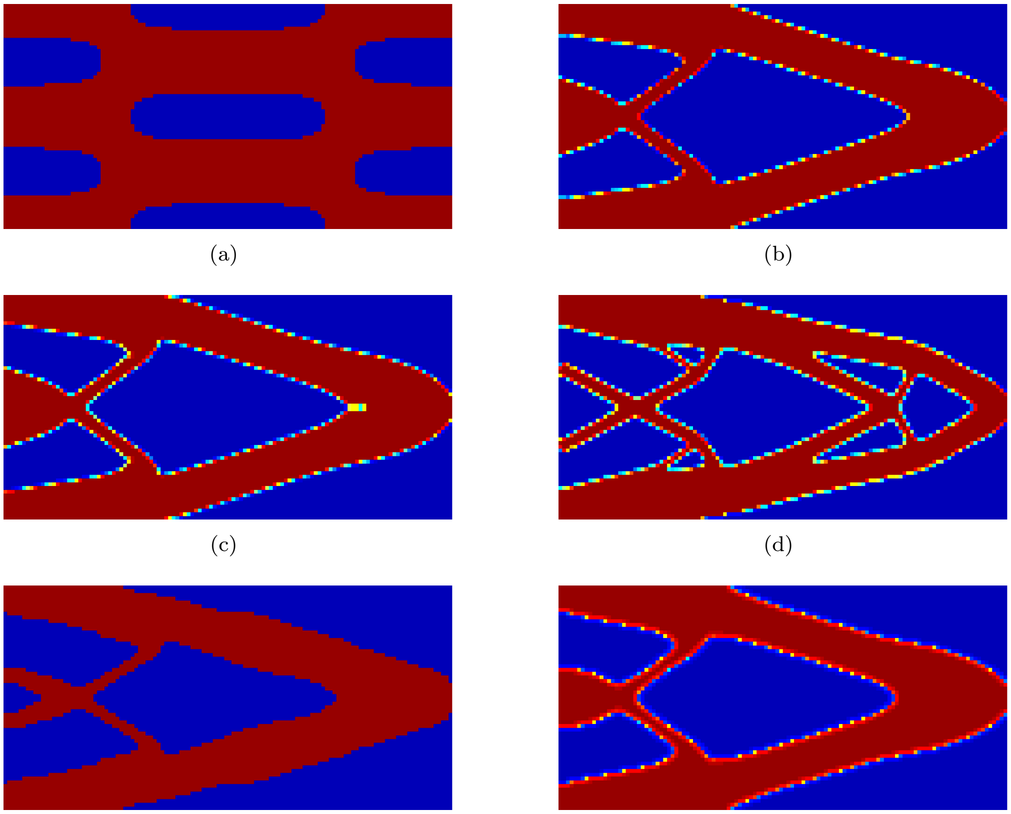
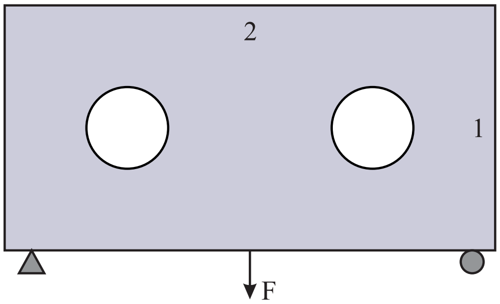

(a) (b) (c) (d) (e) (f)

Figure 2.8: Another initial configuration mesh for cantilever beam problem. (a) Initial topology. (b) AJT ($J = 76.2$). (c) DLK ($J = 75.6$). (d) WW ($J = 72.3$). (e) Challis' ($J = 80.7$). (f) TNK ($J = 80.4$).

Figure 2.9: The bridge problem with holes in the design domain.



orientation of the middle two members. The results of the DLK method (Fig. 2.10(c)) and the WW level-set method (Fig. 2.10(d)) are clearly different from the others. In the DLK method, there are traces of intermediate densities right next to the fixed circular holes which is undesirable. The DLK method produces the least compliance of 18.5, and compliances for the other methods are fairly close to each other (in the range 18 to 22). In the following sections, we examine each method individually.

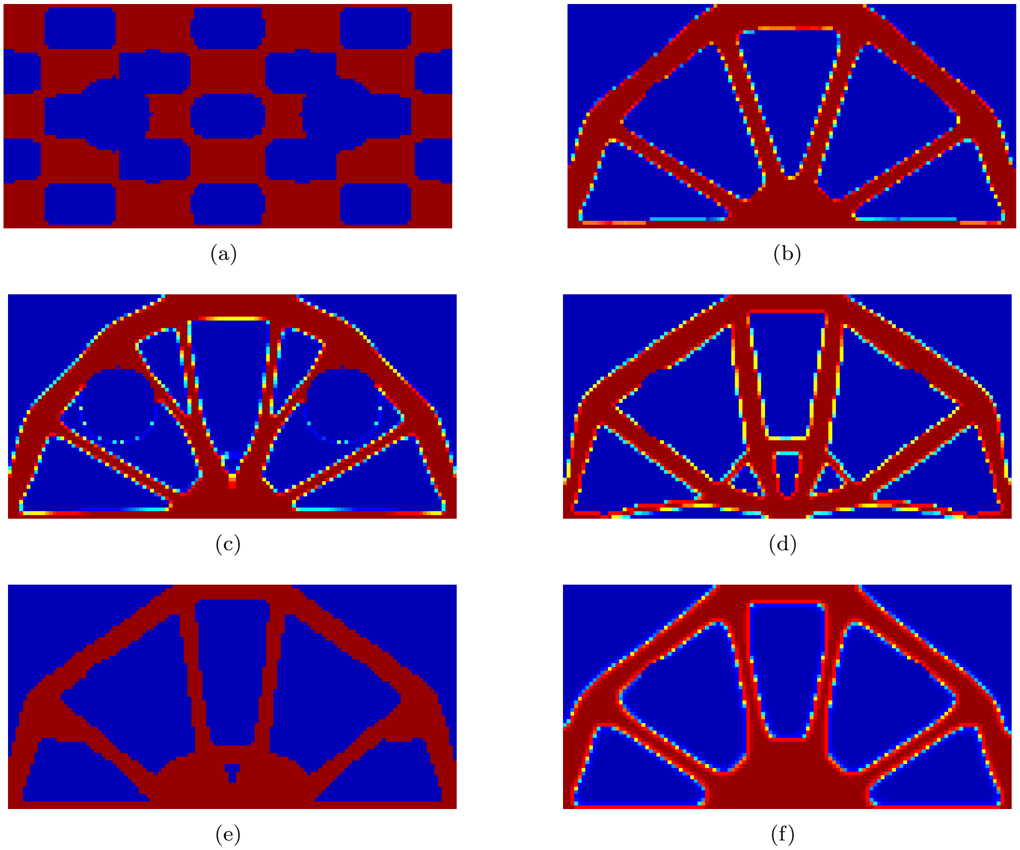

(a) (b) (c) (d) (e) (f)

Figure 2.10: Converged topologies for bridge with holes problem on a domain discretized using 120 × 60 mesh. (a) Initial configuration. (b) AJT ($J = 18.6$). (c) DLK ($J = 18.5$). (d) WW ($J = 20.1$). (e) Challis' ($J = 20.5$). (f) TNK ($J = 21.4$).

2.4.4 Discussion on AJT level-set method

The ability to handle the merging/cancellation of holes makes the AJT level-set method suitable for topology optimization problems. As mentioned before, in the AJT method, the lack of length scale control, and an inherent hole nucleation mechanism, results in designs with topological complexity dependent on the initial guess. For such a method, it is important that the results should at least be invariant to mesh refinement when the optimization starts from a similar topology i.e. starting from a similar initial topology, the method should produce designs of similar topological complexity for different mesh discretizations. We test the invariance of the AJT method with respect to mesh refinement using the cantilever beam problem (Fig. 2.6) for the initialization shown in Fig. 2.7(a). The Lagrangian multiplier is kept constant at 25. The resulting converged designs, shown in Fig. 2.11, confirm that the current method is invariant to mesh refinement for similar starting topology. The compliance for the different mesh discretizations are also similar (65.8, 65.9, 66.0 and 66.2 for 80 × 40, 100 × 50, 160 × 80 and 200 × 100 mesh discretizations, respectively).

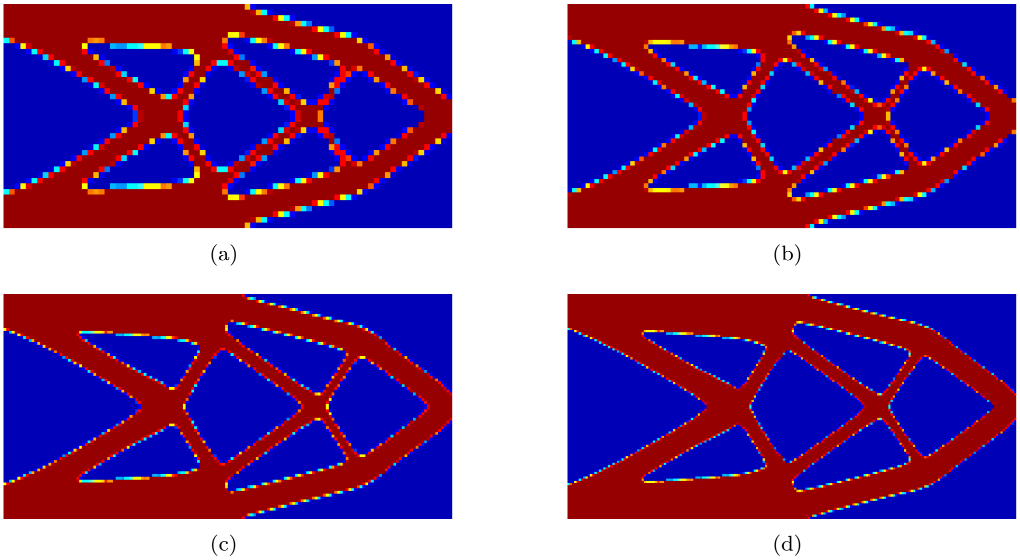


Figure 2.11: Study of invariance of the AJT level-set approach to mesh refinement when optimization starts from similar topology of Fig. 2.7(a). Converged configurations for mesh discretizations of (a) 80×40 ($J = 65.8$), (b) 100×50 ($J = 65.9$), (c) 160×80 ($J = 66.0$), and (d) 200×100 ($J = 66.2$).

In their approach, Allaire et al. update the level-set 20 times using the Hamilton-Jacobi equation (2.7) for every solution of the state equation. Although the scheme is justified, since one explicit update step (2.7) is much cheaper than the solution of state equation (1.1, 1.2), estimating the appropriate number of update steps per state equation solution is a vital and difficult task. A high number of steps results in an inaccurate design and a small number of steps results in a slow rate of convergence. A physically meaningful, perhaps adaptive, scheme needs to be devised to estimate the optimum number of update steps of the Hamilton-Jacobi equation (2.7) to achieve a balance between faster convergence and accurate results. It should be noted that Allaire et al. reduce the number of update steps during the course of optimization if the objective function $\bar{J}$ is not decreasing, but the exact implementation details are not provided. Similar arguments can be made for the appropriate frequency of reinitialization and the number of explicit time steps of (2.15) that need to be performed to maintain the signed distance nature of the level-set function. If the reinitialization equation (2.15) is solved fully until the level-set function does not change anymore, or if the reinitialization is performed very frequently, then the algorithm converges slowly. On the contrary, if insufficient reinitialization is performed, then the algorithm tends to become unstable, leading to an inaccurate estimation of the boundary normal, and ultimately converging to suboptimal topologies.

Since there are no inherent hole nucleation mechanisms in this method, the converged topology is dependent on the initial chosen topology. Allaire et al. [4] proposed a remedy to this problem using the topological gradient method or bubble method [80, 189, 57, 92]. In their method, the authors [4] use the topological gradient, at a predefined frequency, to nucleate holes in the domain. For hole nucleation, the sign of the level-set function is changed from negative to positive in the regions where the topological derivative attains minimum negative values. As is evident from the work of Allaire et al. [4], their topological derivative approach is effective in nucleating holes, but it still does not fully alleviate the dependence on the initial guess. Again, the frequency at which one uses topological derivatives to nucleate holes is chosen heuristically. If they are employed too often, the domain becomes highly irregular, and if employed sparingly, the algorithm may have already converged to a local minimum which might not be desirable. Some guidelines on choosing the parameters, discussed above, are provided by Allaire et al. [9].

2.4.5 Discussion on DLK level-set method

In the test problem, we have seen that the DLK approach; which uses sensitivities of the objective function, instead of shape sensitivities, to update the level-set, produces similar configurations when compared to other level-set methods. This method is incapable of generating new holes as other level-set methods, so the converged topology heavily depends on the initial starting configuration. Topological derivatives can be used (discussed before for the AJT level-set approach), to nucleate holes in the design domain and alleviate the dependence on the initial guess to a certain extent. We do not study hole nucleation for the DLK approach in the current work. Here, we first investigate if the DLK level-set approach is invariant to mesh refinement for a similar starting topology. We solve the cantilever beam problem starting with the initial topology, seen in 2.7(a), for mesh discretizations of 80×40, 100×50, 160×80 and 200×100. Figures 2.12(a)-(d) indicate that the DLK approach produces consistent results for various mesh discretizations and thus it is invariant to mesh refinement when optimization starts from a similar topology.

In the DLK level-set approach, the scaling parameter t is chosen as $h_x/2$ (constrained problem) and the subdivided time step is taken as $t_i = h_x/20$, where h_x is the length of the side of the finite element. In this work, as mentioned before, we chose a slightly larger scaling parameter $t = h_x$ and subdivided time step $t_i = h_x/4$ to speed up the convergence. The convergence is affected if a bigger scaling parameter is chosen. Limits on the scaling

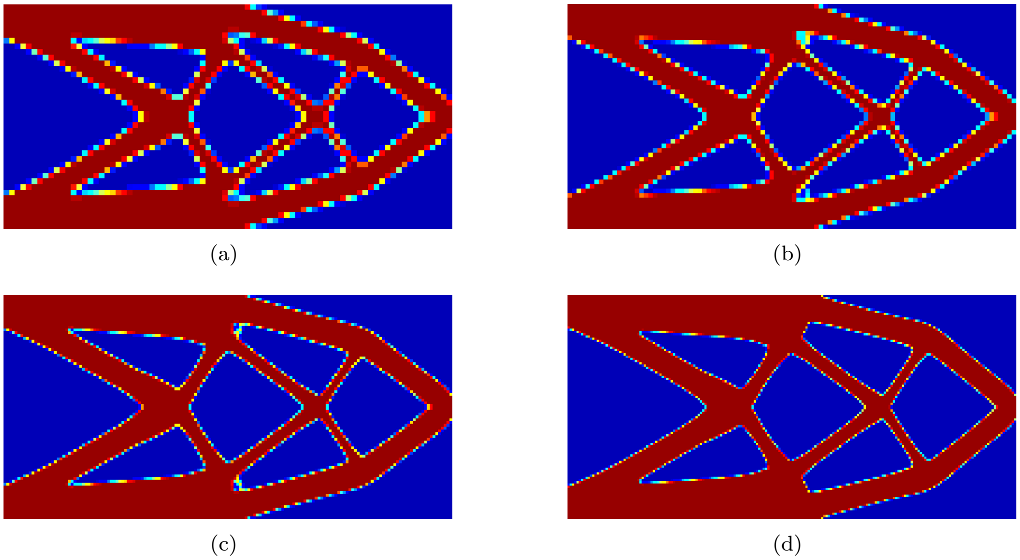

(a)
(b)
(c)
(d)

Figure 2.12: Study of invariance of the DLK level-set approach to mesh refinement when optimization starts from similar topology. Initial topology is chosen same as Fig. 2.7(a). Converged configurations for mesh discretizations of (a) 80×40 ($J = 64.6$), (b) 100×50 ($J = 65.0$), (c) 160×80 ($J = 65.5$), and (d) 200×100 ($J = 65.9$).

parameter t should be provided, similar to the CFL condition, but are not shown in their work.

In this work, the level-set function needs to be reinitialized after each update step; the reason being, that the steepest-descent type updates (2.23) do not preserve the gradient of the level-set function. The level-set function slope, near the design boundaries, deviates faster from unity in the DLK method than in the AJT method, which necessitates more periodic reinitialization to achieve numerical convergence. Now, consider (2.17) and (2.18). By some algebraic manipulations, we can see that:

$$\frac{\partial \bar{J}}{\partial \phi_i} = -\left(\boldsymbol{\varepsilon}(\boldsymbol{u})^T \mathbf{C}^0 \boldsymbol{\varepsilon}(\boldsymbol{u}) - \lambda\right) \frac{\partial \rho_e}{\partial \phi_i} \tag{2.46}$$

Also, from (2.21) and (2.22), it is clear that $\frac{\partial \rho_e}{\partial \phi_i} \leq 0$. Let, $\zeta = -\frac{\partial \rho_e}{\partial \phi_i} \geq 0$; thus, (2.23) can be simplified as:

$$\frac{\phi_i^{n+1} - \phi_i^n}{t} \approx -\bar{\zeta} v_1 \tag{2.47}$$

where $v_1 = \left(\boldsymbol{\varepsilon}(\boldsymbol{u})^T \mathbf{C}^0 \boldsymbol{\varepsilon}(\boldsymbol{u}) - \lambda\right)$ and $\bar{\zeta} = \zeta / \left|\frac{\partial \bar{J}}{\partial \phi_i}\right| \geq 0,\ \bar{\zeta} \in [0, 1]$. When compared with the discrete form of the Hamilton-Jacobi equation used in the AJT method (2.7), i.e.,

$$\frac{\phi_i^{n+1} - \phi_i^n}{t} \approx -v |\nabla \phi_n| \tag{2.48}$$

we see that (2.47) and (2.48) are similar in nature. For both methods, the level-set function propagates in the direction in which the strain energy is minimized.

In this method, the bandwidth of the approximate Heaviside function h is chosen as $d/10$, where d is the length of the diagonal of the finite element. No justification is provided for this choice. Ideally h should be as close to zero as possible (to accurately map the level-set function to the density domain), but h cannot be exactly zero because it will result in infinite gradients (c.f. (2.21) and (2.22)). So a scheme needs to be devised to obtain the optimal value of h. Finally, we would like to mention that; since, in this work, we have limited ourselves to compliance minimization problems, we did not investigate the compliant mechanism problem, treated as a multiple constraint optimization problem by applying displacement constraints, in Van Dijk et al.'s work. Effective treatment of multiple constraints is one of the motivations behind Van Dijk et al.'s work. To demonstrate that, Van Dijk et al. [216] solved the force inverter problem as a multiple constraint problem. Their result for the constrained optimization problem indicates the presence of single node connection and their results for both constrained and unconstrained problems contain tiny voids inside the solid region. This shows that the optimization converges to local minimums, which can be avoided by using regularization algorithms such as filtering. So far Van Dijk et al. have studied the compliance minimization problem and the compliant mechanism problem as a multiple constraint problem. Their method needs to be investigated for other optimization problems such as non-linear elasticity and design dependent load problems, to assess its robustness.

Recently, Van Dijk et al. [217] published an updated version of their previous method (studied in this work). In their current research, the authors persist with a steepest-descent type updating scheme for the level-set function and look at multiple constraint problems. There are two contrasting differences between the two approaches. First, in the updated version, no reinitialization is used. Second, an exact Heaviside function is used to relate the level-set function and the element densities. It is known that, without some kind of reinitialization or regularization, the level-set function values drift to large absolute values. In order to obtain accurate predicted responses, the authors impose an upper bound on the steepest descent update step for the level-sets and also impose a limit on the size of density change for each level-set update step, both of which are heuristic in nature. Additionally, they utilize heuristic diagonal preconditioner to obtain uniform level-set increments. To deal with the integration of the Heaviside function and its derivatives, in order to obtain the element densities and their sensitivities, the authors define a piecewise linear shape function for Q4 finite elements. Their results show non-physical gray region (compliance minimization results in [217]) and single node connections (compliant mechanism results in Van Dijk et al. [217]), even though good convergence is obtained for the imposed multiple constraints. There are also undesirable oscillations in the convergence history. The authors argue that consistent sensitivity analysis and update scheme lead to good convergence of multi-constrained optimization problem but may also result in numerical artifacts such as excessive gray region and point hinges (similar to density-based topology optimization methods [39]). The authors tried administering perimeter constraint and density filters to avoid the aforementioned numerical anomalies. Effective treatment of multiple constraints is an important issue which needs to be addressed by the level-set community in the future.

2.4.6 Discussion on WW level-set method

Following the same trend as the previous methods, we first check if the WW method is invariant to mesh refinement for a similar starting topology. We solved the cantilever beam problem, using the initial guess in Fig. 2.7(a), for mesh discretizations of 80×40 and 100×50. The result for the 120×60 mesh is already shown in Fig. 2.7(d). We could not run the problem for meshes larger than 120×60, for reasons which will be discussed later in this section. Results (Figs. 2.13(a)-(b) and Fig. 2.7(d)) suggest that the WW level-set method is also invariant to mesh refinement when the optimization algorithm starts from a similar topology.

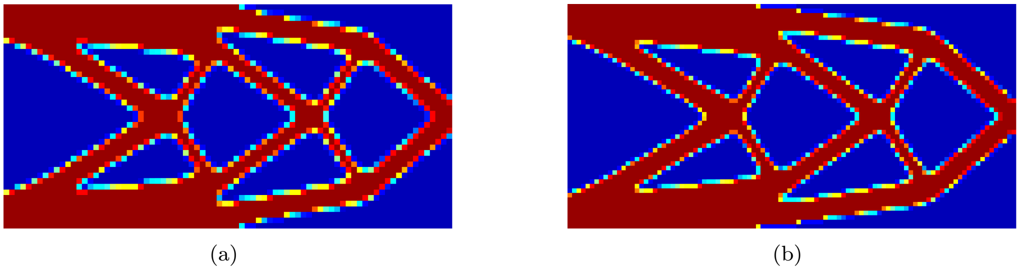
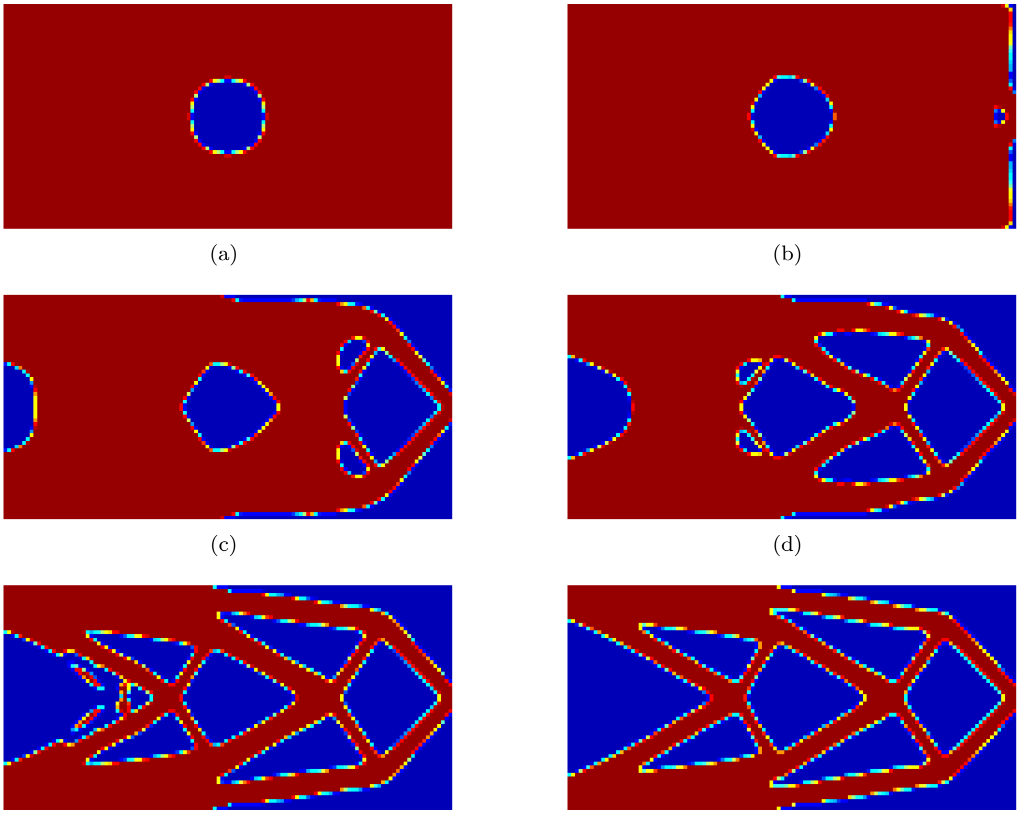


Figure 2.13: Study of invariance of the WW level-set approach to mesh refinement for similar starting topology. Figure 2.7(a) is the chosen starting topology. Converged configurations for mesh discretizations of (a) 80×40 ($J = 65.8$) and (b) 100×50 ($J = 66.3$).

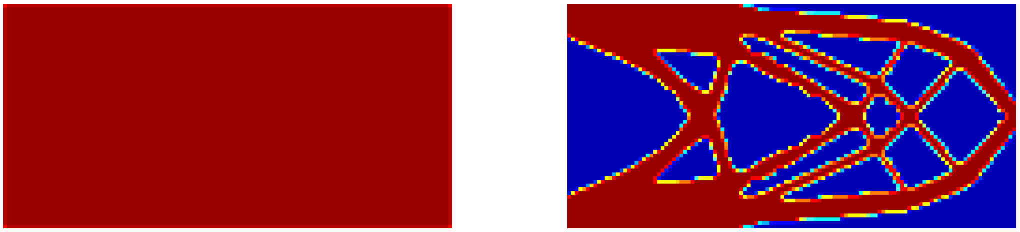
To investigate the issue of sensitivity of the final topology to the initial chosen configuration, we ran the cantilever beam problem with two different initial configurations (Figs. 2.14(a) and 2.15(a)). The corresponding converged topologies (Figs. 2.14(f) and 2.15(b)) support the author's claim of hole nucleation; however, lack of a unique converged topology (c.f. Figs. 2.7(d), 2.8(d), 2.14(f) and 2.15(b)) for various initial guesses confirms that dependence on the initial guess has not been fully resolved. The method still converges to local minimums, dictated by the starting topology.

In the present approach, Wang and Wang [231] used radial basis functions to parameterize the level-set function. Our implementation of the WW approach suggests that although the infinite smoothness of MQ splines helps maintain smoothness of $\boldsymbol{N}^T(\boldsymbol{x})$ (c.f. (2.29)), the



(a) (b) (c) (d) (e) (f)

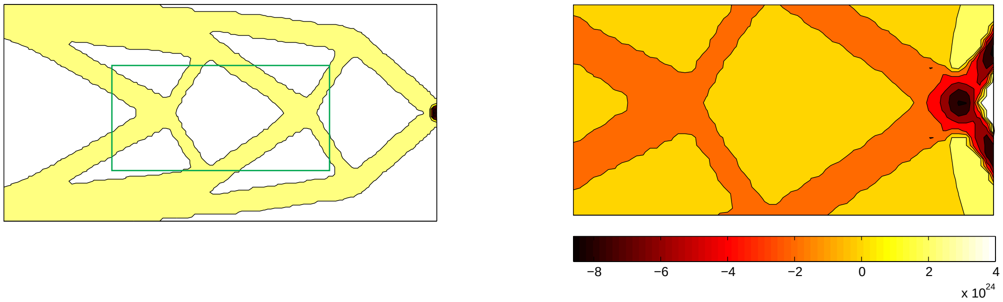
Figure 2.14: Evolution of topology for curved cantilever beam problem starting with initial topology (a) for the WW approach on a 120 × 60 mesh. (b) Iteration 56. (c) Iteration 80. (d) Iteration 85. (e) Iteration 92. (f) Converged topology.



(a) (b)

Figure 2.15: Different initial topology for the WW approach on a 120 × 60 mesh. (a) Initial guess. (b) Converged configuration.

level-set function grows to very high absolute values because of the rapid growth in $\boldsymbol{\alpha}(t)$. To observe this, consider Fig. 2.16; which shows the level-set function for the converged configuration in Fig. 2.7(d). Towards the end of optimization, at some locations across the design boundaries, the level-set function (Fig. 2.16(b)) varies in the order of $+10^{24}$ to -10^{24} (much higher at some other locations). Such unbounded growth and high variation implies that the boundary is too steep (hence high gradients) and, thus, obtaining an accurate approximation of the normal at the design interface is difficult. This might cause the algorithm to converge to incorrect results. Smooth radial basis functions do alleviate the problem of high gradients to a certain extent, by providing some sort of smoothness to the level-set function, without which, the algorithm does not converge.



(a) (b)

Figure 2.16: Level-set function corresponding to the converged topology in Fig. 2.7(d). (a) Level-set function. (b) Zoomed in section.

In the WW approach, a new scheme is introduced to extend the advection velocity field, defined on the front using shape derivatives, to the entire design domain. In order to provide a physically meaningful extension velocity, they assumed a strain field $\boldsymbol{\varepsilon}(\boldsymbol{u}) = 0$, resulting in a strain energy of $\boldsymbol{\varepsilon}^T\mathbf{C}\boldsymbol{\varepsilon} = 0$, for $\phi(\boldsymbol{x}) > \delta$. This assumption adds little value when we are using an Ersatz material approach, since in the region $\phi > 0$ (void region) we assume $\mathbf{C} \approx 0$ which results in a strain energy of $\boldsymbol{\varepsilon}^T\mathbf{C}\boldsymbol{\varepsilon} \approx 0$. The authors also smooth the advection velocity near the design interface, citing the reason that the advection velocity is discontinuous at the boundaries. It should be noted that the advection velocity is C^0 continuous (c.f. (2.33)). The smoothing of the advection velocity helps bring down the gradients to a certain extent, but, high inaccurate gradients persist in some regions, especially the region around the point of application of the load. Such high values of level-set function can be attributed to the fact that the WW method does not utilize any regularization schemes (for example, reinitialization), other than velocity smoothing, to control the gradients of the level-set function near the design boundaries. When we tried to periodically reinitialize the level-set function in the WW scheme, we were able to control the magnitude of the gradients (as expected) but it resulted in the loss of hole nucleation capability.

c	Compliance (J)	Converged volume fraction	Condition number of $\boldsymbol{H}$ (c.f. (2.26), (2.27))	Iterations
10^{-5}	66.4	0.45	2.67×10^5	93
10^{-4}	65.8	0.45	2.70×10^5	147
10^{-3}	65.8	0.45	2.94×10^5	212
10^{-2}	65.7	0.45	7.97×10^5	298
10^{-1}	-	-	1.06×10^{12}	No convergence

Table 2.2: Influence of shape parameter, c, on the WW level-set method for 80×40 mesh.

Wang and Wang [231] argued that reinitialization is computationally expensive and time consuming, which is true to a certain extent, but this cost is minimal compared to the solution of the Hamilton-Jacobi equation using multi-quadratic radial basis functions. Multi-quadratic splines produce a dense interpolation matrix $\boldsymbol{H}$ which is known to be ill-conditioned. So, the design update (2.32) is very expensive since it requires inversion of a dense matrix. We could run examples only up to a mesh discretization of 120×60; which converged within a reasonable time frame. Finer meshes either take too long to converge (more than a few hours) or the system runs out of memory.[2] Many techniques are available to efficiently handle dense, ill-conditioned matrices such as the domain decomposition method, fast multi-pole method and pre-conditioning [51]. In their work, in order to speed up the convergence Wang et al. chose time steps of 10^{-3} or 10^{-4} which is much larger than the CFL time steps (the CFL time step is of the order of 10^{-5}, 10^{-6} for 40×20, 120×60 meshes, respectively). Although rapid convergence can be achieved by using bigger time steps, it creates instability in the system due to large accumulated errors.

Finally, we also investigated the influence of shape parameter c, which controls the flatness of the radial basis function, on the optimization algorithm. We varied c in the range $10^{-5} - 10^{-1}$ for mesh discretization of 80×40 and the initial guess shown in Fig. 2.7(a). Our study (Table 2.2) shows that the condition number of $\boldsymbol{H}$ increases with c, indicating an increase in instability in the system. The compliances are approximately the same with no discernible trend. Visually, the converged topologies are all similar to the test problem solution for the WW method (Fig. 2.7(d)). We did observe that for $c = 10^{-1}$, the condition number of $\boldsymbol{H}$ becomes too high and thus the algorithm fails to convergence. This sets an upper limit on the choice of c. Similar results were observed for other mesh discretizations.

---

[2] All the numerical problems were performed on Intel(R) Core 2 Quad, 2.49 GHz processor and 8 GB RAM running MATLAB R2009a.

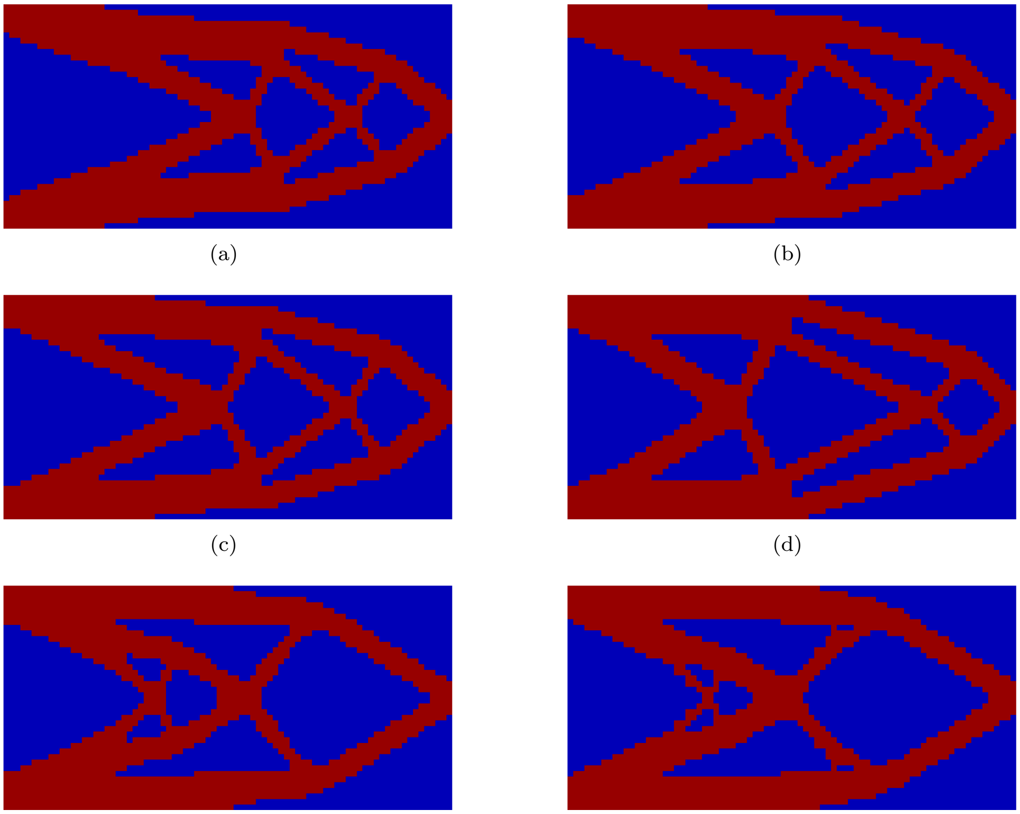
2.4.7 Discussion on Challis' level-set method

In its current form, Challis' algorithm possesses some limitations. Her implementation takes a long time to converge if the mesh discretization is greater than 5000 elements. Because her original code was developed for educational purpose, justifiably little effort was made on the part of the author to make it more efficient. We have alleviated this problem by using sparse matrix assembly as mentioned before. Similar to the AJT method, in Challis' implementation, it is difficult to estimate the optimum number of Hamilton-Jacobi equation update steps that need to be performed for every finite element update. There is a trade off between accuracy and computational time. Challis uses the MATLAB function `bwdist` for reinitialization which excessively smooths the level-set function. It is also not possible to control the degree of smoothness with `bwdist`. Other approaches, such as the one used in the AJT method, where (2.15) is used for reinitialization, although computationally more expensive, can provide the desired control over smoothness. As indicated earlier, in Challis' approach the traditional Hamilton-Jacobi equation (2.7) is modified to include topological derivatives which aid in the generation of holes. Frequency of reinitialization is a significant issue with regards to hole nucleation. In Challis' method, reinitializing the level-set function too often neutralizes the effect of topological derivatives and thus, the hole generation capability is lost. Therefore, it is vital that a suitable reinitialization frequency is established, which not only keeps the level-set function gradients near the design boundaries under control, but also does not hamper hole nucleation.

In Challis' approach, positive constant w controls the influence of topological derivatives. We next examine the influence of w on the optimization using the cantilever beam problem Fig. 2.6. Parameter w is varied in the set $\{2, 3, 4, 6, 8, 10\}$. The domain discretization is fixed to 80×40 and the other parameters are kept constant ($V_f = \texttt{volRec} = 0.45$, $\texttt{stepLength} = 2$, $\texttt{numReinit} = 4$). It can be seen from Fig. 2.17 that parameter w significantly influences the final topology. For $w = \{2, 3, 4\}$, the topologies are visually similar.

The parameter w is also mesh dependent and the next study confirms this statement. We keep w constant at 3, all other parameters are kept the same as they were in the last study, and we vary the mesh discretization. We observe that the final configurations are different for different mesh discretizations (Fig. 2.18). These two studies indicate that w significantly influences the optimization algorithm. A smaller value prevents new holes from nucleating and a larger value causes the topological derivatives to dominate, producing too many holes. More work needs to be carried out to arrive at an optimal value for w.

In this approach, shape and topological sensitivities are smoothed after each state equation solution over the entire domain. Explicit sensitivity smoothing is typically employed to reduce the probability of convergence to a local minimum. But, care must be taken when



(a) (b)

(c) (d)

(e) (f)

Figure 2.17: Study of the influence of w on Challis' approach using cantilever beam problem on a 80×40 mesh. Initial configuration is a fully solid domain. Converged configurations for (a) $w = 2$ ($J = 72.1$), (b) $w = 3$ ($J = 68.6$), (c) $w = 4$ ($J = 70.0$), (d) $w = 6$ ($J = 68.4$), (e) $w = 8$ ($J = 71.2$), and (f) $w = 10$ ($J = 70.6$).

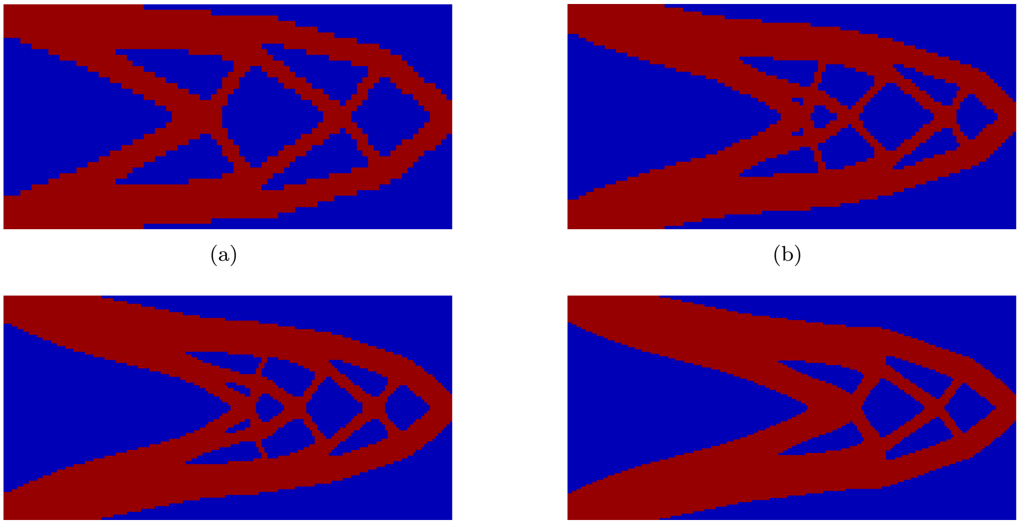

(a) (b) (c) (d)

Figure 2.18: Mesh refinement study with $w = 3$ for Challis' approach. Initial configuration is a fully solid domain. Converged configurations on mesh discretization of (a) 80×40 ($J = 68.6$), (b) 120×60 ($J = 72.7$), (c) 160×80 ($J = 73.0$), (d) 200×100 ($J = 73.2$).

applying such smoothing techniques, because it affects the hole nucleation process and also the rate of convergence of the algorithm. In Challis' approach, a border of void elements around the design domain is included. The author claimed that they are necessary to represent the boundary of the structure accurately; which actually aids the algorithm to converge. This approach has been applied to compliance minimization problems and needs to be tested for other problems such as compliant mechanisms, design dependent loads and non-linear elasticity problems, for robustness.

2.4.8 Discussion on TNK phase-field method

The phase-field method by Takezawa et al. [208] utilizes a time dependent reaction-diffusion equation, known as the Allen-Cahn equation, for the evolution of topologies. The sensitivity of the objective function, employed extensively in density methods, is used to define a double well potential function which is a part of the evolution equation. Unlike level-set methods, the phase-field method eliminates the need for reinitialization, which can be costly. This method has been verified for minimum compliance, compliant mechanism and eigenfrequency maximization problems. Problems such as nonlinear elasticity, design dependent loads and minimum stress are yet to be explored.

The phase-field method performed well for the test problem shown earlier and produced expected topologies. As shown in the cantilever beam problem, like level-set methods, the phase-field method does not have an embedded hole generation mechanism and thus the final topologies are influenced by the initial topology. Takezawa et al. suggest the use of topological derivatives to alleviate this problem which is similar to the approach in level-set methods [4]. We did not investigate the use of topological derivatives in the phase-field method to generate holes. From Takezawa et al.'s work [208], it is evident that the use of topological derivatives is a feasible option to nucleate holes. However, further work is required to verify the accuracy of this approach and more test problems need to be investigated to establish its robustness.

The diffusion coefficient, κ, plays a critical role in the phase-field method. We investigate the effects of κ on the optimization process using the cantilever beam problem. For the first study, the domain discretization is fixed at 160×80 elements; a volume fraction of 0.45 is prescribed and κ is varied. Figure 2.19(a) is chosen as the initial configuration. Optimization is performed for $\kappa = 0.5 \times 10^{-5}$, 2×10^{-5} and 5×10^{-5}. From Fig. 2.19(b)-(d), it is evident that as κ increases, the thickness of the diffuse interface increases. The percentage of total elements in the diffuse interface (Fourth column in Table 2.3), gives a general idea about the diffuse interface thickness. Design boundary becomes increasingly fuzzy. Also, the number of holes decreases with increasing κ. Thus, κ also affects the perimeter of the final configuration.

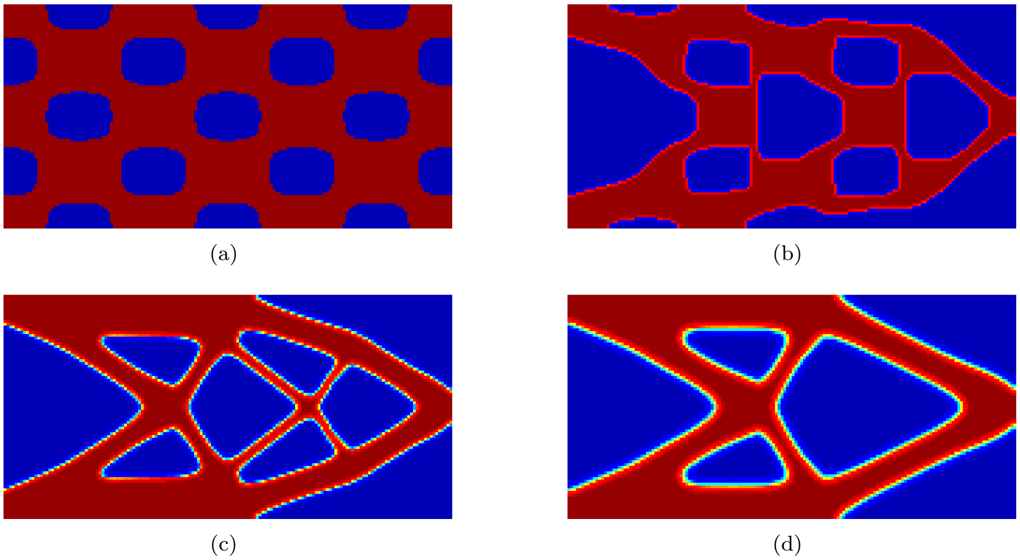

(a) (b) (c) (d)

Figure 2.19: Study of the influence of diffusion coefficients κ on the TNK phase-field approach. Mesh discretization is 160×80 and (a) is the starting configuration. Converged configurations for (b) $\kappa = 0.5\times10^{-5}$, (c) $\kappa = 2 \times 10^{-5}$, and (d) $\kappa = 5 \times 10^{-5}$.

For the next study, we fix κ to 1×10^{-5} and vary the mesh discretizations. The initial guess is kept the same as in the previous study. We see that for a particular κ (1×10^{-5} in this case), too coarse of a mesh discretization (Figs. 2.20(a) and (b)) hinders the optimization and the algorithm stops prematurely, which is evident from the final volume fractions in Table 2.4. Finer discretizations in the range 160×80 and 200×100 (Fig. 2.20(c), (d)) seem

κ	Compliance (J)	Converged volume fraction	Elements in the diffuse interface ($0.01 < \phi < 0.99$)
0.5×10^{-5}	87.6	0.45	13.8 %
2.0×10^{-5}	71.9	0.45	27.6 %
5.0×10^{-5}	74.7	0.45	33.0 %

Table 2.3: Influence of diffusion coefficient, κ, on the TNK phase-field method for a fixed 160×80 mesh and $V_f = 0.45$ (c.f. Fig. 2.19).

to be more ideal for $\kappa = 1 \times 10^{-5}$ (least compliance).

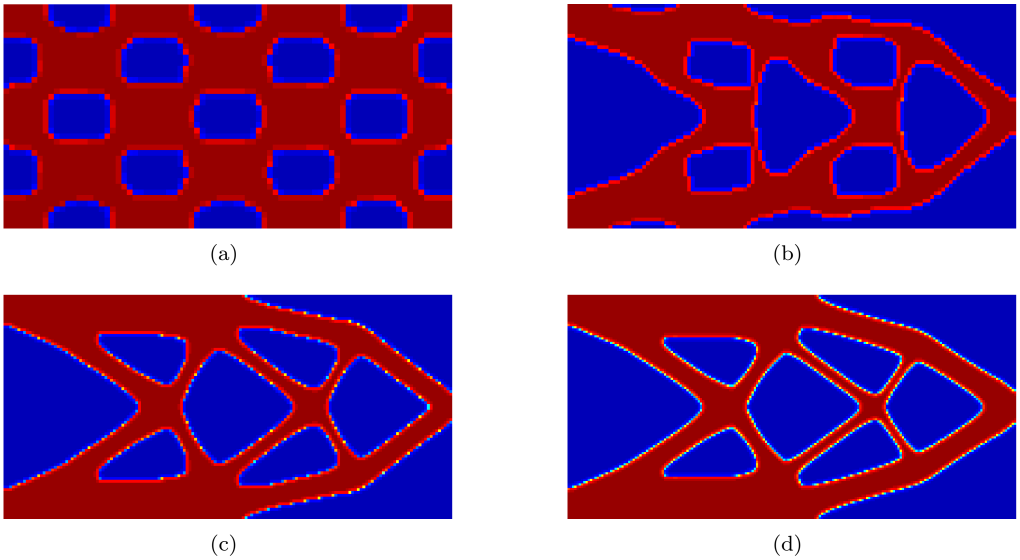

(a)
(b)
(c)
(d)

Figure 2.20: Cantilever beam problem solved on different mesh discretizations with diffusion coefficient $\kappa = 1 \times 10^{-5}$. Figure 2.19(a) is choosen as the initial guess. Converged topologies for mesh discretizations of (a) 80×40, (b) 120×60, (c) 160×80, and (d) 200×100.

From these two studies it is clear that the choice of κ is mesh dependent. It not only controls the thickness of the diffuse interface but also the number of holes, thus perimeter, of the final configuration. If κ is too small the resulting thickness of the diffuse interface is too small and thus the evolution of the topology is hindered. On the other hand, if κ is too big, then there is excess gray region and fewer holes in the converged topology(Fig. 2.21). Our numerical experimentation shows that the ideal choice of κ is one which produces topologies with approximately three-to-four elements in the diffuse interface.

To conclude, the diffusion coefficient, κ, needs to be scaled appropriately, when the mesh discretization is varied, to get consistent designs. It should also be noted that, in the interpolation function (2.40) (same as SIMP model), the power p also affects the gray region. A penalty, p, greater than or equal to 3 is recommended in the SIMP model.

Mesh discretization	Compliance (J)	Converged volume fraction
80×40	123.9	0.66
120×60	78.5	0.46
160×80	70.7	0.45
200×100	70.6	0.45

Table 2.4: Study of affect of diffusion coefficient ($\kappa = 1 \times 10^{-5}$) with variations in mesh discretization on the TNK phase-field method (c.f. Fig. 2.20). Volume fraction, $V_f = 0.45$, is imposed on the system.

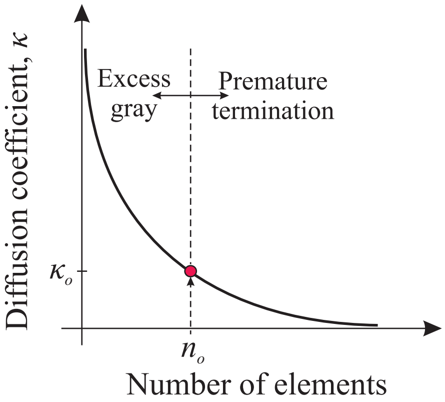


Figure 2.21: Qualitative illustration of the variation of the choice of diffusion coefficient κ versus the mesh refinement. For a particular choice of mesh refinement n_o, if κ_o represents the ideal choice of κ then any $\kappa > \kappa_o$ will result in lesser holes and excess gray region in the converged configuration. On the other hand if $\kappa < \kappa_o$ then the algorithm stops prematurely. Here, the ideal choice of κ is defined as the value which produces topologies with approximately three-to-four elements in the diffuse interface.

The phase-field method discussed here (TNK method) is similar to gradient-based density methods in many aspects. Both methods have the same domain representation and their optimization algorithms are primarily driven by sensitivity information. Material interpolation schemes such as SIMP, RAMP [192, 39] and ECP [245, 244, 246], are equally applicable to both. The main difference between the two methods is in the updating algorithms they employ. Mathematical programming algorithms; such as, Method of Moving Asymptotes (MMA) [205], Optimality Criteria (OC) [39], Sequential Linear Programming (SLP), Sequential Quadratic Programming (SQP), CONvex LINearization approximations (CONLIN) [83]) have been applied to density methods. Additional techniques such as filtering, perimeter constraint [39] and manufacturing constraints [122, 12] are used to regularize the design. In the TNK method, the design is updated using the Allen-Cahn equation (2.41, 2.42). Perimeter constraint is built-in to the updating scheme and can be controlled via the diffusion coefficient, κ. Moreover, the double well potential function (2.42) has a regulating effect as it drives the design towards a 0-1 solution.

Finally, Table 2.5 summarizes the key features of the differential equation-driven methods discussed in this study.

2.5 Concluding remarks

In this work, we study four level-set methods and one phase-field method. The AJT level-set method [8] is the first method we analyze. In this method, shape derivatives are combined with the Hamilton-Jacobi equation for the design front propagation. The second method we look at is the DLK level-set method [216] in which the objective function sensitivity is used in a steepest-descent type update scheme. The WW level-set method [231] transforms the Hamilton-Jacobi equation into a set of ordinary differential equations (ODEs) using multi-quadratic radial basis functions. The ODEs are solved using Euler's method. Challis' level-set method [59] utilizes Burger et al. [52] approach of modifying the Hamilton-Jacobi equation to include topological sensitivity to nucleate holes in the design domain. Finally, we also study the TNK phase-field method [208] for structural topology optimization. The design domain is represented in terms of a phase-field function and evolved using the Allen-Cahn equation; which utilizes objective function sensitivity.

Our study shows that, in all five methods, the final topologies are dependent on the starting initial configurations. Although, the WW method and Challis' method nucleate holes in the design domain, these methods still possess the initial configuration dependency. The AJT method produces designs which are invariant to mesh refinement when optimization starts from a similar topology i.e. starting from a similar initial topology, the method should produce designs of similar topological complexity for different mesh discretizations. In the AJT method, certain parameters such as the number of Hamilton-Jacobi updates per state equation solution, frequency of reinitialization and number of times the reinitialization equation is solved, are chosen heuristically. A physically meaningful scheme needs to be devised which can estimate the optimum parameter values. The DLK method also produces designs which are mesh invariant for similar starting topology but requires very frequent reinitialization for the algorithm to converge, making it computationally expensive and resulting in a slow rate of convergence. Reinitialization is required frequently, because the steepest-descent type update (2.23) used in the DLK method does not preserve the magnitude of the gradient of the level-set function. The slope of the level-set function needs to be controlled near the design interface for convergence. Unlike other level-set methods, the WW method does not periodically reinitialize the level-set function, to save computational cost, and it also aids in the hole nucleation process. But, this also causes the level-set function and its gradients to grow to large values, making accurate approximation of the normal at design boundaries difficult. The WW method also produces dense matrices which need to be inverted frequently, rendering them computationally very expensive. Challis' method possesses a built-in hole nucleation mechanism, because of the modified Hamilton-Jacobi equation which incorporates topological derivatives. But, the final topologies depend on the choice of mesh-dependent parameter w; which determines the influence of topological derivatives. The TNK phase-field method does not require the phase-field function to be a signed distance function, so no reinitialization is needed. In this method, the diffusion coefficient, κ; which controls the thickness of diffuse interface and the perimeter of the final configuration, is mesh dependent and needs to be scaled appropriately for meshes of different discretization. It should be noted that, of all the five methods we studied, only Challis' method enforces black and white solutions.

In this work, we focused on two-dimensional optimization problems for which hole nucleation is a challenge. For the three-dimensional case, Allaire et al. [8] have shown that the traditional level-set approach of using an advection equation with shape derivatives is able to nucleate new holes in the domain due to the pinching of thin walls. Also, we limited ourselves to compliance minimization problems for simplicity. Most methods in the literature tend to perform relatively well in the case of compliance minimization and thus may appear to be similarly effective. Methods which produce similar results for compliance minimization might exhibit drastically different behavior for other objectives, such as compliant mechanism, stress criterion and nonlinear elasticity. Only when these methods are tested against objectives other than compliance can one ascertain the actual robustness of the method.

We would like to point out that the evolution equations for the level-set methods and phase-field method that we have examined in this work can be regarded as some sort of steepest decent method. In the case of compliance, the algorithms strive to minimize the strain energy in the system by placing more material at high strain energy locations. Thus, the differential equation-driven methods follow the same design philosophy as the gradient-based density methods for topology optimization.

Authors	Acronym	Method classification	Evolution equation	Reinitialization	Inherent hole nucleation[9]	Dependence on initial guess	Remarks
Allaire, Jouve and Toader [8]	AJT	Level-set method	Hamilton-Jacobi	Yes	No	Yes	• Parameters, such as, number of update steps of Hamilton-Jacobi equation, frequency of reinitialization, and number of solution steps of reinitialization equation chosen heuristically.
Van Dijk, Langelaar and Van Keulen [216]	DLK	Level-set method	Steepest-descent	Yes	No	Yes	• Requires frequent reinitialization to maintain signed distance nature of the level-set function at all times for convergence which makes the algorithm computationally expensive and slow.
Wang and Wang [231]	WW	Level-set method	Hamilton-Jacobi	No	Yes	Yes	• Elimination of reinitialization and adoption of smoothed naturally extended velocities aids the creation of new holes. • Requires frequent inversion of dense matrices which makes the algorithm computationally very expensive.
Challis [59]	-	Level-set method	Hamilton-Jacobi	Yes	Yes	Yes	• Nucleates holes due to the use of modified Hamilton-Jacobi equation which includes topological derivatives but does not alleviate dependence on initial guess. • Introduces mesh dependent parameter, w, which controls the influence of topological derivatives.
Takezawa, Nishiwaki and Kitamura [208]	TNK	Phase-field method	Allen-Cahn	No	No	Yes	• No reinitialization needed. • Diffusion coefficient, κ, needs to be scaled with mesh refinement for mesh independent designs.

[9] For methods which do not have any inherent hole nucleation mechanism, topological derivatives can be used for hole nucleation.

Table 2.5: Summary of the differential equation-driven methods for two-dimensional topology optimization discussed in this study.

Chapter 3

Phase-field based topology optimization with polygonal elements

In Chapter 2, we have seen that uniform grids have been the common choice of domain discretization in the topology optimization literature. Over-constraining geometrical features of such spatial discretizations can result in mesh-dependent, sub-optimal designs. Thus, in the current work, we employ unstructured polygonal meshes constructed using Voronoi tessellations to conduct structural topology optimization. We utilize the phase-field method, derived from phase transition phenomenon, which makes use of the Allen-Cahn differential equation and sensitivity analysis to update the evolving structural topology. The solution of the Allen-Cahn evolution equation is accomplished by means of a centroidal Voronoi tessellation (CVT) based finite volume approach. The unstructured polygonal meshes not only remove mesh bias but also provide greater flexibility in discretizing complicated (e.g. non-Cartesian) domains. The features of the current approach are demonstrated using various numerical examples for compliance minimization and compliant mechanism problems.

3.1 Introduction

Level-set methods and phase-field methods are two types of differential equation-driven methods that have emerged as an alternative to density-based methods. In general, level-set functions become too flat or too steep during the course of evolution and thus, for numerical accuracy, they need to be reinitialized periodically, for example, to a signed distance function. Phase-field methods are attractive in this regard as they do not require frequent reinitialization. It has been widely used in the field of materials science as a means to study phase transition phenomenon. For instance, it is especially suitable to investigate the stability of systems with multiple unstable phases. Cahn and Hillard [54] and Allen and Cahn [11] used the theory of phase transition to study liquid phases with variable densities. *In essence, the phase-field method is a diffuse interface model where the boundary between phases is not sharp, but considered to have a finite thickness, thus providing a smooth transition for the physical quantities between the phases.* In the phase-field method, explicit interface tracking is avoided and topologies are evolved by solving the governing equations over the complete design domain without prior information about the location of phase interfaces.

For simplicity, topology optimization problems are often solved on Cartesian meshes. The orientation of members in the evolving topologies are thus biased because of the geometrical constraints of such meshes. Accurate representation of general design domains and boundary conditions requires additional effort. Moreover, it is well known that traditional density-based topology optimization on Cartesian meshes suffer from numerical artifacts such as checkerboard patterns and one-node connections [76, 186]. Techniques such as filters [186, 45, 98] may alleviate numerical anomalies and mesh bias. However, Rozvany et al. [171] indicated that such heuristic schemes can result in (considerable) weight increase. Polygonal elements address some of the aforementioned problems. They not only provide convenience and flexibility in discretizing complicated design domains but also lead to optimal designs that are not biased by the mesh discretization. For instance, polygonal elements alleviate one-node connection problems and prevent checkerboard pattern from occurring in density methods [209]. The Voronoi diagram, named after Georges Voronoi [221], has been a popular choice for generating polygonal meshes in the field of computer graphics, robotics, pattern recognition [95, 94] and is the method of choice in this work. Voronoi diagram is a mathematical approach of dividing space into a number of regions using a set of points or seeds. The Voronoi cell associated with each seed is the set of all the points whose distance from this seed is not greater than their distances from other seeds. Some of the prominent Voronoi diagram computing algorithms are naive method, divide-and-conquer method, incremental method, geometric transforms-based method, and digital plane algorithm [151]; lower envelopes-based method [181] and octrees-based method [127]. Detailed surveys of Voronoi diagrams and their applications are available in [20, 21, 151].

In this work, we utilize unstructured polygonal meshes, constructed using Voronoi tessellations, for structural topology optimization employing the phase-field method. In order to evolve the partial differential equation (PDE), known as the Allen-Cahn equation, a centroidal Voronoi tessellation based finite volume approach [219] is used. We also present a heuristic finite difference approach, as an alternate scheme, which can be used for assessing the accuracy of results obtained from the finite volume approach. To generate polygonal meshes, we use `PolyMesher` [210], which is an extension of the work by Bolander and Saito [42] and Yip et al. [243].

The remainder of this chapter is organized as follows. In Section 3.2, we review the polygonal finite element method used in this work. Sections 3.3 and 3.4, address the finite volume and finite difference approaches for solving the Allen-Cahn equation on non-Cartesian domains discretized using unstructured polygonal meshes, respectively. Section 3.5 provides several numerical examples. Finally, we conclude with some remarks in Section 3.6.

3.2 Polygonal finite elements

In the literature, often uniform grids of linear quads/triangles (2D) or bricks/tetrahedra (3D) are used for topology optimization problems. Because of their intrinsic geometrical constraints, such spatial discretizations bias the orientation of members and hence can result in mesh-dependent, sub-optimal designs (cf. Fig. 21 in [209]). In the current work, we use polygonal meshes constructed using Voronoi tessellations [95, 209, 210, 94] to implement the phase-field method. The use of such unstructured meshes not only circumvents mesh bias but also provides greater flexibility in discretizing complicated domains (as demonstrated later) with accurate representation of boundary conditions.

We use Voronoi diagrams to generate polygonal meshes. In this approach, the given design domain with smooth boundaries is first populated with a set of random points/seeds. Using the concept of signed distance function, a set of points are generated which are the reflections of the seeds, lying near the boundary, about the boundary. The Voronoi diagram is generated for the set of random seeds and their reflections. The Voronoi cells corresponding to the random seeds represent the discretized design domain. The Voronoi diagram is forced to be centroidal in order to generate high quality meshes. The Lloyd's algorithm [130] is used for the construction of centroidal Voronoi tessellations (CVTs). For more details on the polygonal mesh generation scheme, the reader is referred to [209, 210]. It should also be noted that due to the random placement of seeds, the node and element numbering will be random, resulting in a stiffness matrix of large bandwidth. If needed, the heuristic reverse Cuthill-McKee (RCM) [73] algorithm is used to reduce the bandwidth of the stiffness matrix. Other equivalent algorithms can also be employed [159, 160].

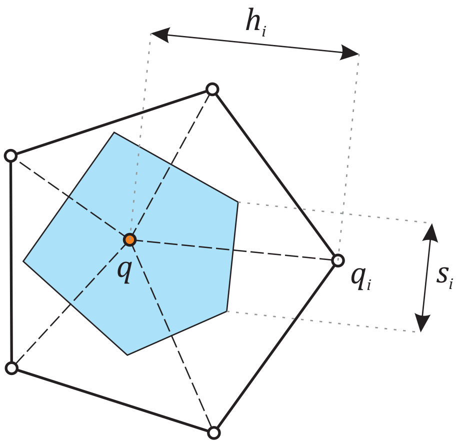
In this work, we use the natural neighbor scheme-based Laplace interpolants to construct finite element shape functions for the polygonal elements [200]. Here we briefly review the finite element scheme for convex polygons. Two points are natural neighbors of each other if they have a common Voronoi edge. Consider a point q and let the set of points $\mathcal{Q} = \{q_1, q_2, ..., q_n\}$ be its natural neighbors. The Laplace shape function for the node q_i is given by:

$$N_i(\boldsymbol{x}) = \frac{\alpha_i(\boldsymbol{x})}{\sum_{\mathcal{Q}} \alpha_j(\boldsymbol{x})}, \qquad \alpha_i(\boldsymbol{x}) = \frac{s_i(\boldsymbol{x})}{h_i(\boldsymbol{x})}, \qquad \boldsymbol{x} \in \mathbf{R}^2 \tag{3.1}$$

where $\boldsymbol{x}$ is the location of $q, \alpha_i(\boldsymbol{x})$ is Laplace weight function, $s_i(\boldsymbol{x})$ is the length of the common Voronoi edge associated with q and q_i, and $h_i(\boldsymbol{x})$ is the distance between q and q_i (Fig. 3.1).

These shape functions satisfy all the desirable properties in the context of a conforming Galerkin approximation such as non-negativity, Kronecker-delta property, and partition of

Figure 3.1: Definition of Laplace shape function. The parameter s_i denotes the length of the common Voronoi edge associated with q and q_i and h_i is the distance between q and q_i.



unity:

$$0 \leq N_i(\boldsymbol{x}) \leq 1, \qquad N_i\left(\boldsymbol{x}_j\right) = \delta_{ij}, \qquad \sum_{\mathcal{Q}} N_i(\boldsymbol{x}) = 1 \tag{3.2}$$

Here, $\boldsymbol{x}_j$ represents the location of node q_j. These functions are also linearly precise or complete:

$$\sum_{\mathcal{Q}} \boldsymbol{x}_i N_i(\boldsymbol{x}) = \boldsymbol{x} \tag{3.3}$$

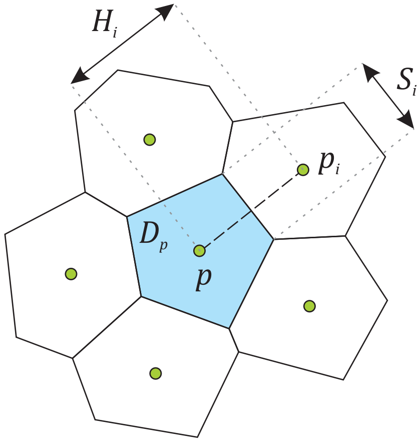
which indicates that a linear function is represented exactly by these shape functions. Furthermore, on the boundary of the domain, the Laplace shape functions are linear which along with Kronecker-delta property ensures that linear essential boundary conditions can be imposed.

3.3 CVT-based finite volume (FV) method for unstructured meshes

In the current work, we employ the approach available in [208] which is discussed in Section 2.3.5. The finite volume (FV) method is a popular method of choice for solving PDEs when dealing with unstructured grids. To solve the Allen-Cahn equation (2.43), we employ a scheme similar to that of Vasconcellos and Maliska [219] who proposed a centroidal Voronoi tessellation (CVT) based finite volume method for fluid flow. Consider a point p and let the set of points $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ be its natural neighbors. The integral form of (2.41), over time t and on each Voronoi cell Ω_p, can be expressed as:

$$\int_{t,\Omega_p} \frac{\partial \phi}{\partial t} dt d\boldsymbol{x} = \int_{t,\Gamma_p} \kappa \nabla \phi \cdot \boldsymbol{n} dt d\boldsymbol{s} - \int_{t,\Omega_p} f'\left(\phi\right) dt d\boldsymbol{x} \tag{3.4}$$

Figure 3.2: Illustration of the CVT-based finite volume scheme. The parameter S_i represents the length of the common Voronoi edge associated with points p and p_i, and H_i denotes the distance between the points p and p_i



Each term in (3.4) can be integrated as shown below (also refer to Fig. 3.2). First,

$$\int_{t,\Omega_p} \frac{\partial \phi}{\partial t} dt d\boldsymbol{x} = \int_{\Omega_p} (\phi^{n+1} - \phi^n) d\boldsymbol{x} \ \approx \ (\phi_p^{n+1} - \phi_p^n) V_p \tag{3.5}$$

where ϕ_p^n is the value of ϕ for the nth iteration at the center of the Voronoi cell corresponding to point p and V_p represents the area of the control volume, in the form of Voronoi cell, centered at p. Next,

$$\int_{t,\Gamma_p} \kappa \nabla \phi \cdot \boldsymbol{n} dt d\boldsymbol{s} \ \approx \int_t \sum_{\mathcal{P}} \left[\kappa \nabla \phi^n \cdot nS\right]_i dt = \left(\sum_{\mathcal{P}} \left[\left(\kappa \frac{\partial \phi^n}{\partial \boldsymbol{n}} \right)_{p,p_i} S_i \right] \right) \Delta t = P_3 \tag{3.6}$$

where S_i is the length of the common Voronoi edge associated with points p and p_i. The directional derivative $(\partial \phi^n / \partial \boldsymbol{n})_{p,p_i}$ can be calculated taking advantage of the local orthogonality property of Voronoi cells:

$$\left(\frac{\partial \phi^n}{\partial \boldsymbol{n}} \right)_{p,p_i} = \frac{\phi_{p_i}^n - \phi_p^n}{H_i} \tag{3.7}$$

where H_i is the distance between points p and p_i. Because an explicit scheme forces the function ϕ to diverge when $\phi \notin [0,1]$, we use a semi-implicit method to simplify the final term in (3.4) as shown below [235]:

$$\int_{t,\Omega_p} f'(\phi)\, dt d\boldsymbol{x} \ \approx V_p \Delta t f'\left(\phi_p^n\right) = V_p \Delta t \begin{cases} \phi_p^{n+1}\left(1 - \phi_p^n\right) r\left(\phi_p^n\right) & \text{for } r\left(\phi_p^n\right) \leq 0 \\ \phi_p^n \left(1 - \phi_p^{n+1}\right) r\left(\phi_p^n\right) & \text{for } r\left(\phi_p^n\right) > 0 \end{cases} \tag{3.8}$$

where

$$r\left(\phi_p^n\right) = \phi_p^n - \frac{1}{2} - 30\eta \frac{\bar{J}'\left(\phi_t\right)}{\left|\bar{J}'\left(\phi_t\right)\right|} \phi_p^n \left(1 - \phi_p^n\right) \tag{3.9}$$

The semi-implicit FV updating scheme for ϕ in (2.43) can thus be expressed as:

$$\phi_p^{n+1} = \begin{cases} \frac{V_p\phi_p^n+P_3}{V_p\left(1-\left(1-\phi_p^n\right)r\left(\phi_p^n\right)\Delta t\right)} & \text{for } r\left(\phi_p^n\right) \leq 0 \\ \frac{V_p\phi_p^n\left(1+r\left(\phi_p^n\right)\Delta t\right)+P_3}{V_p\left(1+\phi_p^n r\left(\phi_p^n\right)\Delta t\right)} & \text{for } r\left(\phi_p^n\right) > 0 \end{cases} \tag{3.10}$$

3.4 A finite difference (FD) method on unstructured meshes

Another method to solve the Allen-Cahn equation is the finite difference (FD) scheme. As before, a semi-implicit scheme is used to discretize the reaction term. The discretized evolution equation on a structured grid follows:

$$\begin{aligned} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} &= \kappa\left(\frac{\phi_{i-1,j}^n - 2\phi_{i,j}^n + \phi_{i+1,j}^n}{(\Delta x)^2} + \frac{\phi_{i,j-1}^n - 2\phi_{i,j}^n + \phi_{i,j+1}^n}{(\Delta y)^2}\right) \\ &\quad + \begin{cases} \phi_{i,j}^{n+1}\left(1-\phi_{i,j}^n\right) r\left(\phi_{i,j}^n\right) & \text{for } r\left(\phi_{i,j}^n\right) \leq 0 \\ \phi_{i,j}^n\left(1-\phi_{i,j}^{n+1}\right) r\left(\phi_{i,j}^n\right) & \text{for } r\left(\phi_{i,j}^n\right) > 0 \end{cases} \end{aligned} \tag{3.11}$$

where

$$r\left(\phi_{i,j}^n\right) = \phi_{i,j}^n - \frac{1}{2} - 30\eta\frac{\bar{J}'\left(\phi_t\right)}{\left|\bar{J}'\left(\phi_t\right)\right|}\phi_{i,j}^n\left(1-\phi_{i,j}^n\right) \tag{3.12}$$

Here Δx and Δy are the distances between grid points in the x and y direction, respectively and $\phi_{i,j}^n$ is the value ϕ for the nth iteration at the grid point $\boldsymbol{x}_{i,j}$. Thus, the scheme to update ϕ using the semi-implicit FD scheme is as follows:

$$\phi_{i,j}^{n+1} = \begin{cases} \frac{\phi_{i,j}^n+\Delta t(P_1+P_2)\kappa}{1-\left(1-\phi_{i,j}^n\right)r\left(\phi_{i,j}^n\right)\Delta t} & \text{for } r\left(\phi_{i,j}^n\right) \leq 0 \\ \frac{\phi_{i,j}^n\left(1+r\left(\phi_{i,j}^n\right)\Delta t\right)+\Delta t(P_1+P_2)\kappa}{1+\phi_{i,j}^n r\left(\phi_{i,j}^n\right)\Delta t} & \text{for } r\left(\phi_{i,j}^n\right) > 0 \end{cases} \tag{3.13}$$

where

$$P_1 = \frac{\phi_{i-1,j}^n - 2\phi_{i,j}^n + \phi_{i+1,j}^n}{(\Delta x)^2}, \qquad P_2 = \frac{\phi_{i,j-1}^n - 2\phi_{i,j}^n + \phi_{i,j+1}^n}{(\Delta y)^2} \tag{3.14}$$

Polygonal meshes are unstructured in nature, i.e., the nodes of the meshes are irregularly arranged. Regular FD scheme based on (3.11) and (3.12) cannot be directly used to solve the Allen-Cahn equation on such a mesh with nodes (or element centers) as grid points. Thus, we propose the following approach to perform the FD, as illustrated by Fig. 3.3.

The ellipse represents the design domain Ω discretized using a polygonal mesh. The ellipse is enclosed within an imaginary rectangular box of length L and width B (represented by dotted lines). The rectangular box is filled with equidistant grid point (represented by small

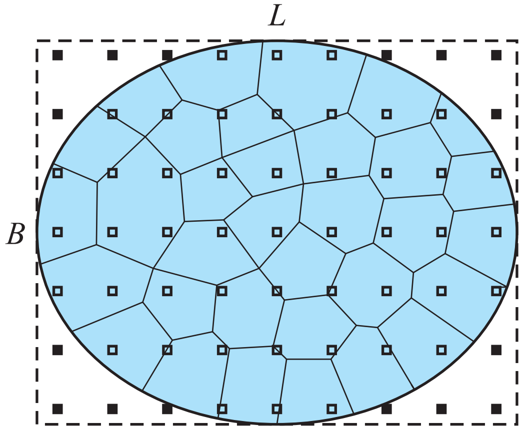


Figure 3.3: Illustration of the finite difference scheme. The design domain Ω, discretized using polygonal elements, is represented by the ellipse and is enclosed within an imaginary rectangular box of length L and width B, represented by dotted lines. The rectangular box is filled with equidistant grid points, shown by small squares. For the grid points lying outside the ellipse (solid squares), the phase-field function value and sensitivity are assigned as zero, whereas, for other grid points, they are assigned the same value as the polygonal element in which they lie.

squares). The regular FD scheme can be applied on this structured grid. We need to resolve the phase-field function value on these grid points. First, a search routine needs to be performed to find the location of each grid point relative to the polygonal elements. For the grid points lying outside the design domain (solid squares), the phase-field function ϕ value is assigned as zero along with zero sensitivity $\bar{J}'(\phi)$. For other grid points, the phase-field function and sensitivity are taken to be the same as the corresponding values of the polygonal element inside which they lie. We have assumed that the phase-field function and the sensitivity are constant inside each element. After conducting an appropriate number of updates of the Allen-Cahn equation, the quantities computed on the structured grid need to be mapped backed to the polygonal mesh. We take the value of phase-field function for each element as the average of the values at all the grid points lying inside that element.

In this approach, we need at least one grid point to lie inside each polygonal element, which can be done by having a structured grid of sufficient refinement. Otherwise, the element's phase will not change throughout the optimization or, in other words, it will act as a "dead" element, resulting in an incorrect topology. To estimate the structured grid size, the number of grid points in the x and y direction are given by $\beta\sqrt{n_{elem}L/B}$ and $\beta\sqrt{n_{elem}B/L}$, respectively. Here n_{elem} is the number of polygonal elements in the finite element mesh and β is a multiplicative factor lying in the range $1.5-2$. The proposed finite difference scheme is heuristic in nature and possesses some approximations. The motivation behind its development is to provide an alternate scheme which gives a general idea of how the converged topologies should look like and thus can be used for estimating the accuracy of the results obtained from the finite volume scheme. In general, the CVT-based finite volume scheme, being more accurate, should be used.

3.5 Numerical examples

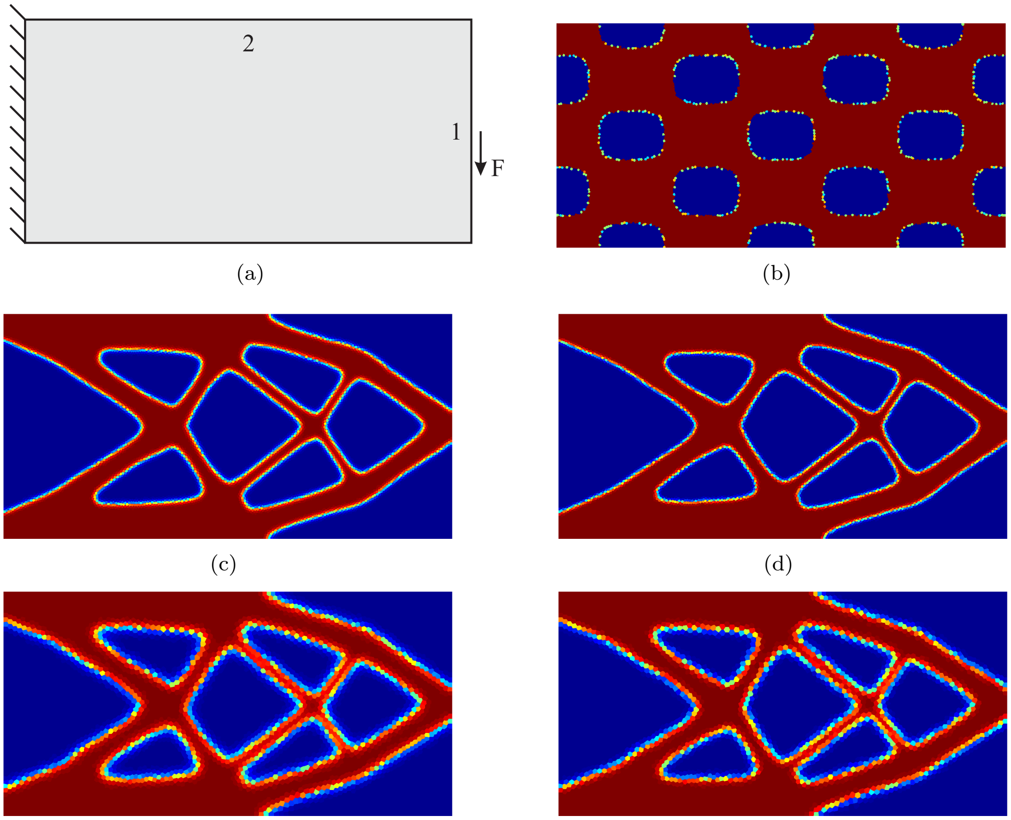
The use of polygonal finite elements makes it possible to perform topology optimization for complicated geometries. The mesh generator by Talischi et al. [210] based on the implicit description of the design domain and centroidal Voronoi diagrams along with the FV/FD scheme makes this possible. In this section, we first illustrate some examples with conventional rectangular design domains for benchmark compliance minimization and linear compliant mechanism problems, followed by examples with non-conventional design domains. For all the examples, the parameters adopted are: $\eta = 10$, $k_{min} = 10^{-4}$, Young's modulus E = 1, and Poisson's ratio $\nu = 0.3$. The time step Δt satisfies the CFL condition [71, 208] and consistent units are employed. Note that the solution of the elasticity problem (1.1, 1.2) is computationally expensive compared to one update step of the Allen-Cahn equation. Therefore, for fast convergence, after each FE iteration, we perform 20 FV/FD update steps of the Allen-Cahn equation [208, 8].

3.5.1 Cantilever beam problem

We first consider cantilever beam with load applied at the middle of right face. The objective is to obtain the stiffest configuration while using the least amount of material. The domain size is 2×1, discretized with 20,000 polygonal elements. The Lagrange multiplier λ is fixed at 95.

The diffusion coefficient is set to $\kappa = 1 \times 10^{-5}$ for all the examples in which the FD scheme is used to solve the Allen-Cahn equation. When the FV scheme is used to update the Allen-Cahn equation, the diffusion coefficient is taken as $\kappa = 2 \times 10^{-5}$, unless otherwise specified. The reason we chose a slightly higher diffusion coefficient for the FV approach is that the diffusion coefficient affects the thickness of diffuse interface and convergence is hindered if the thickness is too small. A suitable value needs to be chosen based on level of mesh refinement. Finer meshes require a higher diffusion coefficient. In case of the FD scheme, the superimposed structured grid, on which FD operations are performed, is finer than the polygonal mesh thus a smaller value of diffusion coefficient is needed. The FV scheme is performed on the polygonal mesh itself, so a slightly larger diffusion coefficient is chosen. Figure 3.4(b) is chosen as the initial topology for the cantilever beam problem. Since the phase-field method can't generate holes, the initial topology needs to have enough holes so that it can converge to a meaningful topology. The resulting topologies, Fig. 3.4(c) (FV scheme is used to update Allen-Cahn equation) and Fig. 3.4(d) (FD scheme is used to update Allen-Cahn equation), are consistent with the ones seen in the literature [8, 208]. For comparison purposes, we also solve the cantilever beam problem on a coarser mesh with 5,000 elements. All the parameters are chosen the same as before except $\kappa = 4 \times 10^{-5}$ for

FD and $\kappa = 8 \times 10^{-5}$ for FV. Converged topologies (Fig. 3.4(e), Fig. 3.4(f)) are similar to the ones on the finer mesh.



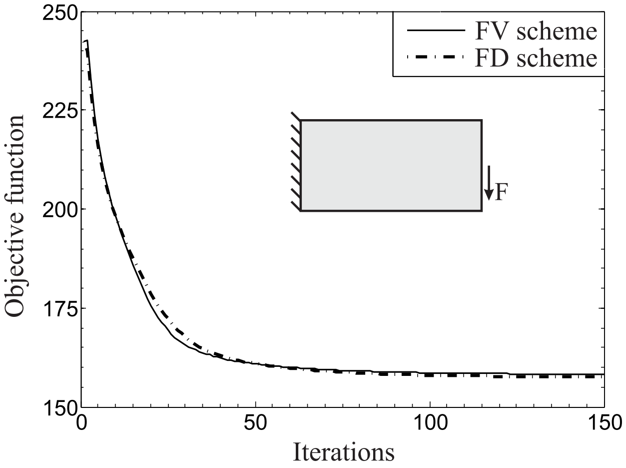
(a) (b) (c) (d) (e) (f)

Figure 3.4: Cantilever beam problem with load applied at the middle of right face. (a) Problem description. (b) Initial topology on 20,000 polygonal element mesh. Converged topologies on mesh discretization of 20,000 polygonal elements using (c) FV scheme, (d) FD scheme, and also on mesh discretization of 5,000 polygonal elements using (e) FV scheme, (f) FD scheme.

The convergence history of the objective function $\bar{J}$ for the above problem, for mesh discretization of 20,000 polygonal elements (Figs. 3.4(c) and (d)), is shown in Fig. 3.5. The FV scheme curve has a steeper slope, indicating a faster rate of convergence. This makes sense because the FD scheme is an approximate scheme which is less accurate than the FV scheme. But, both methods ultimately converge to similar objective function values.

The phase-field method converges to a local minimum, which, like other implicit function methods for topology optimization, such as level set method, is strongly dependent on the initial topology. This is due to the fact that this method can not generate holes in the domain - holes can only collapse. Converged topologies using the FV scheme are shown in Fig. 3.6(b) and Fig. 3.7(b) for the two other initial guesses of Fig. 3.6(a) and Fig. 3.7(a), respectively. All the parameters are kept the same as before.

Figure 3.5: Convergence history of the objective function for the cantilever beam problem of Fig. 3.4 for mesh discretization of 20,000 polygonal elements.



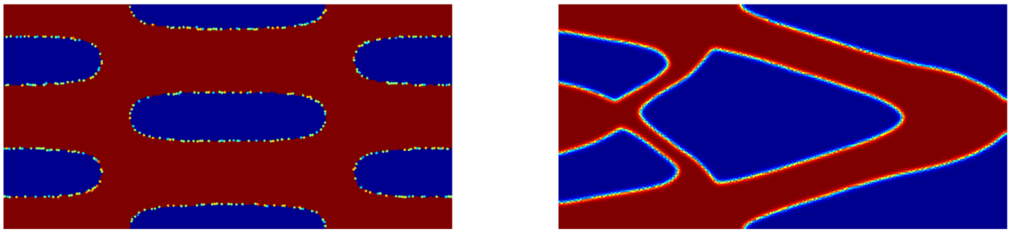

(a)
(b)

Figure 3.6: Cantilever beam problem with a different initial topology. (a) Initial topology with 7 holes. (b) FV scheme.

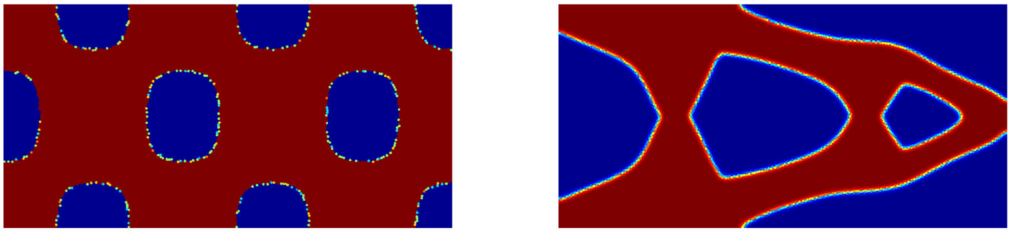
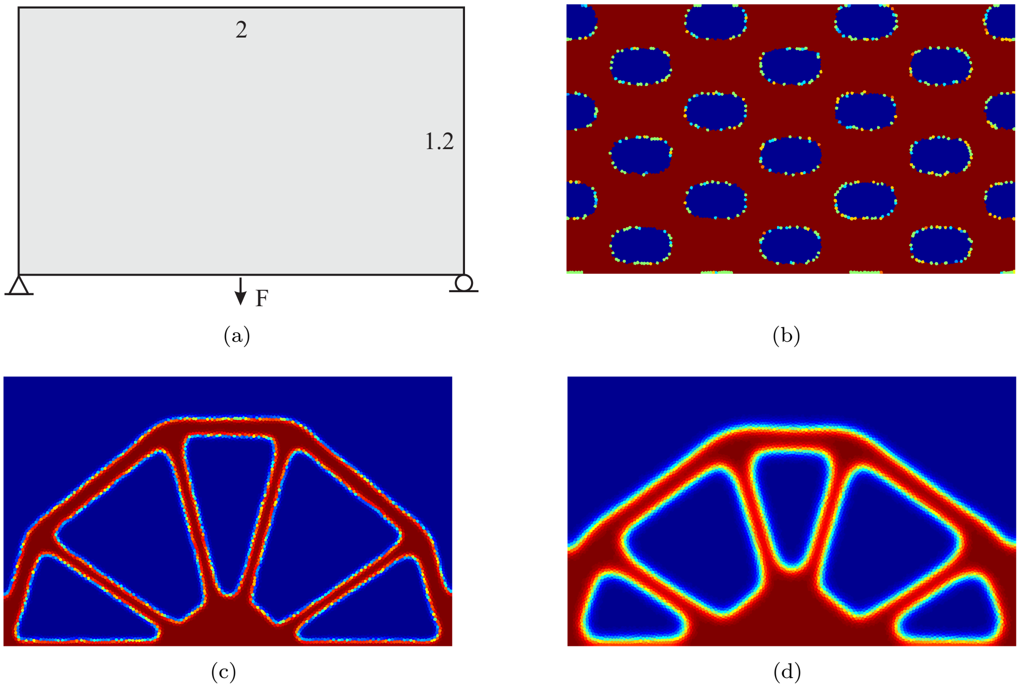

(a)
(b)

Figure 3.7: Cantilever beam problem with another initial topology. (a) Initial topology with 9 holes. (b) FV scheme.

3.5.2 Bridge problem

We next look at the bridge problem and study the influence of the diffusion coefficient, κ, on the optimization. The size of the design domain is a rectangle of size 2×1.2, discretized with 15,360 polygonal elements. The bottom corners are restrained by pin and roller supports, and a unit vertical force is applied at the middle of the bottom face (Fig. 3.8(a)). The objective (same as the last example) is to obtain the stiffest configuration while using the least amount of material. The parameter λ is chosen as 60 and $\kappa = 2 \times 10^{-5},\ 10 \times 10^{-5}$. Approximately 200 finite element iterations are needed for convergence of both the results, with the initial guess shown in Fig. 3.8(b). The CVT based finite volume scheme is used to solve the evolution equation.



(a)
(b)
(c)
(d)

Figure 3.8: Bridge problem solved using different diffusion coefficients κ for the FV approach. (a) Problem description. (b) Initial topology. (c) $\kappa = 2 \times 10^{-5}$. (d) $\kappa = 10 \times 10^{-5}$.

It is evident from Figs. 3.8(c) and (d) that κ influences the thickness of the diffuse interface. For $\kappa = 2 \times 10^{-5}$ and $\kappa = 10 \times 10^{-5}$, 28.2% and 46.3% elements, respectively, have phase-field values between 0.01 and 0.99 (Note that the design boundary $\partial\omega$ lies in the region $0 < \phi(\boldsymbol{x}) < 1$). Thus, larger κ leads to a thicker interface. The current phase-field method implicitly possesses perimeter control effect which can be varied through κ. Bigger voids (larger perimeter) are obtained for lower κ (Fig. 3.8(c)) and smaller voids for larger κ (Fig. 3.8(d)). From this one may conclude that the smaller the value of κ, the better resolved the interface is. However, our numerical experiments using the current phase-field method have shown that, in order for the topologies to evolve smoothly, an appropriate value of κ has to be chosen for a particular mesh discretization. Hence, κ can not be indefinitely reduced to get a sharp interface. If a sharp, perfectly resolved interface is desired then an adaptive mesh refinement strategy near the interfaces may be adopted [81, 224] along with adaptive reduction in κ. Although pertinent, this investigation is beyond the scope of the current work.

3.5.3 Inverter problem on rectangular design domain

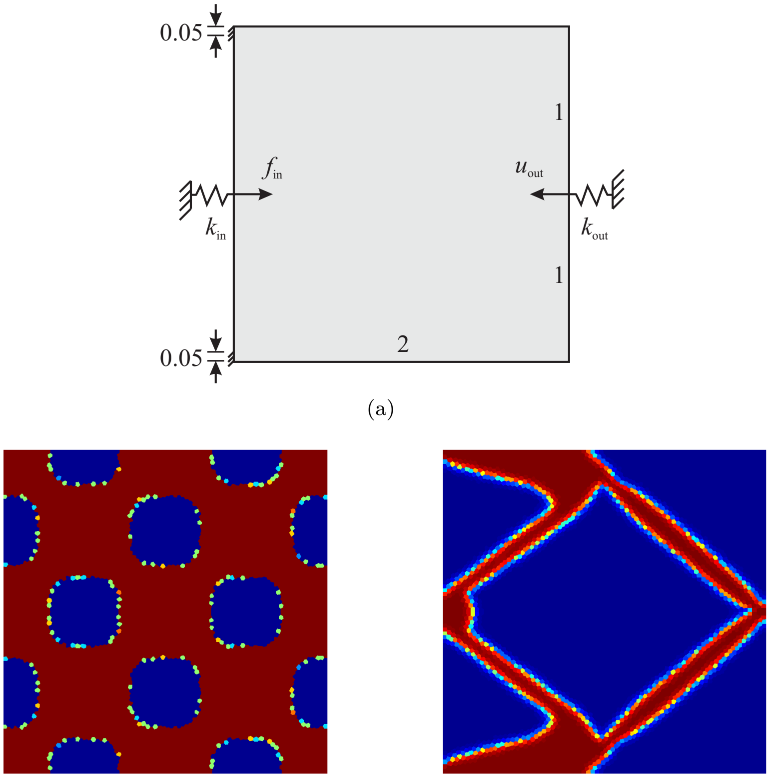
The phase-field method (discussed here) can also be used to solve compliant mechanism problems. We look at the classical inverter problem discussed in, for example, Bendsøe and Sigmund [39]. The problem is shown in Fig. 3.9(a). The domain is a square of size 2×2, discretized using 6,000 polygonal elements. It is fixed on the top and bottom corners on the left face. The objective of the optimization problem is to maximize the output displacement u_{out}. Spring stiffnesses k_{in} and k_{out} are taken to have the same values of the components of the global stiffness matrix at the corresponding degrees of freedom. In order to obtain the sensitivities needed for the double well potential function, an adjoint system needs to be solved (see, for example, Bendsøe and Sigmund [39]). The FV scheme is used to solve the Allen-Cahn equation, and we set $\lambda = 0.02$ and $\kappa = 10 \times 10^{-5}$. For the initial guess of Fig. 3.9(b), the converged configuration is shown in Fig. 3.9(c), which is similar to Fig. 5.5 of [39].

3.5.4 Cantilever beam problem on a circular segment domain

The benchmark example of cantilever beam problem, with vertically downward load applied on the midpoint of the right face, is now solved on a design domain in the shape of a circular segment (Fig. 3.10(a)). Figure 3.10(b) is chosen as the initial guess. The design domain is a symmetric polygonal mesh (about the horizontal axis) with 12,800 elements. Figure 3.10(c) shows the converged topology, utilizing the FV scheme and $\lambda = 95$. Although the design domain chosen here is different from the conventional one, the converged topologies are similar to the ones with rectangular design domains (cf. Fig. 3.10 and Fig. 3.4).

3.5.5 Bridge problem on a semi-circular domain

Next, we consider the bridge problem on a semi-circular design domain (Fig. 3.11(a)). The boundary conditions are the same as the ones for the bridge problem on a rectangular domain discussed before. The polygonal mesh used to discretize the design domain consists of 11,000 elements, and λ is chosen as 60. The optimization is performed with Fig. 3.11(b) as the initial guess and it converges to Fig. 3.11(c) for the FV updating scheme, which resembles

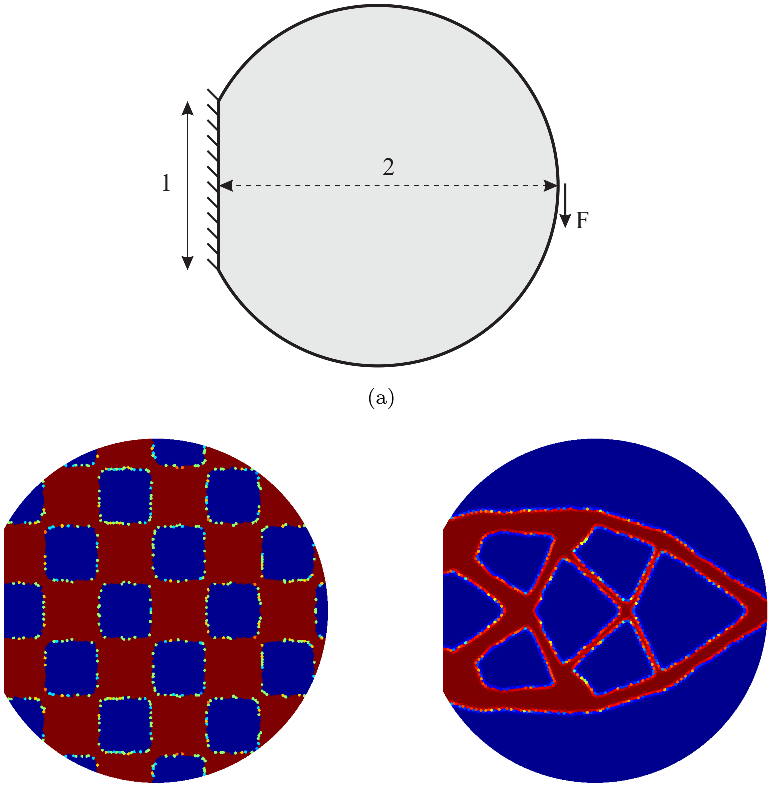


(a)

(b)

(c)

Figure 3.9: Inverter problem on a polygonal mesh with 6,000 elements. (a) Problem description. (b) Initial topology. (c) Final configuration utilizing FV scheme.



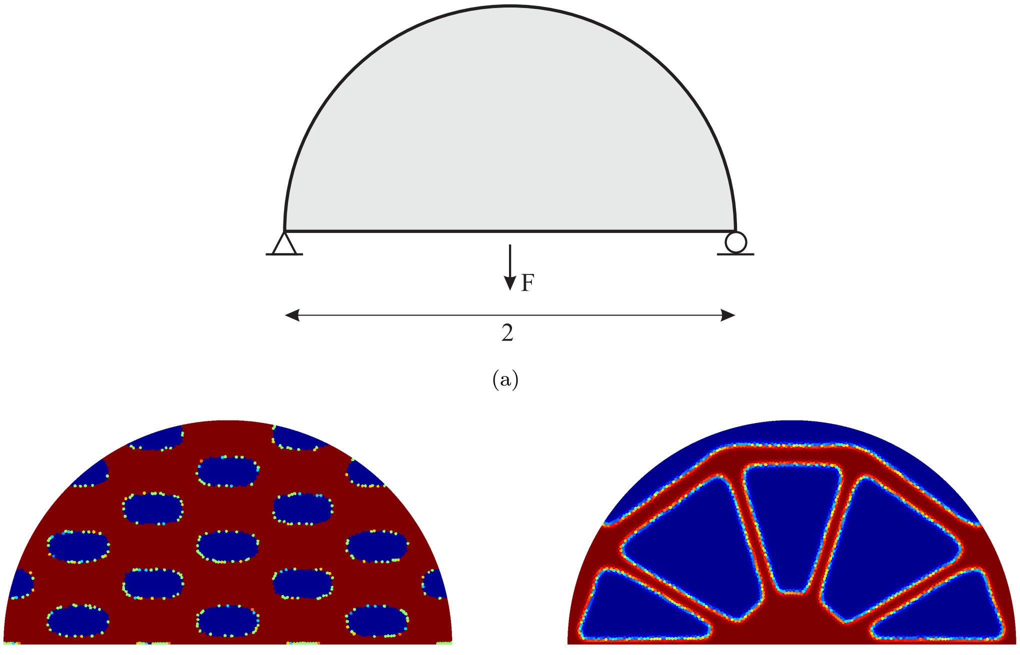
(a)

(b)

(c)

Figure 3.10: Cantilever beam problem on a circular segment design domain. (a) Problem description. (b) Initial topology. (c) Converged topology using FV scheme.

the result obtained on the rectangular domain (Fig. 3.8(c)).



(a)

(b)

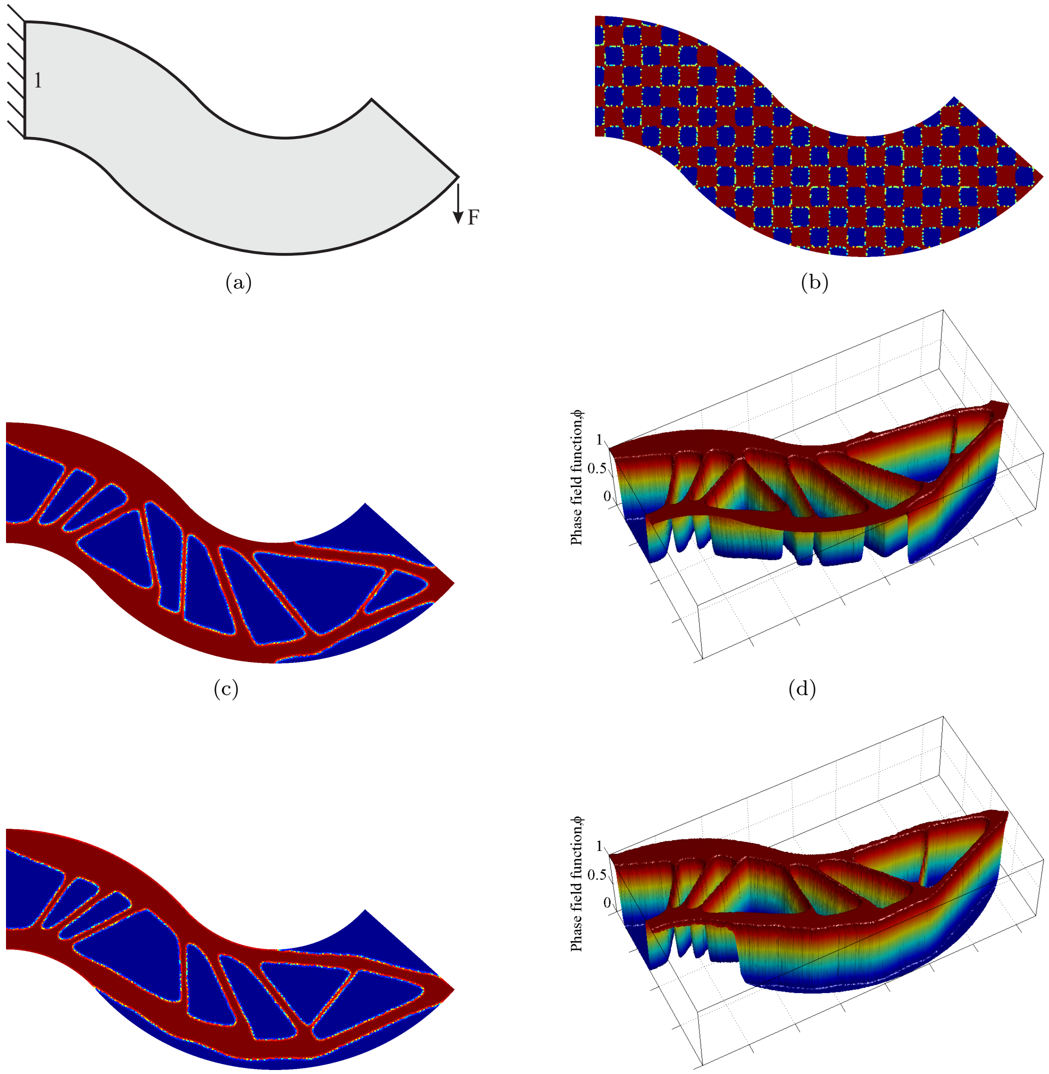
(c)

Figure 3.11: Bridge problem on semi-circular design domain. (a) Problem description. (b) Initial topology. (c) FV scheme.

3.5.6 Curved cantilever beam problem

In the literature, rectangular/cuboidal design domains have been the preferred domain shapes for topology optimization. To depart from this trend, consider the problem of cantilever beam on a doubly curved design domain (Fig. 3.12(a)). The domain is discretized using 20,000 polygonal elements. The Lagrange multiplier λ is chosen as 250. Converged topologies for the chosen initial design, Fig. 3.12(b), are shown in Fig. 3.12(c) for the FV scheme and Fig. 3.12(e) for the FD scheme. Figures 3.12(d) and 3.12(f) are the 3D visualizations of the corresponding phase-field functions.

Figure 3.13 shows the convergence history of the objective function, $\bar{J}$, for the above curved cantilever beam problem (Fig. 3.12). Although both FV and FD schemes seem to converge to similar objective function values, the converged topologies are somewhat different. As stated before, the FD scheme involves more approximation than the FV scheme (refer to Section 3.4). In the FD scheme, the phase-field function values at the grid points (lying inside the design domain) are assumed to be the same as the corresponding value of the polygonal element inside which they lie. This leads to inefficient evaluation of $\nabla^2\phi$ in (2.43) which results in a different converged topology when compared to the FV scheme. This inefficiency reduces as the mesh becomes finer. The differences between the schemes



(a) (b)

(c) (d)

(e) (f)

Figure 3.12: Curved cantilever beam problem. (a) Problem description. (b) Initial topology. (c) FV scheme. (d) 3D visualization of (c). (e) FD scheme. (f) 3D visualization of (e).

are amplified because of the unstructured nature of the polygonal meshes used here. When structured quad elements are used and only one grid point lies inside each quad for the FD scheme, then both methods produce exactly the same result.

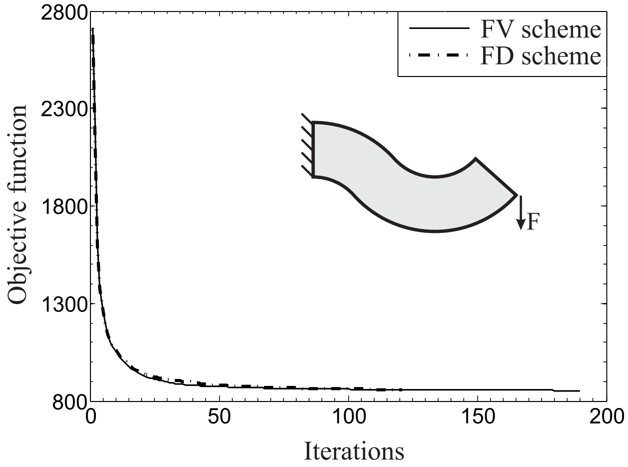


Figure 3.13: Convergence history of the objective function for the curved cantilever beam problem of Fig. 3.12.

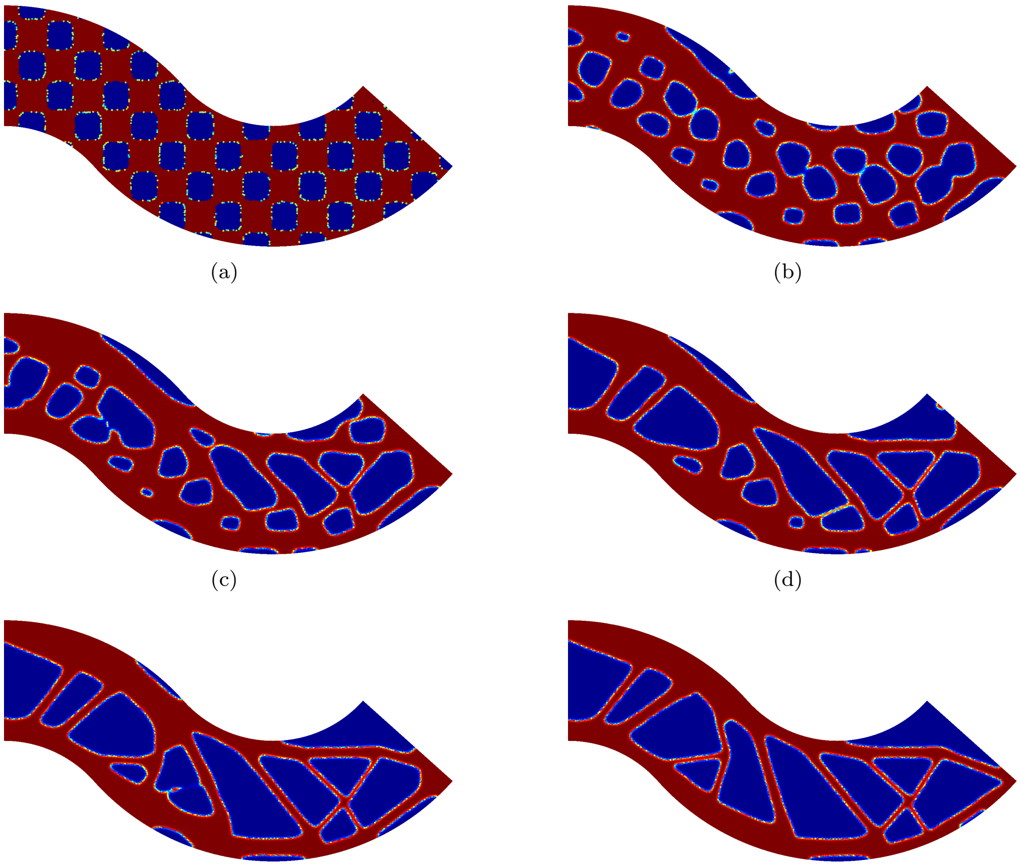
To demonstrate the previously stated fact that, in the phase-field method, converged topologies depend on the initial guess, we solved the doubly curved cantilever problem on a different initial guess (Fig. 3.14(a)). Figures 3.14(b)-(f) show the evolution of topologies over time for the new initial guess.

3.5.7 Inverter problem on circular segment domain

Finally, we consider the inverter problem on a non-rectangular design domain. The domain is in the shape of a circular segment, discretized into 6,000 polygonal elements. All the other parameters are kept same as the inverter problem on the rectangular domain discussed before (cf. Fig. 3.9). The problem description is shown in Fig. 3.15(a). The design domain of circular segment chosen here, although non traditional, has similar boundary and loading conditions to the traditional example on a rectangular domain (see Bendsøe and Sigmund [39]). The converged topology (Fig. 3.15(c) for FV updating schemes) is similar to the one with rectangular design domain (cf. Figs. 3.15(c) and 3.9(c)).

3.6 Concluding remarks

In the current work, we employ a fully unstructured polygonal finite element based mesh to implement a phase-field method for structural topology optimization. The polygonal meshes are based on Voronoi tessellations [210] which not only facilitate non-mesh biased designs but also provide greater flexibility in discretizing non-Cartesian design domains. A CVT-based finite volume method is used to solve the phase-field evolution equation (Allen-Cahn PDE)



(a)

(b)

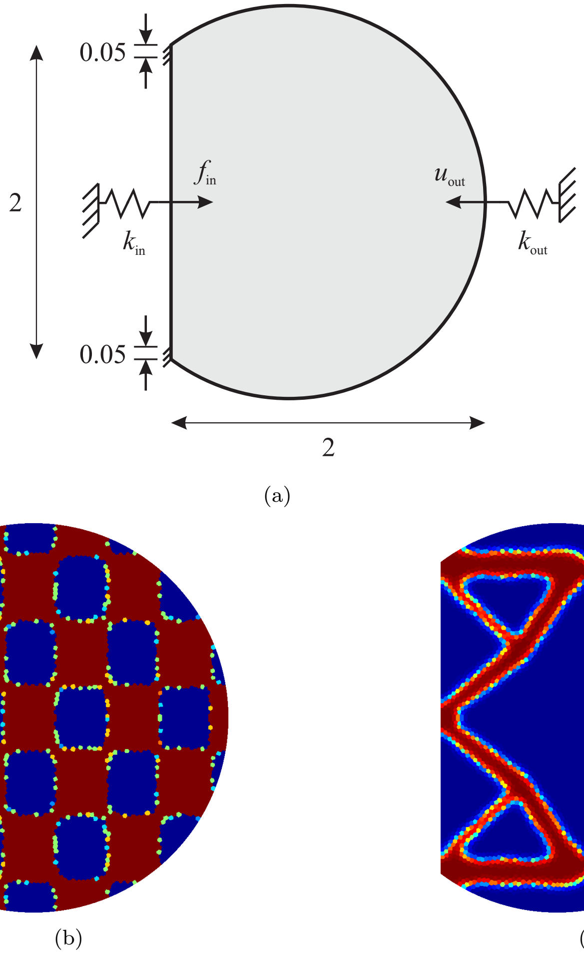
(c)

(d)

(e)

(f)

Figure 3.14: Evolution of the topology for the curved cantilever beam problem using FV scheme for a different initial guess. (a) Initial topology. (b) Iteration 13. (c) Iteration 19. (d) Iteration 29. (e) Iteration 51. (f) Converged topology.



(a)

(b)

(c)

Figure 3.15: Inverter problem on a circular segment design domain. (a) Problem description. (b) Initial topology. (c) FV updating scheme.

on unstructured polygonal meshes. An alternate approach using a finite difference scheme is also presented to solve the phase-field equation.

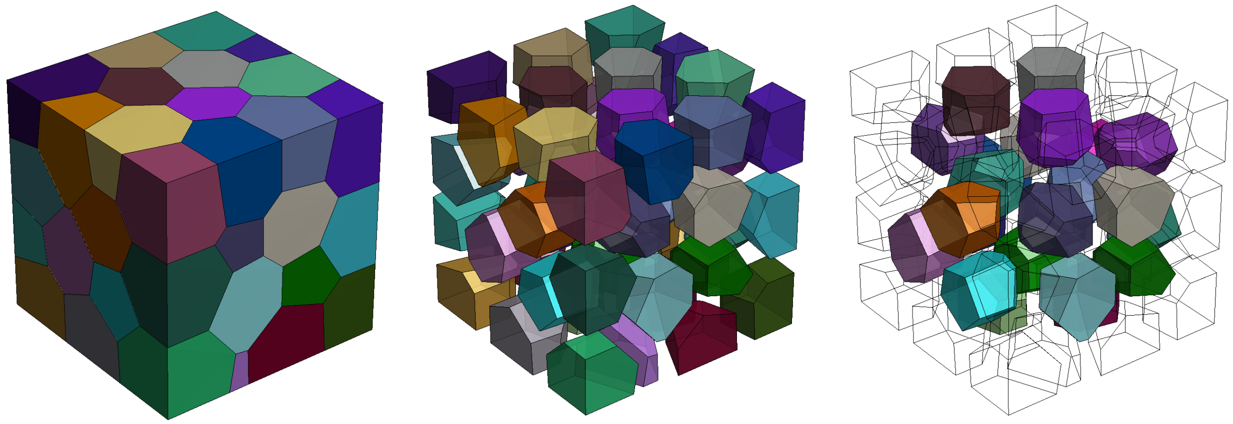
Phase-field methods, similar to the one discussed in this work, bear a resemblance to the level-set methods in the sense that the topologies are represented in terms of implicit functions and evolved using certain governing PDEs. The most characteristic difference between the two is the fact that in level-set methods the interface is explicitly defined and tracked ($\phi = 0$ contour), whereas, in the phase-field approach the interfaces have a finite thickness (requiring no tracking of the interfaces). In the phase-field method, topologies are evolved by solving the governing equations over the complete design domain without any prior knowledge of location of phase boundaries. In order to resolve the phase interfaces (obtaining a 0-1 design) an adaptive mesh refinement strategy or other alternative approaches may be used, e.g. Sun and Beckermann [201].

The present approach has been used to solve 2D compliance minimization and compliant mechanism problems on complicated design domains. It can also be used to solve structural optimization problems such as eigenvalue problems, design dependent load problems, and nonlinear elasticity problems on any desired design domain. For this purpose, sensitivities need to be evaluated, for each particular objective function, to define the double well potential function. Since the phase-field method employed in this work has no embedded hole generation mechanism, the final topologies are greatly influenced by the initial shapes. Topological derivatives [80, 189, 57] can be used to alleviate this issue.

Chapter 4

A Mimetic Finite Difference-inspired approach for three-dimensional linear elasticity problem using polyhedrons

Polyhedral meshes are one of the popular choices in the industry for obtaining accurate and reliable finite element solutions on complicated domains. Polyhedral meshes provide flexibility in discretizing complicated domains [79] and capturing the domain and boundary conditions accurately. Moreover, techniques such as mesh refinement and coarsening produce elements which are inherently polyhedral. In fluid dynamics simulations, the fluid flow should ideally be perpendicular to the element surfaces [79]. Polyhedral elements with many faces have a greater probability of satisfying such a criteria. Figure 4.1 illustrates a domain discretized using polyhedral elements. In three-dimensions, available finite element interpolants on polytopes such as natural neighbor-based coordinates and maximum entropy shape functions are undesirable, because numerical computations need to be performed in physical coordinates, as iso-parametric formulations can not be applied. Also, high quadrature order needs to be adopted for accurate results.



(a) (b) (c)

Figure 4.1: Mesh visualization. (a) Original mesh. (b) Split view. (c) Split view with wire frame.

The Virtual Element Method (VEM), which has evolved from the Mimetic Finite Difference methods, addresses both the issues of accuracy and efficiency. The goal of the current work is to explore VEM for three-dimensional linear elastic problems. The Virtual Element Method is a Galerkin approach, but unlike the classical finite element methods does not require explicit computation of the approximation space. Its underlying philosophy is the explicit construction of the discrete bilinear and load linear forms without actually evaluating the canonical basis functions and their derivatives inside the elements. The construction begins with the kinematic decomposition of the element deformation states into rigid body, constant strain and higher order modes. Two projection matrices are defined and constructed to extract rigid body and constant strain components of deformation and subsequently used to explicitly construct the stiffness matrix and the load vector. As it will be shown later, the construction reduces to the computation of purely geometric quantities. We use linear polyhedral elements in which degrees of freedom are associated with the vertices. Twelve basis functions are chosen which span the linear polynomial space. Surface integrals encountered during the construction of the stiffness matrix are evaluated numerically using a nodal quadrature and a conforming interpolant quadrature schemes. The chapter focuses on elaborating upon the core mathematical concepts underlying the VEM within the context of elasticity problems and details the implementation of the method for general polyhedral meshes. We hope that this work will be beneficial to the engineering and industrial community.

The construction of the bilinear form by splitting the discrete space into polynomial and non-polynomial spaces is based on similar techniques developed in the early 1980's to tackle the issue of hourglass instabilities arising out of under integration of the stiffness matrix. In large computational problems, particularly dynamic problems, which require repeated computation of the stiffness matrix, the use of full integration schemes can be computationally expensive. Reduced integration schemes are often used to alleviate this issue but may cause mesh instabilities known as hour glassing. Consider a linear brick element which has twenty four degrees of freedom; out of which, six are rigid body, six are uniform strain and twelve are non-constant strain modes. Due to the reduced integration scheme (one-point quadrature in this case), hourglass modes or zero energy modes lead to singular stiffness matrix for certain boundary conditions. One of the early attempts to deal with this phenomenon was by Maenchen and Sack [136] who introduced an artificial viscosity to tackle hourglass modes in two-dimensional meshes for finite difference methods. The finite element version of Maenchen and Sack's stabilization was developed by Belytschko and Kennedy [33]. It was realized later that merely introducing viscous damping is not the most effective technique to handle hourglassing. Stabilization approaches were then developed to deal with the hourglass modes which enforced the requirement of orthogonality of hourglass modes to the rigid-bodyand uniform-strain modes [117, 123]. This requirement is essential, because the hourglass control should not prevent the element from passing the patch test (also known as the consistency condition) and should eliminate rank deficiency of the discrete bilinear form. In their work, Kosloff and Frazier [123] added an hourglass stiffness matrix, constructed from the pure bending modes, to the one-point quadrature stiffness matrix to suppress hourglass modes. For non-rectilinear elements, their method involves the solution of four sets of eight equations for hexahedrons. Subsequently, Flanagan and Belytschko [82] and Belytschko et al. [35] developed explicit expressions for hourglass control in two and three-dimensional problems which does not involve solving any equations. The connections between hourglass stabilization and the VEM are elaborated in the subsequent sections.

The remainder of this chapter is organized as follows. In Section 4.1, we discuss the three-dimensional elasticity problem formulation and the requirements for a well defined problem. Section 4.2 provides the implementation details for the elasticity problem. Expressions for the stiffness matrix and the force vectors are derived. In Section 4.3, the VEM implementation is evaluated using the displacement patch test and shear loaded cantilever beam bending problems. Finally, we conclude with some remarks in Section 4.4.

4.1 Elasticity problem

In this work, we consider a linear elastic body, $\Omega \subseteq \mathbb{R}^3$, characterized by fourth order tensor $\mathbf{C}$, under small deformations subjected to displacement boundary conditions $\boldsymbol{g}$, surface tractions $\boldsymbol{t}$ and body forces $\boldsymbol{b}$.

4.1.1 Continuous and discrete formulation

We define the continuous problem as:

$$\begin{gathered} \text{Find} \quad \boldsymbol{u} \in \boldsymbol{\mathcal{V}}^{\boldsymbol{g}} \\ a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}), \qquad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}^{\mathbf{0}} \end{gathered} \tag{4.1}$$

where

$$\begin{gathered} a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\boldsymbol{x}, \qquad \ell(\boldsymbol{v}) = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{v}\, d\boldsymbol{x} + \int_{\Gamma_T} \boldsymbol{t} \cdot \boldsymbol{v}\, d\boldsymbol{s} \\ \boldsymbol{\mathcal{V}}^{\boldsymbol{g}} = \left\{ \boldsymbol{v} \in H^1(\Omega)^3 : \boldsymbol{v}|_{\Gamma_D} = \boldsymbol{g} \right\} \end{gathered} \tag{4.2}$$

For simplicity, let $\boldsymbol{g} = \mathbf{0}$ and subsequently, $\boldsymbol{\mathcal{V}}^{\boldsymbol{g}} = \boldsymbol{\mathcal{V}}^{\mathbf{0}} = \boldsymbol{\mathcal{V}}$. Here, $\boldsymbol{\varepsilon}$ is the second order linearized strain tensor, $\boldsymbol{\varepsilon}(\boldsymbol{u}) = {}^1/_2(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ and Γ_D represents the boundary of Ω on which Dirichlet boundary conditions are applied. Assuming, the bilinear form is continuous and elliptic, i.e., $\exists\, C < \infty$ and $\alpha > 0$ such that:

$$|a(\boldsymbol{u}, \boldsymbol{v})| \leq C \|\boldsymbol{u}\|_{\boldsymbol{\mathcal{V}}} \|\boldsymbol{v}\|_{\boldsymbol{\mathcal{V}}}, \qquad a(\boldsymbol{v}, \boldsymbol{v}) \geq \alpha \|\boldsymbol{v}\|^2_{\boldsymbol{\mathcal{V}}} \qquad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}} \tag{4.3}$$

problem (4.2) has a unique solution. Let, $\mathcal{H}$ to be a null sequence of positive real numbers and $(\boldsymbol{\mathcal{V}}_h)_{h\in\mathcal{H}}$ to be a family of conforming finite-dimensional subsets of space $\boldsymbol{\mathcal{V}}$, i.e. $\boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$, then the discrete Galerkin version of (4.1) is written as:

$$\begin{gathered} \text{Find} \quad \boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h \\ a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = \ell_h\left(\boldsymbol{v}_h\right), \qquad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h \end{gathered} \tag{4.4}$$

Under the assumption of continuity and uniform $\boldsymbol{\mathcal{V}}_h$-ellipticity of $a_h(\cdot,\cdot)$ and continuity of ℓ_h, the discrete form (4.4) admits a unique solution $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h$.

4.1.2 Convergence properties

Strang's first lemma states that for a uniform $\boldsymbol{\mathcal{V}}_h$-elliptic family of bilinear forms $(a_h(\cdot,\cdot))_{h\in\mathcal{H}}$, there exists a constant c independent of the space $\boldsymbol{\mathcal{V}}_h$ such that:

$$\begin{aligned} \|\boldsymbol{u} - \boldsymbol{u}_h\|_{\boldsymbol{\mathcal{V}}} \leq c \Bigg(\inf_{\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h} \Bigg\{ \|\boldsymbol{u} - \boldsymbol{v}_h\|_{\boldsymbol{\mathcal{V}}} + \sup_{\boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h} \frac{|a(\boldsymbol{v}_h, \boldsymbol{w}_h) - a_h(\boldsymbol{v}_h, \boldsymbol{w}_h)|}{\|\boldsymbol{w}_h\|_{\boldsymbol{\mathcal{V}}}} \Bigg\} + \\ \sup_{\boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h} \frac{|\ell(\boldsymbol{w}_h) - \ell_h(\boldsymbol{w}_h)|}{\|\boldsymbol{w}_h\|_{\boldsymbol{\mathcal{V}}}} \Bigg) \end{aligned} \tag{4.5}$$

Next, we show that the VEM discrete solution naturally satisfies a similar constraint due to certain assumptions and characteristics built into the method. To this end, let us assume the discrete bilinear form can be written as the sum of the contributions from the elements as:

$$a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = \sum_E a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) \qquad \forall \boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h \tag{4.6}$$

where the discrete bilinear form $a_h^E(\cdot,\cdot)$ is defined on $\boldsymbol{\mathcal{V}}_h^E \times \boldsymbol{\mathcal{V}}_h^E$ and $\boldsymbol{\mathcal{V}}_h^E$ is the restriction of $\boldsymbol{\mathcal{V}}_h$ to a polyhedral element E. We assume $\boldsymbol{\mathcal{V}}_h$ to be a conforming space of smooth displacement fields which implies the validity of (4.6). From the viewpoint of a Galerkin approximation, $\boldsymbol{\mathcal{V}}_h^E$ represents the space spanned by smooth shape functions which are non-negative, form a partition of unity, satisfy the Kronecker-delta property, are linearly precise and C^0 on the edges. In the VEM, we aim to construct the discrete bilinear form in such a way that the following properties are satisfied.

1. **Consistency**:

In finite element analysis, a patch test is an indicator of the consistency of a method. In the displacement patch test, displacement boundary conditions are applied to the patch and we verify that the patch is able to capture the rigid body and constant strain states. In the VEM, we construct the discrete bilinear form to satisfy:

$$\begin{gathered} a_h^E(\boldsymbol{p}_1, \boldsymbol{v}_h) = a^E(\boldsymbol{p}_1, \boldsymbol{v}_h) \\ \forall E, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E, \quad \forall \boldsymbol{p}_1 \in \mathbb{P}_1(E)^3 \end{gathered} \tag{4.7}$$

where a^E is the restriction of continuous bilinear form a, and $a(\boldsymbol{u}, \boldsymbol{v}) = \sum_E a^E(\boldsymbol{u}, \boldsymbol{v})$, $\forall \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{\mathcal{V}}$. Equation (4.7) is nothing but the displacement patch test condition. It implies that when the solution field is linear, represented by $\boldsymbol{p}_1 \in \mathbb{P}_1(E)^3$, the discrete bilinear form is exactly the same as the continuous bilinear form. Here, $\mathbb{P}_1(E)^3$ is the three-dimensional space spanned by linear polynomials.

2. **Stability**:
For the discrete solution, $\boldsymbol{v}_h$, to be stable, we also need to ensure that the discrete bilinear form is continuous and bounded. Thus, we require that there exist constants $\alpha^*, \alpha_* > 0$ such that:

$$\alpha_* a^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \leq a_h^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \leq \alpha^* a^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \quad \forall E, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E \tag{4.8}$$

This property ensures stability of the numerical solution with mesh refinement.

Under the assumptions (4.6), (4.7), and (4.8), the unique VEM discrete solution, $\boldsymbol{u}_h$, satisfies:

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{\boldsymbol{\mathcal{V}}} \leq c \left(\|\boldsymbol{u} - \boldsymbol{u}_I\|_{\boldsymbol{\mathcal{V}}} + \|\boldsymbol{u} - \boldsymbol{u}_\pi\|_{h,\boldsymbol{\mathcal{V}}} + \sup_{\boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h} \frac{|\ell(\boldsymbol{w}_h) - \ell_h(\boldsymbol{w}_h)|}{\|\boldsymbol{w}_h\|_{\boldsymbol{\mathcal{V}}}} \right) \tag{4.9}$$

Inequality (4.9) is true for every approximation $\boldsymbol{u}_I \in \boldsymbol{\mathcal{V}}_h$, and $\boldsymbol{u}_\pi$ of $\boldsymbol{u}$ that is piecewise in $\mathbb{P}_1^3$. For a detailed proof of (4.9), refer to [27]. We can see from the estimate (4.9) that the discrete solution is bounded by a positive quantity which decreases with mesh refinement resulting in convergence of the discrete solution, $\boldsymbol{u}_h$, to the exact solution, $\boldsymbol{u}$.

4.1.3 Projection and the discrete bilinear form

To incorporate the consistency condition into the discrete bilinear form, let us define a projection operator Π^E as:

$$\begin{gathered} \Pi^E : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1(E)^3 \\ a^E(\Pi^E \boldsymbol{v}_h, \boldsymbol{p}_1) = a^E(\boldsymbol{v}_h, \boldsymbol{p}_1) \quad \text{and} \quad \Pi^E \boldsymbol{p}_1 = \boldsymbol{p}_1, \qquad \forall \boldsymbol{p}_1 \in \mathbb{P}_1(E)^3 \end{gathered} \tag{4.10}$$

Note that the projection $\Pi^E \boldsymbol{u}_h$ for $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h^E$ can be computed, since it is expressed as a combination of linear polynomials and $\boldsymbol{u}_h$ can be computed exactly on element edges using the discrete fields sampled at the degrees of freedom. Now, let us select the discrete bilinear form as:

$$a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = a^E(\Pi^E \boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) \tag{4.11}$$

This choice ensures that the consistency condition (4.7) is satisfied, because it follows from the first part of the definition of projection, Π^E, (4.10) that the right hand side of (4.11) is equivalent to

$$a^E(\Pi^E \boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) = a^E(\boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) \tag{4.12}$$

since, $\Pi^E \boldsymbol{v}_h \in \mathbb{P}_1(E)^3$. Further, if $\Pi^E \boldsymbol{v}_h = \boldsymbol{v}_h = \boldsymbol{p}_1$, which is the second part of the definition of the projection Π^E, we can see that the consistency condition is satisfied. However, the choice (4.11) does not guarantee the stability condition (4.8). We need an additional term in (4.11) to satisfy stability requirements. We have seen that we can use quantities which belong to $\mathbb{P}_1(E)^3$, to compute (4.11). Now, let us consider the space of higher order deformation modes, $\overline{\boldsymbol{\mathcal{V}}}_h^E$ which is spanned by the set of basis that belong to $\boldsymbol{\mathcal{V}}_h^E$ excluding $\mathbb{P}_1(E)^3$, i.e. $\overline{\boldsymbol{\mathcal{V}}}_h^E = \boldsymbol{\mathcal{V}}_h^E \backslash \mathbb{P}_1(E)^3$. Assuming we have the energy orthogonality

$$a_h^E(\boldsymbol{p}_1, \overline{\boldsymbol{v}}_h) = a^E(\boldsymbol{p}_1, \overline{\boldsymbol{v}}_h) = 0 \qquad \forall \boldsymbol{p}_1 \in \mathbb{P}_1(E)^3, \qquad \overline{\boldsymbol{v}}_h \in \overline{\boldsymbol{\mathcal{V}}}_h^E \tag{4.13}$$

then the additional term of the discrete bilinear form can be taken as $a^E(\overline{\boldsymbol{u}}_h, \overline{\boldsymbol{v}}_h)$, for $\overline{\boldsymbol{u}}_h, \overline{\boldsymbol{v}}_h \in \overline{\boldsymbol{\mathcal{V}}}_h^E$. The idea here is to construct the projection Π^E in such a way that the arguments in the consistency term span the $\mathbb{P}_1(E)^3$ space and the arguments in the stability term span the $\overline{\boldsymbol{\mathcal{V}}}_h^E$ space. In other words, the following identity holds:

$$a^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = a^E(\Pi^E \boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) + a^E((\mathbf{I} - \Pi^E)\boldsymbol{u}_h, (\mathbf{I} - \Pi^E)\boldsymbol{v}_h), \qquad \boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E \tag{4.14}$$

which is nothing but the Pythagoras theorem when $\boldsymbol{u}_h = \boldsymbol{v}_h$. But, its unclear how to compute the stability term we discussed above. So we pick an easily computable bilinear form which is any symmetric positive definite matrix and satisfies

$$c_0 a^E(\overline{\boldsymbol{v}}_h, \overline{\boldsymbol{v}}_h) \leq S^E(\overline{\boldsymbol{v}}_h, \overline{\boldsymbol{v}}_h) \leq c_1 a^E(\overline{\boldsymbol{v}}_h, \overline{\boldsymbol{v}}_h) \qquad \forall E, \quad \forall \overline{\boldsymbol{v}}_h \in \overline{\boldsymbol{\mathcal{V}}}_h^E \tag{4.15}$$

Now, the discrete bilinear form is written as:

$$a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = a^E(\Pi^E \boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) + S^E((\mathbf{I} - \Pi^E)\boldsymbol{u}_h, (\mathbf{I} - \Pi^E)\boldsymbol{v}_h), \qquad \boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E \tag{4.16}$$

The final expression for the discrete bilinear form (4.16) still holds the consistency condition and also satisfies the stability condition which can be shown as follows. For $\boldsymbol{p}_1 \in \mathbb{P}_1(E)^3$, the stability term reduces to $S^E((\mathbf{I} - \Pi^E)\boldsymbol{p}_1, (\mathbf{I} - \Pi^E)\boldsymbol{v}_h) = 0$. Hence, in (4.16) we are only left with the first term, which satisfies the consistency condition. Also, using (4.14) and

(4.15), we can show that the stability requirement is also met.

The Virtual Element Method has strong ties with the hourglass stabilization approaches discussed earlier. Both approaches share similarities in terms of construction of the stiffness matrix based on splitting of discrete space into two a^E-orthogonal spaces. Now, consider a rectangular brick element. The stiffness matrix obtained from one-point Gauss quadrature using B8 shape functions is exactly identical to the consistency term of the VEM stiffness matrix. Moreover, both approaches add an additional term to the discrete bilinear form to introduce stability to the solutions. The order of magnitude of the stability term is important because a large magnitude may result in locking of the elements and a small magnitude may not have the desired stabilization effect. The hourglass stabilization approaches discussed earlier [117, 123, 82, 35], were developed specifically for hexahedral elements. The Virtual Element Method imposes no such restriction on the shape of the element.

4.2 Implementation and practical aspects

In this section, we derive the terms in the stiffness matrix and the force vector explicitly, based on the concepts discussed in the previous section. The derivations provided here are motivated by the ones available in [173].

4.2.1 Construction of local stiffness matrix

First, in order to derive the expressions for the stiffness matrix, the basis vectors which span the $\mathbb{P}_1(E)^3$ space need to be selected. One feasible choice is:

$$\begin{aligned}\mathbb{P}_1(E)^3 &= \text{span}\left(\hat{\boldsymbol{p}}_1, \hat{\boldsymbol{p}}_2, ..., \hat{\boldsymbol{p}}_\beta, ..., \hat{\boldsymbol{p}}_{12}\right) \\ &= \text{span}\left(\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}y\\-x\\0\end{bmatrix}, \begin{bmatrix}0\\z\\-y\end{bmatrix}, \begin{bmatrix}-z\\0\\x\end{bmatrix}, \begin{bmatrix}x\\0\\0\end{bmatrix}, \begin{bmatrix}0\\y\\0\end{bmatrix}, \begin{bmatrix}0\\0\\z\end{bmatrix}, \begin{bmatrix}y\\x\\0\end{bmatrix}, \begin{bmatrix}0\\z\\y\end{bmatrix}, \begin{bmatrix}z\\0\\x\end{bmatrix}\right)\end{aligned} \tag{4.17}$$

The strains corresponding to the first six bases are zero, i.e. in Voigt notation the strains are: $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_i) = (0,0,0,0,0,0)^T$ for $i = 1, ..., 6$, which indicates that they span the rigid body space of three rotational and three translational degrees of freedom. The strains of the remaining basis elements are: $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_7) = (1,0,0,0,0,0)^T$, $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_8) = (0,1,0,0,0,0)^T$, $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_9) = (0,0,1,0,0,0)^T$, $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_{10}) = (0,0,0,1,0,0)^T$, $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_{11}) = (0,0,0,0,1,0)^T$, and $\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_{12}) = (0,0,0,0,0,1)^T$. In order to aid in future derivations, let us assume that the linear space $\mathbb{P}_1(E)^3$ can be split as follows:

$$\mathbb{P}_1(E)^3 = \mathbb{P}_1^0(E)^3 \oplus \mathbb{P}_1^1(E)^3 \tag{4.18}$$

where $\mathbb{P}_1^0(E)^3$ represents the rigid body space (RBS) spanned by bases $\hat{\boldsymbol{p}}_i$ for $i = 1, ..., 6$ and $\mathbb{P}_1^1(E)^3$ is the remainder of the linear space. Figures 4.2 and 4.3 illustrate the linear polynomial bases on hexahedral and polyhedral element, respectively.

Consistency term

To build the first term of the discrete bilinear form, the projection $\Pi^E : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1(E)^3$ needs to be constructed. Since the discrete bilinear form involves calculating the strains of the arguments, the contribution of the rigid body terms is lost as they have zero strains. So, we treat the rigid body motions separately and split the projection Π^E as follows:

$$\Pi^E = \bar{\Pi}^E + \Pi^0(\mathbf{I} - \bar{\Pi}^E) \tag{4.19}$$

Here, $\bar{\Pi}^E : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1^1(E)^3$ and $\Pi^0 : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1^0(E)^3$. Note that $a^E(\Pi^E \boldsymbol{u}_h, \Pi^E \boldsymbol{v}_h) = a^E(\bar{\Pi}^E \boldsymbol{u}_h, \bar{\Pi}^E \boldsymbol{v}_h)$. Thus, the consistency term requires determining the projection $\bar{\Pi}^E$. Consider a function $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h^E$ expressed in terms of Lagrangian bases, ϕ_i:

$$\boldsymbol{u}_h = \sum_{i=1}^{3n} \boldsymbol{\phi}_i u_i \tag{4.20}$$

where, u_i are the nodal dof and $\boldsymbol{\phi}_{3i-2} = [\phi_i, 0, 0]^T$, $\boldsymbol{\phi}_{3i-1} = [0, \phi_i, 0]^T$, $\boldsymbol{\phi}_{3i} = [0, 0, \phi_i]^T$. As $\bar{\Pi}^E \boldsymbol{\phi}_i \in \mathbb{P}_1^1(E)^3$, $\bar{\Pi}^E \boldsymbol{\phi}_i$ can be written in terms of the basis functions $\hat{\boldsymbol{p}}_\alpha \in \mathbb{P}_1^1(E)^3$ as:

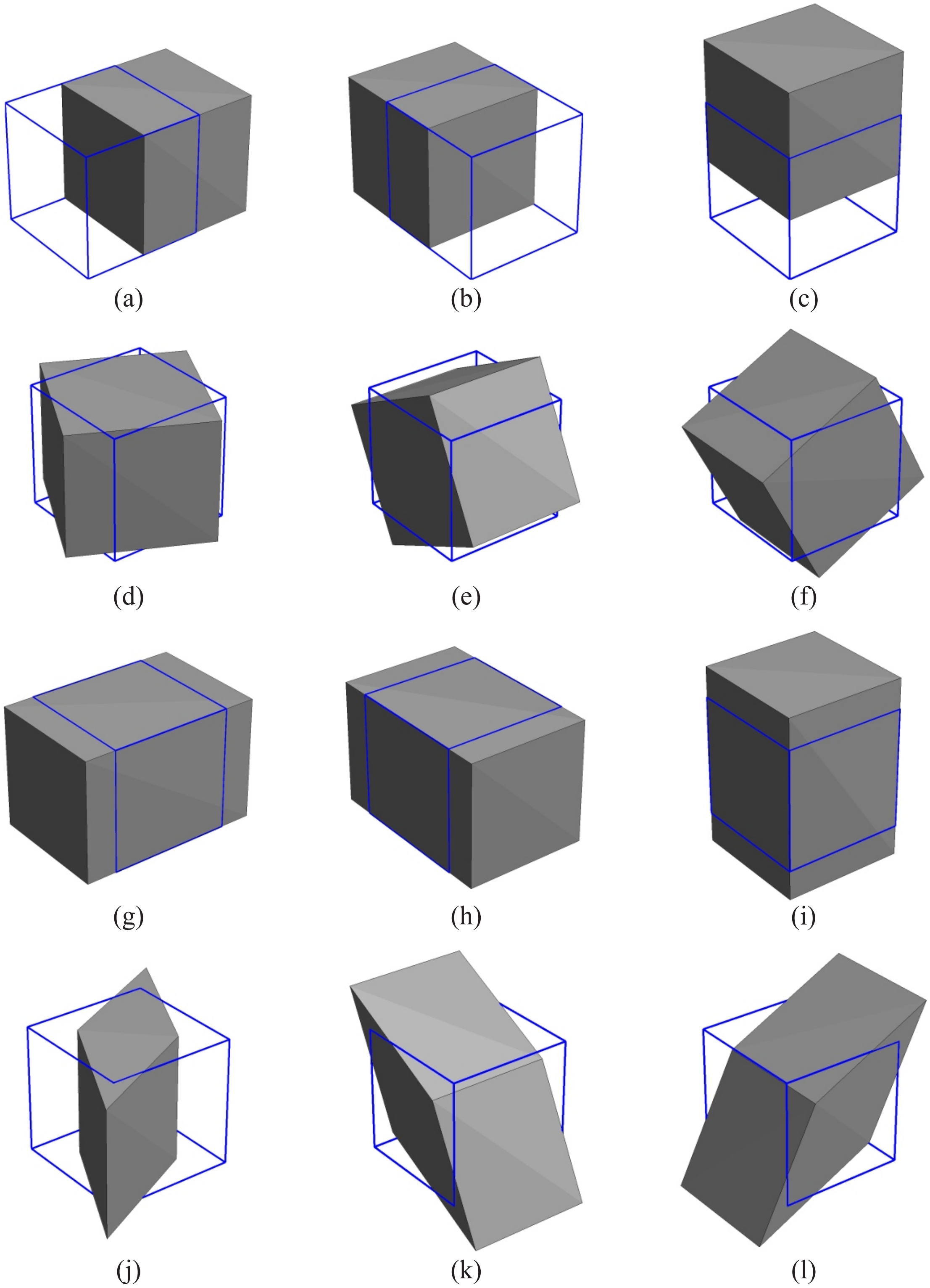
$$\bar{\Pi}^E \boldsymbol{\phi}_i = \sum_\beta s_{i\beta} \hat{\boldsymbol{p}}_\beta \tag{4.21}$$

Using the definition of projection (4.10) and (4.21), the expression $a^E(\bar{\Pi}^E \boldsymbol{\phi}_i, \hat{\boldsymbol{p}}_\alpha) = a^E(\boldsymbol{\phi}_i, \hat{\boldsymbol{p}}_\alpha)$, is written as:

$$a^E(\bar{\Pi}^E \boldsymbol{\phi}_i, \hat{\boldsymbol{p}}_\alpha) = \sum_\beta s_{i\beta} a^E(\hat{\boldsymbol{p}}_\beta, \hat{\boldsymbol{p}}_\alpha) = a^E(\boldsymbol{\phi}_i, \hat{\boldsymbol{p}}_\alpha) \tag{4.22}$$

Let $\mathbf{R}$ represent (4.22) written in matrix form. Now, using Green's first identity, for $\boldsymbol{p}_1 \in \mathbb{P}_1(E)^3, \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E$, we express the bilinear term as:

$$\begin{aligned} a^E(\boldsymbol{v}_h, \boldsymbol{p}_1) &= \int_E \mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{v}_h) : \boldsymbol{\varepsilon}(\boldsymbol{p}_1)\, d\boldsymbol{x} \\ &= -\int_E \boldsymbol{v}_h \cdot \mathbf{div}\left(\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{p}_1)\right)\, d\boldsymbol{x} + \int_{\partial E} \boldsymbol{v}_h \cdot \left(\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{p}_1)\right)\mathbf{n}\, d\boldsymbol{s} \\ &= \int_{\partial E} \boldsymbol{v}_h \cdot \left(\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{p}_1)\right)\mathbf{n}\, d\boldsymbol{s} \end{aligned} \tag{4.23}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.2: Illustration of polynomial basis representing rigid body translations (a) $\boldsymbol{p}_1$ (b) $\boldsymbol{p}_2$ (c) $\boldsymbol{p}_3$, rigid body rotations (d) $\boldsymbol{p}_4$ (e) $\boldsymbol{p}_5$ (f) $\boldsymbol{p}_6$, axial strains (g) $\boldsymbol{p}_7$ (h) $\boldsymbol{p}_8$ (i) $\boldsymbol{p}_9$ and shear strains (j) $\boldsymbol{p}_{10}$ (k) $\boldsymbol{p}_{11}$ (l) $\boldsymbol{p}_{12}$, on a hexahedral element.

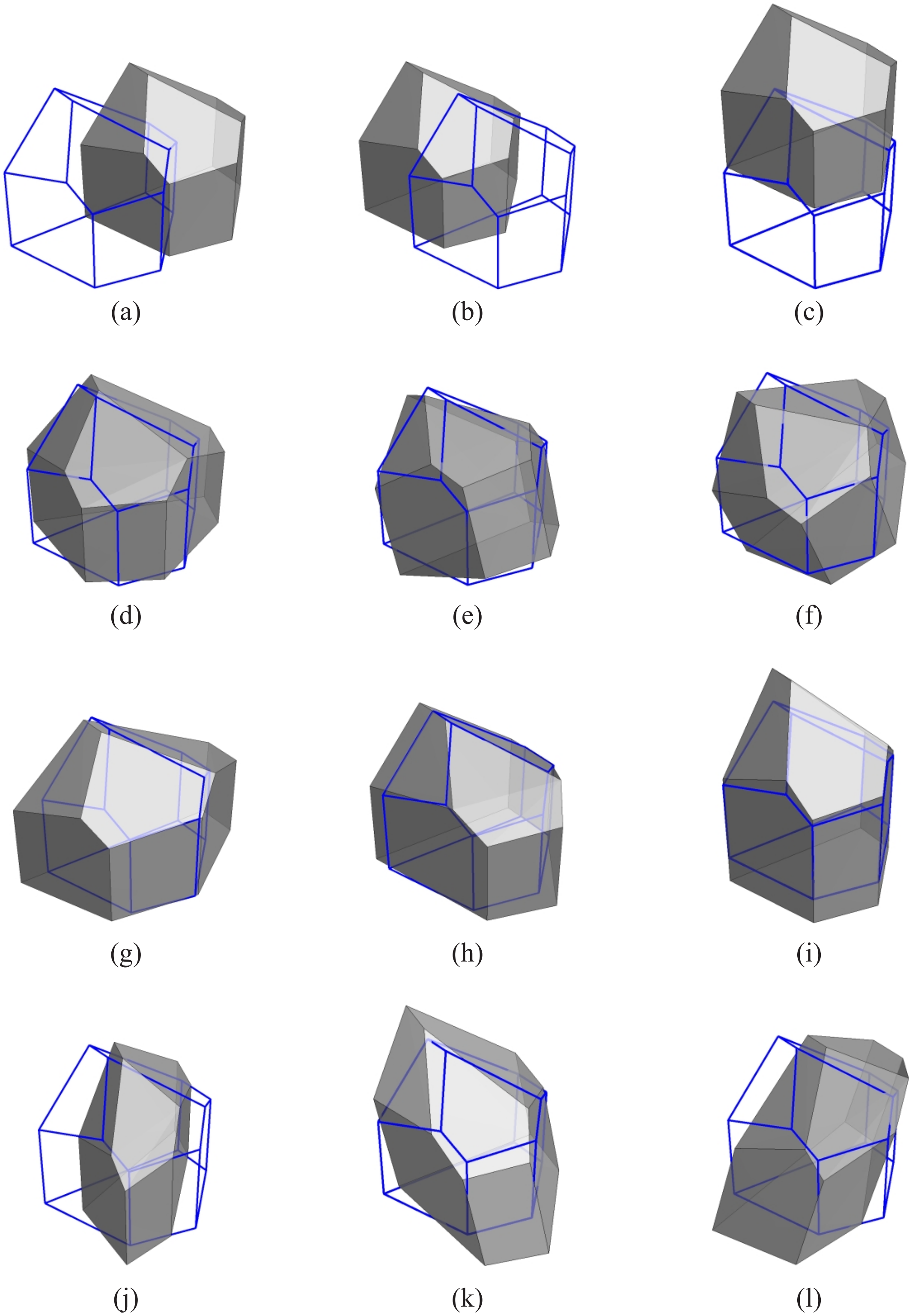
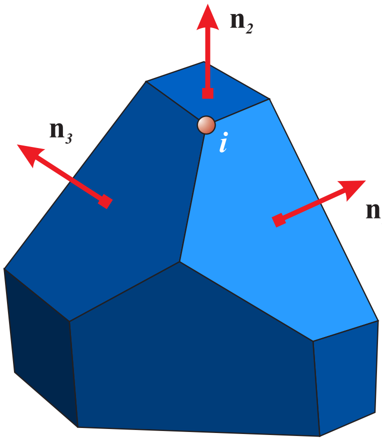

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 4.3: Illustration of polynomial basis representing rigid body translations (a) $\boldsymbol{p}_1$ (b) $\boldsymbol{p}_2$ (c) $\boldsymbol{p}_3$, rigid body rotations (d) $\boldsymbol{p}_4$ (e) $\boldsymbol{p}_5$ (f) $\boldsymbol{p}_6$, axial strains (g) $\boldsymbol{p}_7$ (h) $\boldsymbol{p}_8$ (i) $\boldsymbol{p}_9$ and shear strains (j) $\boldsymbol{p}_{10}$ (k) $\boldsymbol{p}_{11}$ (l) $\boldsymbol{p}_{12}$, on a polyhedral element.

Using (4.23), matrix $\mathbf{R}$ is computed as follows:

$$\begin{aligned}\mathrm{R}_{3i-2:3i,:} = a^E(\boldsymbol{\phi}_i, \hat{\boldsymbol{p}}_\alpha) &= \int_{\partial E} \boldsymbol{\phi}_i \cdot (\mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\alpha))\, \mathbf{n}\, d\boldsymbol{s}, \qquad i = 1, ..., N \\ &= \sum_{j=1}^{N_{neighbor}} \left((\mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\alpha))\, \mathbf{n}_j \cdot \int_{\partial E} \boldsymbol{\phi}_i \, d\boldsymbol{s}_j \right) \end{aligned} \tag{4.24}$$

Figure 4.4: Illustration of the adjacency information needed in the consistency term evaluation. The shape function corresponding to node i is integrated over the three faces containing it. The arrows represent the outward unit normals, $\mathbf{n}_i$.



In (4.24), the summation of the integral of the Lagrangian shape function $\boldsymbol{\phi}_i$ is carried out over all faces j which share node i, as illustrated in Fig. 4.4. Now, the basis $\hat{\boldsymbol{p}}_\beta$ can be written in terms of Lagrangian bases as:

$$\hat{\boldsymbol{p}}_\beta = \sum_{i=1}^{3n} \boldsymbol{\phi}_i \mathrm{N}_{i\beta} \tag{4.25}$$

where $\mathrm{N}_{i\beta}$ is the nodal coordinate matrix of dimension $3N \times 6$, and N is the total number of nodes in an element and is expressed as:

$$\mathrm{N}_{3i-2:3i,:} = \begin{bmatrix} x_i & 0 & 0 & y_i & 0 & z_i \\ 0 & y_i & 0 & x_i & z_i & 0 \\ 0 & 0 & z_i & 0 & y_i & x_i \end{bmatrix}, \qquad i = 1, ..., N \tag{4.26}$$

Thus, we have all the ingredients to construct the consistency part of the stiffness matrix. Using (4.22), (4.24) and (4.25), we show that

$$\sum_\beta s_{i\beta} a^E(\hat{\boldsymbol{p}}_\beta, \hat{\boldsymbol{p}}_\alpha) = \sum_\beta s_{i\beta} \mathrm{N}_{i\beta} \mathrm{R}_{i\alpha} = \mathrm{R}_{i\alpha} \tag{4.27}$$

which can be written in matrix form as, $\boldsymbol{S} = [s_{ij}] = \mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}$. Thus, from the definition of projection (4.21) and (4.22), the matrix form, $\mathbf{P}_M$, of the projection $\bar{\Pi}^E$ can be expressed as $\left[\mathbf{N}(\mathbf{R}^T\mathbf{N})^{-1}\mathbf{R}^T\right]$, and the consistency term, $\boldsymbol{K}_1$, can be written as:

$$a^E(\bar{\Pi}^E\boldsymbol{\phi}_i, \bar{\Pi}^E\boldsymbol{\phi}_j) = a^E(\bar{\Pi}^E\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \sum_\beta s_{i\beta}a^E(\hat{\boldsymbol{p}}_\beta, \boldsymbol{\phi}_j) = \left[\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\mathbf{R}^T\right]_{ij} \tag{4.28}$$

Stability term

The discrete bilinear term contains a second term, $S^E((\mathbf{I}-\Pi^E)\boldsymbol{\phi}_i, (\mathbf{I}-\Pi^E)\boldsymbol{\phi}_j)$, which imparts stability to the solution. Using (4.19), we can write $(\mathbf{I}-\Pi^E) = (\mathbf{I}-\bar{\Pi}^E)(\mathbf{I}-\Pi^0)$. We need to construct and compute the projection $\Pi^0 : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1^0(E)^3$ (projection onto the rigid body space) in order to compute the stability term. The projection Π^0 is defined as:

$$\Pi^0\boldsymbol{v} = \bar{\boldsymbol{v}} + \left\langle \frac{1}{|E|}\int_E \nabla\boldsymbol{v}d\boldsymbol{x}\right\rangle(\boldsymbol{x}-\bar{\boldsymbol{x}}) \tag{4.29}$$

where, $\bar{\boldsymbol{v}}$ and $<>$ are the mean of the vector and skew symmetric part of the tensor, respectively and $\bar{\boldsymbol{x}}$ is the mean of the nodal coordinates of the element. The projection Π^0 in (4.29) is chosen such that, if $\boldsymbol{v} \in \mathbb{P}_1^0(E)^3$ (i.e. $\boldsymbol{v} = \sum_{i=1}^6 c_i\hat{\boldsymbol{p}}_i$ where c_i's are constants) then $\Pi^0\boldsymbol{v} = \boldsymbol{v}$. Hence, the stability term, represented by $\boldsymbol{K}_2$, is expressed as:

$$S^E((\mathbf{I}-\Pi^E)\boldsymbol{\phi}_i, (\mathbf{I}-\Pi^E)\boldsymbol{\phi}_j) = (\mathbf{I}-\mathbf{P}_M^T)(\mathbf{I}-\mathbf{P}_1^T)S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)(\mathbf{I}-\mathbf{P}_1)(\mathbf{I}-\mathbf{P}_M) \tag{4.30}$$

where $\mathbf{P}_1$ is the matrix representation of the projection operator Π^0. The final piece in the construction of the stability term is the choice of $S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$. It should be a positive definite matrix and should scale like the consistency term. Let us choose $S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$ as $\alpha\delta_{ij}$, where α is a scaling factor. Thus, the final stiffness matrix expression is:

$$a_h^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \boldsymbol{K}_{ij} = \left[\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\mathbf{R}^T\right]_{ij} + \left[\alpha(\mathbf{I}-\mathbf{P}_M^T)(\mathbf{I}-\mathbf{P}_1^T)(\mathbf{I}-\mathbf{P}_1)(\mathbf{I}-\mathbf{P}_M)\right]_{ij} \tag{4.31}$$

In this work, the scaling factor α is chosen as $\alpha^*\ trace(\boldsymbol{K}_1)$, where α^* is a scaling coefficient chosen based on a parametric study conducted in section 4.3.2. Other choices of the scaling factor, which satisfy the requirement discussed above, are possible.

4.2.2 Surface integration schemes

To evaluate the surface integral in (4.24), we encountered during the derivation of the stiffness matrix, we use two schemes - nodal quadrature and conforming interpolant quadrature.

Nodal quadrature scheme

The nodal quadrature described here is first order accurate. Consider a Lagragian basis, ϕ_i, corresponding to the node i. We define the surface integral of ϕ_i over any face of the polyhedral element E using nodal quadrature as:

$$\int_{\partial E} \phi_i d\boldsymbol{s} = \phi_i(\boldsymbol{x}_i)\ A_i = A_i \tag{4.32}$$

where A_i is the nodal weight of node i, which is calculated as follows. Consider any face of a polyhedral element (Fig. 4.5). The nodal weight for each node on the face is taken as the area of the quadrilateral formed by the node, the centroid of the face, and mid-points of the edges containing the node. For a regular hexahedron, nodal quadrature exactly integrates (4.24) and each nodal weight is equal to a quarter of the area of the face.

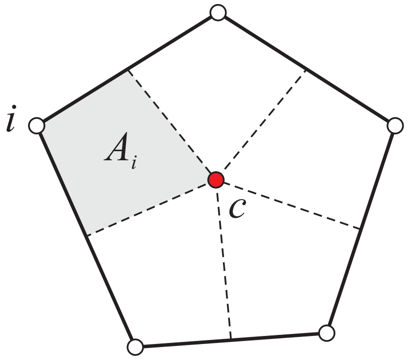


Figure 4.5: Nodal quadrature scheme for surface integration. The variables c and A_i represent the centroid of the polygon and the nodal weight associated with the node i on any face, respectively.

For concave polygonal faces, the nodal quadrature scheme is valid only for polygons of type (a), shown in Fig. 4.6, which are star-convex with respect to their centroid. A polygon is considered star-convex with respect to a specific point, c, if the line segment connecting the point c to any point on the boundary of the polygon does not intersect the boundary at any other point [40]. Concave polygons shown in Fig. 4.6(b) and (c), are non-star-convex. The star convexity constraint avoids inaccuracies in the computation of the nodal weights. For Galerkin mesh-free methods conforming nodal integration schemes are available [66, 65] which could be potentially be explored for the VEM.

Conforming interpolant quadrature

The conforming Galerkin approximations satisfy the Kronecker-delta property, i.e. the Lagrangian shape functions for a particular node takes unit value at the location of the node and goes to zero at all other nodes in the element. For three-dimensional polyhedrons, this implies that, on a particular face of a polyhedral element, only the shape functions associated with the nodes lying on that face contribute to the interpolation of a function. Taking advantage of this fact, (4.24) can be evaluated with any order of accuracy on polygonal faces of any shape using any of the well known polygonal shape functions available in the literature such as Wachspress [200, 210], mean value coordinates [84], natural neighbor

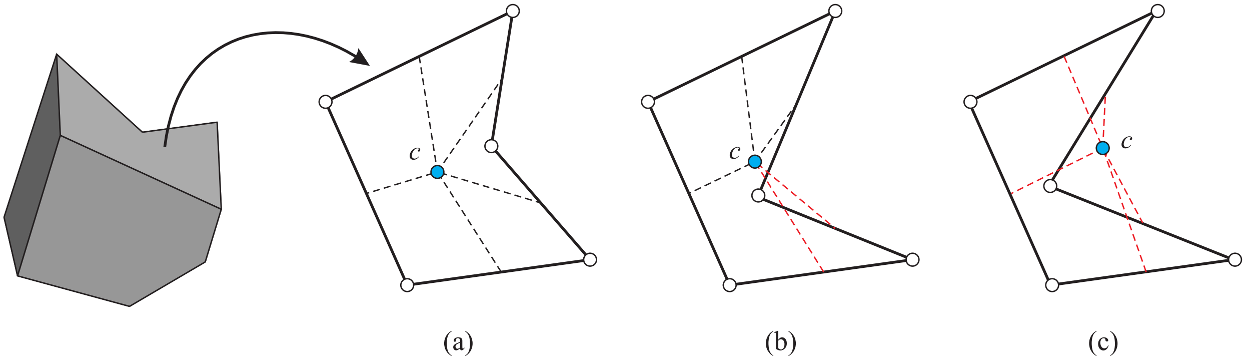


Figure 4.6: Nodal quadrature scheme for surface integrals is valid only for concave polygons of type (a) only, which are star-convex [40] with respect to centroid c.

[199] or MAXENT [196, 19]. In this work, we choose the mean value coordinates due to their robustness. Before conducting the numerical integration, the planar polygonal face in three-dimensional domain is rotated to a two-dimensional plane. The proposed scheme is illustrated in Fig. 4.7. Integration on the polygonal face is conducted by partitioning it into triangles, by connecting the centroid to the vertices, and then using standard quadrature rules of a triangle. Alternatively, special numerical quadrature rules developed for polygonal domains can be used [135, 146, 145]. The conforming interpolant quadrature scheme using mean value coordinates is valid for all concave polygons (even non-star-convex).

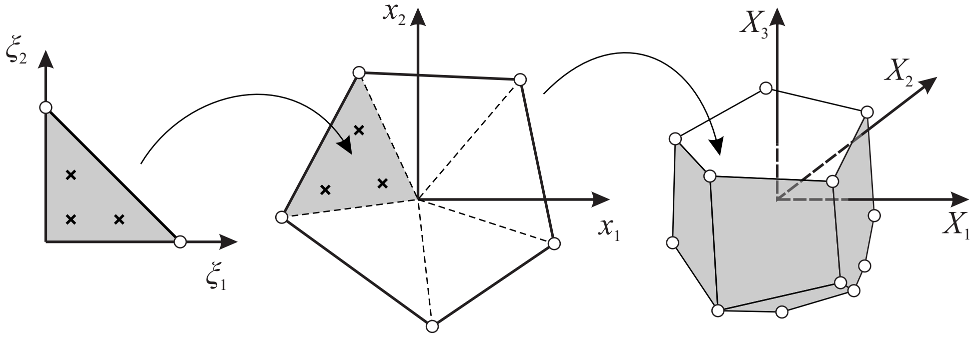


Figure 4.7: Conforming interpolation quadrature for surface integration. The planar polygonal faces in three-dimensional domain are rotated to a two-dimensional plane before conducting the numerical integration. Subsequently, integration on the polygonal face is conducted by partitioning it into triangles, by connecting the centroid to the vertices, and then using standard quadrature rules of a triangle.

4.2.3 Load vector

Solution of the discrete elasticity equations (4.4) results in a system of equations, written in matrix form as **KU**=**F**. So far, we have dealt with the construction of the stiffness matrix **K**. In this section, we discuss how to handle the right hand side vector, **F**. The global load vector $\mathbf{F} = \ell_h(\boldsymbol{\phi})$ can split as $\mathbf{F} = \sum_E \mathbf{F}^E = \sum_E \ell_h^E(\boldsymbol{\phi})$. For point loads, the load vector

$\mathbf{F}$ consists of all zeros except at degrees of freedom corresponding to the point loads where it takes the value of the load. For surface loads, in order to obtain equivalent nodal loads, surface integration schemes discussed in section 4.2.2 can be used. To handle volumetric or body forces a volumetric nodal quadrature can be used, due to the lack of explicit expressions for the shape functions in VEM. A volumetric nodal quadrature, similar to surface nodal quadrature (section 4.2.2), is approximately first order accurate. Thus, for an element E subjected to body forces $\boldsymbol{b}$, the equivalent nodal loads, $\mathbf{F}^E$, are given by:

$$\begin{aligned}\mathbf{F}^E = \int_E \boldsymbol{\phi}\boldsymbol{b}d\boldsymbol{x} &\approx [\phi_i(\boldsymbol{x}_i)\boldsymbol{b}(\boldsymbol{x}_i)w_i]^T, \qquad \forall i = 1 \text{ to } n \\ &= [\boldsymbol{b}(\boldsymbol{x}_1)w_1 \quad \boldsymbol{b}(\boldsymbol{x}_2)w_2, \ldots, \quad \boldsymbol{b}(\boldsymbol{x}_n)w_n]^T \end{aligned} \tag{4.33}$$

Here, ϕ_i and n are the Lagrangian basis functions and number of nodes in the element, respectively. The volumetric nodal weights w_i are obtained by summing up the volumes of the pyramids[1] surrounding each node, as illustrated in Fig. 4.8.

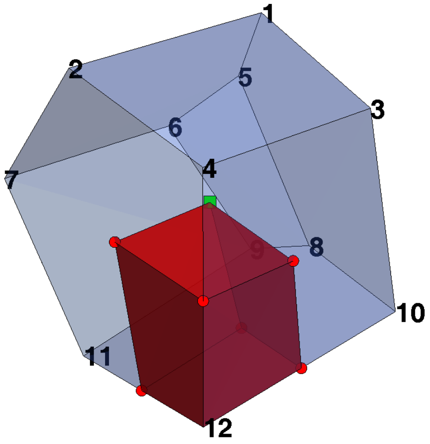


Figure 4.8: Illustration of volumetric load calculation. Red polyhedron (formed using the element center, three edge mid-points, three face centroids and node 12) represents the volumetric weight associated with the node 12.

4.2.4 Numerical verification studies

In order to complement the discussions presented in the previous sections, we conduct an eigenvalue analysis for the generic hexahedral element shown in Fig. 4.9. The Young's modulus and Poisson's ratio are chosen as 25 and 0.3, respectively, for all the examples in this work and consistent units are employed.

The eigenvalues of the stiffness matrix corresponding to finite element analysis using B8 shape functions are $\boldsymbol{\lambda} = \{0, 0, 0, 0, 0, 0, 1.47, 3.11, 3.55, 4.28, 4.82, 6.04, 6.44, 6.98, 8.75, 9.85, 11.50, 12.75, 13.34, 14.19, 15.06, 16.24, 16.62, 45.91\}$. The same problem is solved with the VEM using a scaling factor of 0.1 $trace(\boldsymbol{K}_1)$. For surface integration both nodal integration

---

[1] A polyhedron is divided into pyramids using polyhedron's vertices, edge mid-points, face centroids and polyhedral centroid [188].

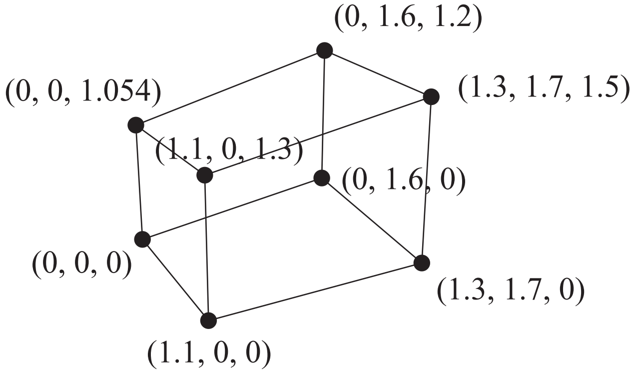


Figure 4.9: A generic hexahedral element for the eigenvalue test.

schemes and conforming interpolant quadrature (third order) are used. The eigenvalues of the stiffness matrix for the nodal integration scheme are $\boldsymbol{\lambda} = \{0, 0, 0, 0, 0, 0, 9.77, 10.52, 11.08, 11.10, 11.12, 11.25, 11.25, 11.25, 11.25, 11.25, 11.25, 11.32, 11.32, 12.24, 13.57, 16.21, 16.42, 45.89\}$ and for the conforming interpolant quadrature scheme are $\boldsymbol{\lambda} = \{0, 0, 0, 0, 0, 0, 9.71, 10.48, 11.06, 11.11, 11.13, 11.25, 11.26, 11.26, 11.26, 11.27, 11.27, 11.32, 11.40, 12.30, 13.58, 16.22, 16.42, 45.90\}$. The hexahedron has 8 nodes, hence 24 degrees of freedom, six of which represent the rigid body motion. Similar to the FEM, the VEM is able to capture six zero eigenvalues corresponding to the physical rigid-body modes and $\lambda_i > 0$, for $i = 7 - 24$, which indicates that the stiffness matrix is full rank and devoid of any spurious modes. Finite element analysis differentiates between different non-rigid body deformation modes, as indicated by the distinct eigenvalues, where as the VEM makes less of a distinction. Also, the maximum and minimum eigenvalues are the same for both methods.

4.3 Numerical examples

In this section, we evaluate the accuracy of the method and implementation using two test cases - displacement patch test and shear loaded cantilever beam bending. The accuracy and convergence of the numerical results are verified in terms of relative L^2 error norms. The displacement error norm is one of the metric used here and is expressed as:

$$U_{Err} = \sqrt{\frac{\left(\mathbf{U}_{exact} - \mathbf{U}_{VEM}\right)^T \boldsymbol{W} \left(\mathbf{U}_{exact} - \mathbf{U}_{VEM}\right)}{\mathbf{U}_{exact}^T \boldsymbol{W} \mathbf{U}_{exact}}} \tag{4.34}$$

where $\mathbf{U}_{exact}$ and $\mathbf{U}_{VEM}$ are the exact and VEM solutions, respectively. The diagonal matrix $\boldsymbol{W}$ consists of volumetric nodal weights w_i as its diagonal entries (refer to section 4.2.3 for the procedure to calculate w_i). We also use the stress error norm to measure the accuracy of the method. In the VEM, since we do not have explicit expressions for Lagrangian shape functions, we can not calculate the stresses point-wise inside each element. The next best approach is to calculate stresses in an average/weak sense. From the definition of the projection, $\bar{\Pi}$, (4.10) and some algebraic manipulation, we see that

$$\int_E \mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{u}_h)d\boldsymbol{x} = \int_E \mathbf{C}\boldsymbol{\varepsilon}(\bar{\Pi}^E\boldsymbol{u}_h)d\boldsymbol{x} \tag{4.35}$$

Thus, using (4.20), (4.21), (4.27) and (4.35), the VEM average element stresses, $\bar{\boldsymbol{\sigma}}_{VEM,\,e}$, are calculated as:

$$\begin{aligned} \bar{\boldsymbol{\sigma}}_{VEM,\,e} &= \mathbf{C}\boldsymbol{\varepsilon}(\bar{\Pi}^E\boldsymbol{u}_h) \\ &= \boldsymbol{u}_e^T\left(\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\right)\mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\beta) \end{aligned} \tag{4.36}$$

The error measure that we use for the stresses, $\boldsymbol{\sigma}$ (written in Voigt notation), is defined as:

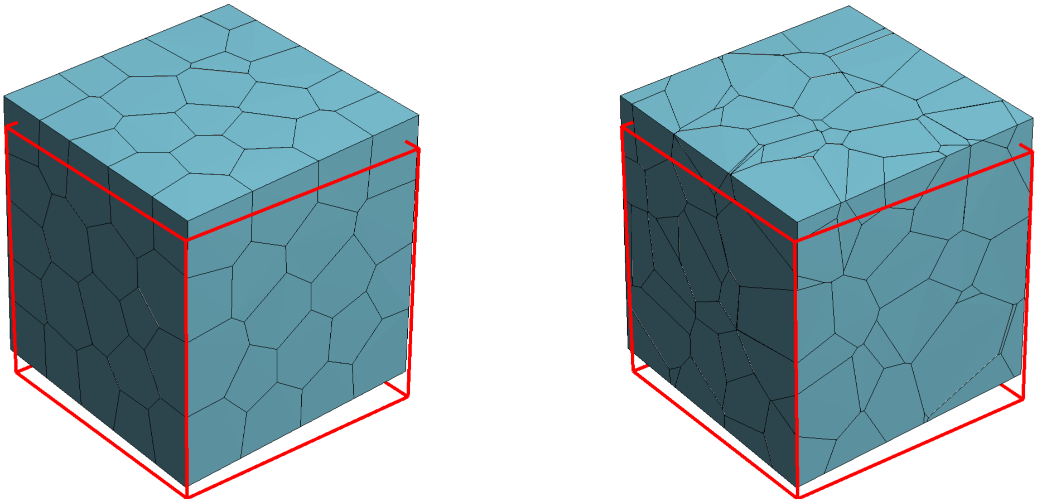
$$\bar{\sigma}_{Err} = \frac{|\boldsymbol{E}_1|_2}{|\boldsymbol{E}_2|_2} \tag{4.37}$$

Matrices $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ are of dimension (Number of elements) × 6 and constructed such that each row of $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ corresponds to $\int_E \left(\boldsymbol{\sigma}_{exact} - \bar{\boldsymbol{\sigma}}_{VEM,\,e}\right)^2 d\boldsymbol{x}$ and $\int_E \left(\boldsymbol{\sigma}_{exact}\right)^2 d\boldsymbol{x}$, respectively, which are numerically integrated over each element. Here, $\boldsymbol{\sigma}_{exact}$ is the exact stress. For numerical integration, each polyhedron is divided into pyramids [188] and the location of Gauss points and their corresponding weights are obtained by mapping from a solid hexahedral isoparameteric element. A fourth order quadrature rule, i.e. 64 Gauss points per pyramid, is used for the integration.

The relative L^2 error norms are plotted against the mean element diameter of the mesh or the number of nodes in the mesh. An element diameter is defined as the maximum pairwise distance of all the vertices of the polyhedron. We use the open source MATLAB toolbox, Multi-Parametric Toolbox (MPT) [126], for generating the polyhedral meshes. Two types of polyhedral meshes are used in the investigations - Centroidal Voronoi Tessellation (CVT) meshes and random Voronoi meshes (RND). The CVT meshes are high quality meshes and are generated by successively replacing the seeds, which are used to generate the Voronoi cells, by the centroids of the Voronoi cells.

4.3.1 Displacement patch test

We start with the displacement patch test on a cube of dimension $1 \times 1 \times 1$, discretized using a different number of polyhedrons (e.g. 50, 100, 200). Both CVT and RND polyhedral meshes are tested. An arbitrary linear displacement of the form $\mathbf{U} = \mathbf{AX} + \mathbf{B}$, where $\mathbf{A} = 1/100\,[2\ 1\ 3;\ 3\ 4\ 2;\ 4\ 3\ 1]$ and $\mathbf{B} = 1/100\,[1\ 2\ 3]^T$, is applied to the nodes on the boundaries of the cube. No forces are applied to the system. For surface integration, both nodal and conforming interpolant quadratures are considered. The stability terms discussed in section 4.2.1; along with different scaling factors of the form $\alpha^* \; trace(\boldsymbol{K}_1)$, with $\alpha^* \in [10^{-3} - 10^{3}]$ are investigated. The relative L^2 displacement error and relative L^2 average stress errors are of the order $10^{-12} - 10^{-15}$ for CVT Voronoi meshes, indicating that the VEM passes the displacement patch test exactly. For random Voronoi meshes the errors are approximately one order of magnitude higher than CVT mesh errors, but still in the machine precision range. The tests indicate that the choice of surface integration scheme, stability term, or scaling factor has no impact on the patch test results. Figure 4.10 illustrates one of the deformed configurations from the patch test for both CVT and random Voronoi meshes.



(a)

(b)

Figure 4.10: Displacement patch test using (a) CVT Voronoi mesh and (b) random Voronoi mesh of 100 polyhedral elements. Blue lines and gray discretized boxes represent the initial configurations and deformed configurations, respectively.

The reason the displacement patch test is passed for the polyhedral meshes, in spite of the approximate evaluation of surface integrations, is as follows. For $\boldsymbol{p}_1 \in \mathbb{P}_1(E)^3$ and $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E$, from (4.7), definition of projection (4.10) and (4.23), we have:

$$a_h^E(\boldsymbol{p}_1, \boldsymbol{v}_h) = a^E(\boldsymbol{p}_1, \Pi^E \boldsymbol{v}_h) = \int_{\partial E} \boldsymbol{v}_h \cdot (\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{p}_1))\, \mathbf{n}\, d\boldsymbol{s} = \int_{\partial E} \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}\, d\boldsymbol{s}$$

The last surface integral can be expressed as the sum of the numerically computed surface integrals over each face of the polyhedron $F \in \partial E$, using the quadrature schemes discussed previously:

$$\int_{\partial E} \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}\, d\boldsymbol{s} = \sum_{F\in\partial E} \left(\int_F \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}\, d\boldsymbol{s} \right) \tag{4.38}$$

Now, for a generic patch test, the exact solution is $\boldsymbol{u} = \boldsymbol{p}_1 \in \mathbb{P}_1(E)^3$. Let tractions $\boldsymbol{t} = \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}$ be imposed on Γ_T and displacements $\boldsymbol{g} = \boldsymbol{p}_1$ applied to the remaining boundary Γ_D. For an arbitrary test function $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h$, we have:

$$\begin{aligned} a_h(\boldsymbol{p}_1, \boldsymbol{v}_h) &= \sum_E a_h^E(\boldsymbol{p}_1, \boldsymbol{v}_h) = \sum_E \sum_{F\in\partial E} \left(\int_F \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}\, d\boldsymbol{s} \right) = \sum_{F\in\Gamma_T} \left(\int_F \boldsymbol{v}_h \cdot \boldsymbol{\sigma}(\boldsymbol{p}_1)\mathbf{n}\, d\boldsymbol{s} \right) \\ &= \ell_h(\boldsymbol{v}_h) \end{aligned} \tag{4.39}$$

In the second last integral, we use the fact that surface integrals on the internal faces cancel each other out and $\boldsymbol{v} = \boldsymbol{0}$ on Γ_D. As $\boldsymbol{p}_1 \in \boldsymbol{\mathcal{V}}_h$, this indicates that $\boldsymbol{u}_h = \boldsymbol{p}_1$ is the unique solution to the discrete problem with prescribed linear displacement boundary conditions. Hence, the global patch test is passed.

We also conducted the displacement patch test on meshes containing concave polyhedral elements. Figure 4.11 shows the deformed configuration of one such mesh. As before, we test both stability terms, along with different scaling factors of the form $\alpha^*\ trace(\boldsymbol{K}_1)$, with $\alpha^* \in [10^{-3} - 10^3]$. Both surface integration schemes are also tested. For all the aforementioned cases, the VEM passes the displacement patch test on meshes containing concave polyhedral elements.

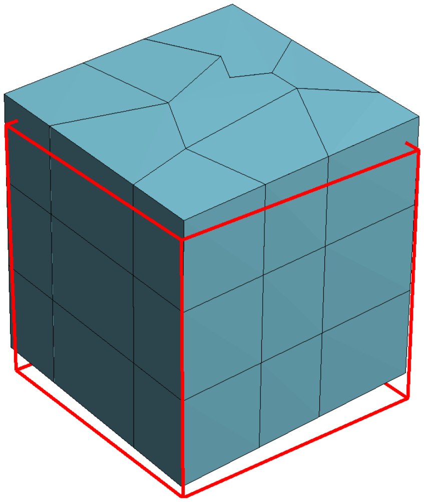


Figure 4.11: Displacement patch test on polyhedral mesh containing concave elements. Blue lines and gray discretized boxes represent the initial configurations and deformed configurations, respectively.

4.3.2 Shear loaded cantilever beam bending

Here, we study the performance of VEM using brick, CVT and random Voronoi (RND) meshes for the shear loaded cantilever beam bending problem. Consider a rectangular beam with a cross-section bounded by planes $x = \pm a$, $y = \pm b$; length bounded by planes $z = 0, c$ and loaded by shear force $F_y = F$ in the negative y direction acting through the origin. The actual expressions for stresses are available in [23] and are repeated here for completeness.

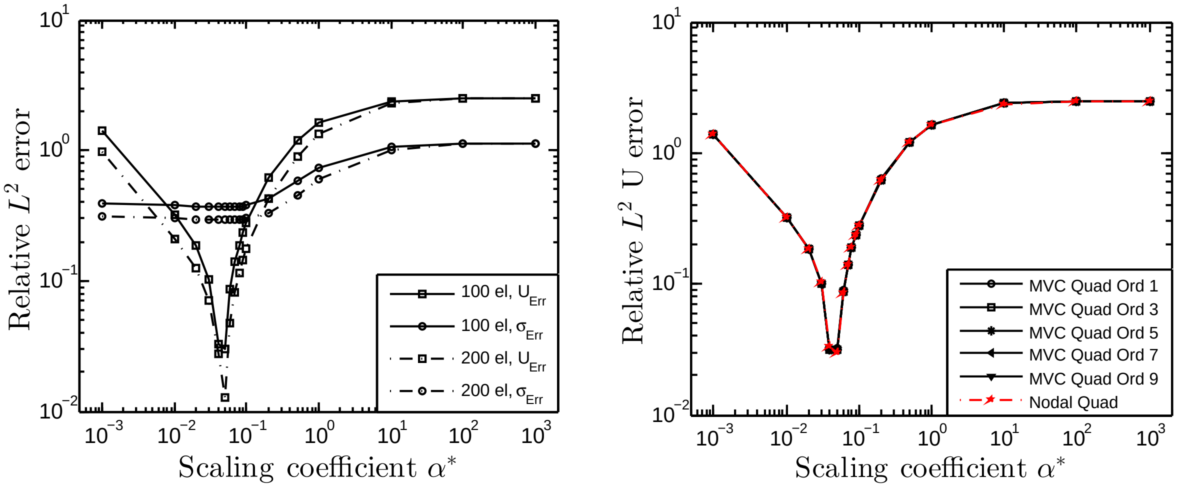
$$
\begin{aligned}
&\sigma_{xx} = 0, \qquad \sigma_{yy} = 0, \qquad \sigma_{zz} = \frac{3F}{4ab^3} yz \qquad \sigma_{xy} = 0 \\
&\sigma_{zx} = \frac{3F\nu a}{2\pi^2 b^3 (1+\nu)} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} sin\left(\frac{n\pi x}{a}\right) sinh\left(\frac{n\pi y}{a}\right) \Big/ cosh\left(\frac{n\pi b}{a}\right), \\
&\sigma_{yz} = \frac{3F\left(b^2 - y^2\right)}{8ab^3} + \frac{F\nu\left(3x^2 - a^2\right)}{8ab^3(1+\nu)} \\
&- \frac{3F\nu a}{2\pi^2 b^3 (1+\nu)} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} cos\left(\frac{n\pi x}{a}\right) cosh\left(\frac{n\pi y}{a}\right) \Big/ cosh\left(\frac{n\pi b}{a}\right)
\end{aligned}
\tag{4.40}
$$

Using the stress-strain relationships and definitions of displacement gradients, we derived the displacement fields corresponding to the stresses (4.40):

$$
\begin{aligned}
u &= -\frac{3F\nu}{4ab^3 E} xyz + C_1 + C_2 y - C_3 z, \\
v &= \frac{3F\nu}{8ab^3 E} z\left(x^2 - y^2\right) - \frac{Fz^3}{8ab^3 E} + C_4 - C_2 x + C_5 z, \\
w &= \frac{3F}{8ab^3 E} yz^2 + \frac{F\nu}{8ab^3 E} y\left(y^2 - 3x^2\right) + \int \frac{2(1+\nu)}{E} \sigma_{yz} dy + C_6 + C_3 x - C_5 y
\end{aligned}
\tag{4.41}
$$

Here, u, v and w represent the displacements in the x, y and z directions, respectively. Constants $C_1 - C_6$ account for the six rigid body motions in three-dimensions. For the simulations, the beam is assumed to be of dimension $1 \times 1 \times 5$ and the shear load is taken as $F = 0.1$.

As mentioned before, in the VEM formulation, the choice of scaling factor for the stability term in the stiffness matrix (4.31) is significant and affects the accuracy of results. We selected the scaling factor as $\alpha^*\ trace(\boldsymbol{K}_1)$. In order to evaluate this feasible choice of α^*, we conduct a parametric study where α^* is varied in the range $[10^{-3} - 10^{3}]$. The study use two mesh discretizations of 100 and 200 CVT Voronoi elements. In the plots (Fig. 4.12), relative L^2 displacement error (4.34) and relative L^2 average stress error (4.37) are plotted against scaling coefficient α^*. It can be seen in Fig. 4.12(a) that the relative L^2 errors attain a minimum somewhere in the range $[0.03, 0.08]$ for both the errors and both CVT mesh discretizations. Similar results are obtained for other mesh discretizations and are not shown here. We also investigated the effect of various orders of quadrature for the conforming interpolant quadrature scheme. The studies show (Fig. 4.12(b)) that there is no discernible improvement in accuracy with increase in the order of the quadrature scheme. In fact, the accuracy of higher order quadrature for the conforming interpolant quadrature scheme is comparable to that of the nodal quadrature scheme for almost the entire range of α^* studied. The reason the results are invariant to the order of integration is the same as the one provided for the displacement patch test. Based on this study, we chose $\alpha^* = 0.05$ and nodal quadrature for all remaining investigations.



(a)

(b)

Figure 4.12: Study of the scaling coefficient α^* for the shear loaded cantilever beam problem. (a) Error plots for two polyhedral mesh discretizations of 100, 200 elements using nodal quadrature for surface integration. (b) Comparison of nodal and conforming interpolant quadrature of different orders, for polyhedral mesh of 100 elements. Each curve represents the average errors of 5 sets of CVT meshes of same discretization.

The shape of the polyhedrons dictates the accuracy of the numerical simulations. To verify this hypothesis, we tested the performance of the VEM using CVT, RND Voronoi meshes and uniform hexahedral/brick meshes. The results of the study are shown in Fig. 4.13. The relative L^2 errors are plotted against the mean element diameters. For polyhedral meshes, each curve represents the average errors of 5 sets of CVT meshes of the same level of discretization. For the displacement field, using bricks and RND Voronoi meshes, a second order convergence is achieved which is consistent with the finite element literature, c.f. Fig. 4.13(a). For the choice of $\alpha^* = 0.05$, the VEM using CVT meshes showed a superior performance with an average rate of convergence slightly over two. For a given mean element diameter, CVT meshes produce the smallest error, followed by RND and brick meshes. The shape of elements has a strong influence on the numerical results. Babuska and Aziz [22] demonstrated that the accuracy of finite element analysis on triangular meshes degrades as the angles approach 180°. The results for polyhedral meshes show a similar pattern. Numerical analysis on random Voronoi meshes (RND), which contain elements of bad aspect ratio, are almost one order of magnitude less accurate than CVT meshes. Note that the VEM error depends on the scaling coefficient α^*. We chose an optimum α^* which produced the smallest error for CVT meshes (Fig. 4.12(a)). Other meshes such as hexahedrons or random Voronoi may have a different optimum value. The relative L^2 stress errors, on the other hand, show almost no noticeable difference (Fig. 4.13(b)). Brick meshes seem to be marginally more accurate compared to CVT and RND meshes. A first order rate of convergence is observed for the computed average stresses.

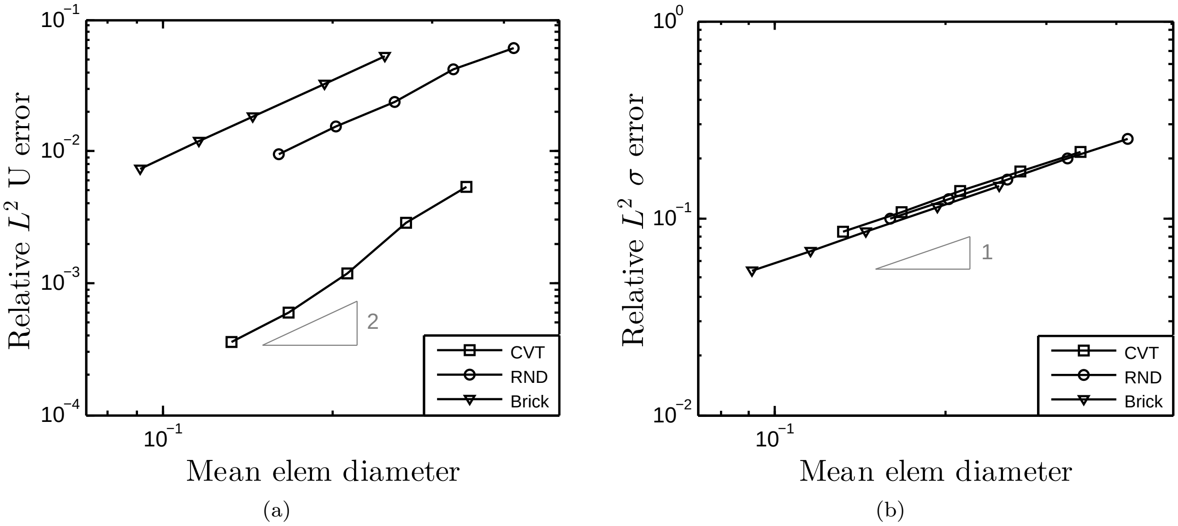
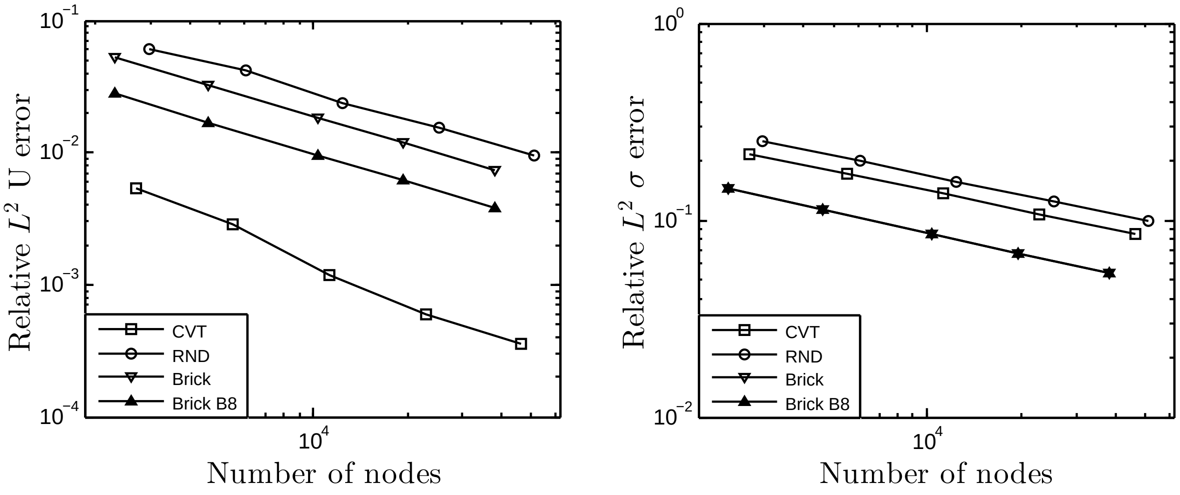


Figure 4.13: Performance of the VEM on different meshes, namely CVT Voronoi, random (RND) Voronoi and Brick meshes. (a) L^2 displacement error. (b) L^2 average stress error. Results pertaining to polyhedral meshes are average errors of 5 sets of meshes.

Evaluating the accuracy of the numerical method on different meshes, based on error estimates plotted against mean/max element diameters, may not be the most appropriate measure. The RND meshes, which might have highly skewed elements, may have a higher mean element diameter in a mesh, compared to CVT or brick meshes. Error estimates with respect to the total number of nodes in the mesh may provide better insight into the effectiveness of the method (Fig. 4.14). As a frame of reference, we included the relative L^2 errors computed on brick meshes using the B8 finite element (FE) shape functions. As seen before, for the displacement fields using $\alpha^* = 0.05$, VEM on CVT meshes continues to outperform VEM on RND and uniform brick meshes for the current measure as well. The FE results using B8 shape functions are less accurate compared to the VEM on CVT meshes (Fig. 4.14(a)). For the stress errors, the difference is bit more apparent. It is interesting to note that, the relative L^2 average stress errors are almost identical for analysis using the VEM and B8 shape functions on a uniform brick mesh.

In their work, Beirão Da Veiga et al. [28] use the concept of decomposition of the approximation space into two orthogonal spaces and solve the Stokes problem on polygonal meshes. Their consistency term construction is similar to the approach discussed in this chapter, but for the stability term they provide a different expression which still satisfies all



(a) (b)

Figure 4.14: Comparison of B8 shape functions with VEM on different meshes (CVT Voronoi, RND Voronoi and Brick mesh). (a) L^2 displacement error. (b) L^2 average stress error. Polyhedral mesh results are average errors of 5 sets of meshes.

the desired requirements discussed previously. Their stability term is:

$$\boldsymbol{K}_2 = \alpha \mathbf{P} \tag{4.42}$$

where $\mathbf{P}$ is the orthogonal projector of $\mathbb{P}_1(E)^3$ and is chosen as:

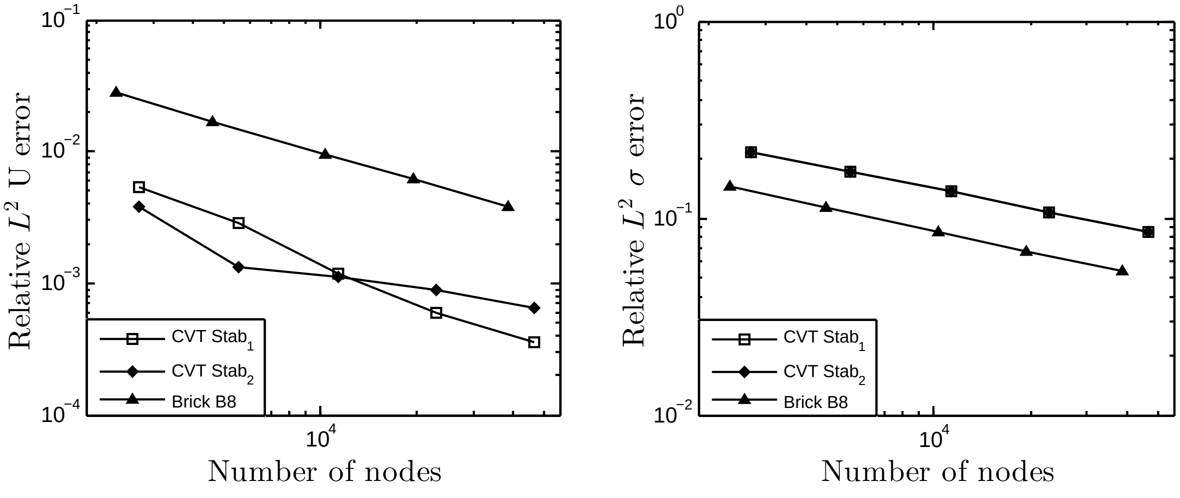
$$\mathbf{P} = \mathbf{I}_{3n\times 3n} - \mathbf{T}\left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T \tag{4.43}$$

The matrix $\mathbf{T}$ is the modified nodal coordinate matrix $\mathbf{N}$ of dimension $3n \times 12$,

$$T_{3i-2:3i,:} = \begin{bmatrix} 1 & 0 & 0 & y_i & 0 & -z_i & x_i & 0 & 0 & y_i & 0 & z_i \\ 0 & 1 & 0 & -x_i & z_i & 0 & 0 & y_i & 0 & x_i & z_i & 0 \\ 0 & 0 & 1 & 0 & -y_i & x_i & 0 & 0 & z_i & 0 & y_i & x_i \end{bmatrix} \tag{4.44}$$

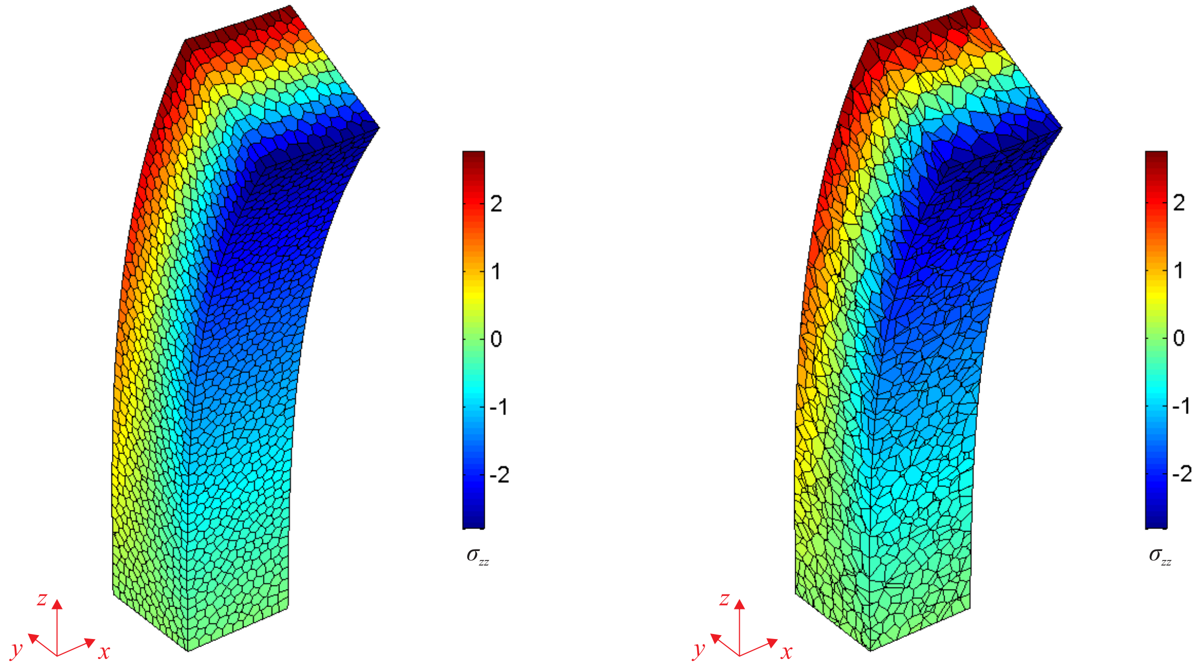
Here, we compare the procedure to compute the stability term in the stiffness matrix discussed in section 4.2.1 to the one discussed above (referred to as Stab_1 and Stab_2, respectively). Illustrated in Fig. 4.15, the relative L^2 average stress curves for Stab_1 and Stab_2 almost overlap each other, indicating similar accuracy and rate of converge between the two stability terms. However, Stab_1 seems to converge at a faster rate than Stab_2 as the meshes become more fine.

As a means to visualize the beam deformation under uniform shear load and the stress fields generated as a result of the deformation, we illustrate one of the results in Fig. 4.16 on CVT and RND meshes.



(a) (b)

Figure 4.15: Comparison of the two stability terms using CVT Voronoi meshes. (a) L^2 displacement error. (b) L^2 average stress error. Polyhedral mesh results are average errors of 5 sets of meshes.



(a) (b)

Figure 4.16: Deformation plots for shear loaded cantilever beam bending problem using (a) CVT Voronoi mesh and (b) Random Voronoi mesh. The colors indicate the magnitude of σ_{zz} stress. Shear load is applied on the bottom face in the negative y direction.

4.4 Concluding remarks

In this work, we present a Virtual Element Method approach for three-dimensional elasticity problems using linear polygonal elements in which degrees of freedom are associated with the nodes. The Virtual Element Method is a Galerkin method, but unlike typical finite element schemes does not compute the approximation space explicitly. The discrete bilinear form and load linear form are constructed directly without the computation of shape function derivatives in the interior of the elements. The construction begins with the definition of projection maps which extract the rigid body and constant strain components of an element deformation. Twelve basis functions are chosen, which span the linear polynomial space, to define and construct projection maps. Virtual Element Method reduces the volumetric integration, encountered in typical finite element methods, to compute the stiffness matrix to surface integrals, thereby reducing the associated computational cost. The previous statements are true for any generic polyhedron except linear tetrahedrons where one point volume integral is able to integrate the discrete bilinear form exactly and thus is cheaper than four surface integrations on the four faces. In VEM, the stiffness matrix is constructed to capture the linear deformation modes exactly, thereby passing the engineering patch test and ensuring optimum numerical convergence. It is shown in this chapter that the computations reduce to surface integrals consisting of only geometric quantities.

The method is tested using the displacement patch test and shear loaded beam bending problems. The formulation passes the displacement patch test irrespective of the choice of the type of surface integration scheme, stability term, or scaling factor for the stability term. Next, we evaluated the method using the shear loaded cantilever beam bending problem. A parametric study of the scaling coefficient, α^*, for the stability term of the stiffness matrix showed that the ideal choice of α^* is in the range $[0.03, 0.08]$ in order to obtain the lowest displacement and average stress errors for CVT meshes. Note that the optimum range for α^* may be different for other types of meshes such as hexahedrons or random Voronoi. A second and first order rate of convergence is observed for the displacements and average stresses for all meshes considered. The results indicate that, for the choice of $\alpha^* = 0.05$, polyhedrons perform better, when it comes to displacement fields in shear loaded bending problems, when compared to uniform bricks. Displacement field obtained using finite element method on hexahedral mesh were relatively more accurate than the VEM on the same mesh. In terms of average stress errors, similar trends were seen for different meshes. Random Voronoi meshes containing many degenerate elements perform well, with expected rates of convergence, using the VEM formulation, and its accuracy is close to the high quality CVT Voronoi meshes. Finally, we compared the two different expressions for the stability terms. Relative L^2 average stress errors for both were almost identical, whereas the stability term 1 showed a faster rate of convergence for relative L^2 displacement errors.

The Virtual Element Method (VEM) is in many ways similar to the Discrete Exterior Calculus (DEC). DEC [104, 96] is the discrete counterpart of Exterior Calculus, which is a generalization of vector calculus or is calculus on manifolds[2]. Essentially, in one of the DEC approaches, on a given discrete mesh, complete calculus is developed using only discrete combinatorial and geometric operations. DEC defines discrete equivalents of differential forms along with vector fields and operators acting on these objects. Using these discrete quantities, a calculus is developed by defining special operators such as discrete exterior derivatives (which uses topological information), codifferential operator (uses geometric information) and Hodge star operator (which encodes the metric[3]). In DEC, in order to solve the elasticity problem, the governing state equation is rewritten in the exterior calculus notation and all operators and objects are replaced by their discrete counterparts. In particular, this involves the discrete Hodge star which is defined using purely geometric quantities defined on dual and primal meshes (for example, Voronoi diagrams are dual of Delaunay triangulation). So, similar to the VEM, no canonical shape functions are involved. The similarities come from the fact that both VEM and DEC belong to a class of methods that are generally referred to as compatible discretization schemes which attempt to mimic the governing physics behind a physical problem.

---

[2] Manifold is a topological space which is locally Euclidean

[3] Metric refers to measure of geometric quantities, most commonly distances. For instance, in Euclidean space, the metric is the Euclidean distance/L^2 norm.

Chapter 5

Polyhedral topology optimization

Typical engineering applications require unstructured meshes to capture the design domain, boundary and load conditions accurately. Unstructured polyhedral meshes provide great flexibility in discretizing complicated domains. To solve the governing linear elasticity problem, the available polyhedral interpolants such as maximum entropy shape functions and Laplace shape functions, tend to be computationally prohibitive in order to obtain numerically accurate results. The reason being that, to solve the elasticity problem on polyhedral meshes, the computation of the global stiffness matrix would require conducting numerical integration in physical coordinates and dealing with each polyhedral element individually. In the current work, we utilize the Virtual Element Method approach, discussed in Chapter 4, to develop a scheme for three-dimensional linear elastic topology optimization. In this approach, the stiffness matrix is constructed in such a way that the displacement patch test is passed exactly and the bilinear form of the elasticity problem is continuous and bounded to ensure unique and stable solutions. The stiffness matrix terms require evaluation of surface integrals, in contrast to the volume integrals encountered in conventional FEM, thus reducing the overall computational cost.

Topology optimization strives to obtain an optimal distribution of material which satisfies certain design objectives. In the past, Talischi et al. [209, 211], Gain and Paulino [87] have explored two-dimensional topology optimization on arbitrary polygonal meshes. In two dimensions, using the iso-parametric mapping scheme for numerical integration [200], polygonal shape functions can be utilized as efficiently as the typical triangle and quad elements. However, in three-dimensions, the polytope shape functions discussed earlier loose their numerical efficiency. The solution of discrete elasticity problems requires the evaluation of derivatives of polyhedral shape functions, which are far from trivial. The numerical integration can only be performed in physical coordinates which increases the computational cost. Moreover, to achieve accurate results, a very high order quadrature rule needs to be used. The approach discussed in Chapter 4, known as the Virtual Element Method (VEM), addresses some of the challenges pertaining to the use of polyhedral elements for three-dimensional problems. This method provides a greater flexibility in the geometric shapes of the admissible elements. High quality, skewed, degenerate and even non-convex polyhedra are all admissible.

The remainder of this chapter is organized as follows. In Section 5.1, we discuss the topology optimization problem formulation along with the sensitivity calculations and the filter technique used in this work. We discuss the centroidal Voronoi tessellation (CVT) based approach to generate polyhedral meshes in Section 5.2. In Section 5.3, we show several numerical examples used to evaluate the current approach. Finally, we provide some concluding remarks in Section 5.4.

5.1 Density-based topology optimization problem formulation

In this work, we concentrate on a linearized elastic system discussed in Section 1.2. We deal with two categories of problems, compliance minimization (1.3) and linear compliant mechanism problems (1.4). The effective elasticity tensor $\mathbf{C}$ is a function of density, $\rho(\boldsymbol{x})$ and, as per the Solid Isotropic Material with Penalization (SIMP) model [36, 172], is expressed as:

$$\mathbf{C}(\rho) = \left[k_{min} + (1 - k_{min})\,\rho^p\right]\mathbf{C}^0 \tag{5.1}$$

The solid region is filled with material of elasticity tensor $\mathbf{C}^0$. Parameter k_{min} is arbitrarily chosen as 10^{-4} and the penalization parameter, p, is set to 3 for all numerical examples. In order to solve (1.5) numerically, it is discretized as follows:

$$\begin{aligned} &\inf_{\rho}\ J = \mathbf{P}^T\mathbf{U} \\ \text{subject to:}\quad &\mathbf{K}(\rho)\mathbf{U} = \mathbf{F}, \qquad \boldsymbol{V}^T\boldsymbol{\rho} \le V_f \end{aligned} \tag{5.2}$$

where $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{F}$ are the global stiffness matrix, global nodal displacement vector and global nodal force vector, respectively. Also, $\boldsymbol{V}$ is an array of the fractional volumes of elements, $\boldsymbol{V} = [V_e^1, V_e^2, ..., V_e^n]^T/|\Omega|$, where the V_e^i's are element volumes, and $\boldsymbol{\rho}$ is the element density array. The vector $\mathbf{P}$ represents the global force vector $\mathbf{F}$ for compliance minimization problems since the objective is to minimize external work. For the displacement inverter and gripper problems, $\mathbf{P}$ is a vector with all zeros except at locations corresponding to the output node, where it is unity. For more details on the displacement inverter problem refer to the book by Bendsøe and Sigmund [39]. The numerical method discussed in Chapter 4 is used to solve the above equilibrium equation.

5.1.1 Sensitivity analysis and optimizer

We use a gradient-based optimization algorithm for solving the discrete problem (5.2), which requires computation of the gradient of the objective function J. Using the adjoint method

[39], the sensitivity of J with respect to element density, ρ_e, is given by:

$$\frac{\partial J}{\partial \rho_e} = -p(1 - k_{min})\rho_e^{p-1}\boldsymbol{\lambda}_e^T \mathbf{K}_e \mathbf{U}_e \tag{5.3}$$

Here, $\mathbf{K}_e$ and $\mathbf{U}_e$ are the element stiffness matrix and element displacement vector, respectively. As per the adjoint method, for the compliance minimization problem (1.3), the vector $\boldsymbol{\lambda}_e$ is the same as the element displacement vector $\mathbf{U}_e$. For the displacement inverter problem (1.4), $\boldsymbol{\lambda}_e$ is the elemental component of $\boldsymbol{\lambda}$ that solves the adjoint system $\mathbf{K}\boldsymbol{\lambda} = \mathbf{P}$. Also, the sensitivity of the volume constraint with respect to element density, ρ_e, in (5.2) is expressed as $V_e/|\Omega|$, where V_e is the element volume. For the simulations, we use Optimality Criteria (OC) [39] as the optimization algorithm.

5.1.2 Projection method - Filtering

In order to obtain mesh independent designs, we enforce a minimum length scale using a linear filter which assigns a weighted average of the nearby elemental densities to each element. Thus, the projected element density of an element e, ρ_e, is written as:

$$\rho_e = \frac{\sum_{j \in N_{ej}} w_{ej}\rho_j}{\sum_{j \in N_{ej}} w_{ej}} \tag{5.4}$$

Here, N_{ej} is the set of all elements j, with element densities ρ_j, whose centers lie within a distance r_{min} from the center of the element e under consideration. The linear weights w_{ej} are:

$$w_{ej} = \max\left(0, \frac{r_{min} - r_{ej}}{r_{min}}\right) \tag{5.5}$$

where r_{min} and r_{ej} are the enforced minimum member size and the distance between centroids of elements e and j, respectively.

To compute the sensitivity of the objective function J with respect to the independent design variables ρ_j, we use the chain rule:

$$\frac{\partial J}{\partial \rho_j} = \frac{\partial J}{\partial \rho_e}\frac{\partial \rho_e}{\partial \rho_j} \tag{5.6}$$

which is computed using (5.3) and partial derivative of (5.4) with respect to ρ_j, given as:

$$\frac{\partial \rho_e}{\partial \rho_j} = \frac{w_{ej}}{\sum_{j \in N_{ej}} w_{ej}} \tag{5.7}$$

5.2 On centroidal Voronoi tessellation meshing

We use Voronoi diagram to generate three-dimensional polyhedral meshes [20, 243, 94, 210]. Given a set of n distinct seeds $\mathbf{Q} = \{\boldsymbol{q}_i\}_{i=1}^{n}$, the Voronoi tessellation of the domain $\Omega \subset \mathbb{R}^3$ is defined as:

$$\mathfrak{D}(\mathbf{Q}) = \{\Omega_i \cap \Omega : \boldsymbol{q}_i \in \mathbf{Q}\} \tag{5.8}$$

where Ω_i is the Voronoi cell corresponding to seed $\boldsymbol{q}_i$:

$$\Omega_i = \left\{\boldsymbol{x} \in \mathbb{R}^3 : |\boldsymbol{x} - \boldsymbol{q}_i| < |\boldsymbol{x} - \boldsymbol{q}_j|, \quad \forall j \neq i\right\} \tag{5.9}$$

The above definition of Ω_i represents a domain consisting of all points that are closer to seed $\boldsymbol{q}_i$ than any other seed $\boldsymbol{q}_j \in \mathbf{Q}$. Here, $|\cdot|$ is the standard Euclidean norm in $\mathbb{R}^3$. Note that the Voronoi cells are necessarily convex polyhedrons since they are formed by the finite intersection of convex half-planes.

Following the guidelines discussed in [243, 210], a polyhedral discretization is obtained from the Voronoi diagram of a given set of seeds and their reflections about the closest boundary of Ω. Our meshing algorithm is implemented for general domains using the concept of a signed distance function. A signed distance function $d_\Omega(\boldsymbol{x})$ is defined as:

$$d_\Omega(\boldsymbol{x}) = s(\boldsymbol{x}) \min(|\boldsymbol{x} - \boldsymbol{y}|), \qquad \forall \boldsymbol{y} \in \partial\Omega \tag{5.10}$$

where $s(\boldsymbol{x})$ is the sign function defined as:

$$s(\boldsymbol{x}) = \begin{cases} -1, & \boldsymbol{x} \in \Omega, \\ +1, & \boldsymbol{x} \in \mathbb{R}^3 \backslash \Omega. \end{cases} \tag{5.11}$$

Thus, $d_\Omega(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in \partial\Omega$ and $d_\Omega(\boldsymbol{x}) < 0$ if $\boldsymbol{x} \in \Omega \backslash \partial\Omega$. Using the signed distance function and its gradient, the reflection, $\boldsymbol{q}_i^r$, of the seed $\boldsymbol{q}_i$ can be calculated as:

$$\boldsymbol{q}_i^r = \boldsymbol{q}_i^r - 2 d_\Omega(\boldsymbol{q}_i) \nabla d_\Omega(\boldsymbol{q}_i) \tag{5.12}$$

First, to construct a polyhedron discretization of the domain Ω, each point in $\mathbf{Q}$ is reflected about the closest boundary of Ω. The resulting set of points are denoted by $R_\Omega(\mathbf{Q})$. Subsequently, we construct the Voronoi diagram of the space using the original point set and its reflection, $\mathcal{T}(\mathbf{Q} \cup R_\Omega(\mathbf{Q}); \mathbb{R}^3)$. For a given point set $\mathbf{Q}$, the discretization of the domain Ω is uniquely defined and denoted by:

$$\mathcal{M}_\Omega(\mathbf{Q}) = \left\{V_{\mathbf{y}} \in \mathcal{T}(\mathbf{Q} \cup R_\Omega(\mathbf{Q}); \mathbb{R}^3) : \mathbf{y} \in \mathbf{Q}\right\} \tag{5.13}$$

If the Voronoi cell of a seed $\boldsymbol{q}_i$ and its reflection have a common edge, then this edge forms an approximation to the domain boundary and a reasonable discretization of Ω is obtained. In order to mesh complicated geometries, a signed distance function along with set operations such as union, difference, and intersection are used [168, 210].

In the meshing algorithm, a set of signed distance functions corresponding to basic geometric shapes such as three-dimensional plane, sphere, cylinder and rectangular box is defined. We also construct a bounding box B, that contains the domain Ω, to generate the random seeds in $\mathbb{R}^3$. A random seed is accepted only if it lies inside the domain Ω, determined by evaluating the sign of the resultant distance function, d_Ω, associated with Ω. Figure 5.1 shows the basic steps for obtaining a random point set of size n.

```
input: $B, n$        %% $B \supset \Omega \in \mathbb{R}^3$ and $n$ is the number of seeds
  set $\mathbf{Q} = \emptyset$
  while $|\mathbf{Q}| < n$ do
      generate random point $\mathbf{y} \in B$
      if $d_D(\mathbf{y}) < 0$ then
          $\mathbf{Q} \leftarrow \mathbf{Q} \cup \{\mathbf{y}\}$
      end if
  end while
output: $\mathbf{Q}$
```

Figure 5.1: Initial random seed placement.

We handle *convex* and *non-convex* features of Ω by carefully choosing a set of seeds to be reflected w.r.t. the boundary. Reflection of a seed far from the boundary may land inside the domain or interfere with the reflection of another seed. Since the reflection of most of the seeds in the interior of the domain has no effect on the approximation of the boundary, we reflect only the seeds that are in a band near the boundary. A seed $\mathbf{y} \in \mathbf{Q}$ is reflected about boundary segment $\partial\Omega$ provide that:

$$|d_\Omega(\mathbf{y})| < c\left(\frac{|\Omega|}{n}\right)^{1/3} \tag{5.14}$$

where n is the number of seeds and c is the proportionality constant, chosen to be greater than 1 to make the band size near the boundary larger than the average element volume.

Clearly, Voronoi meshes generated from random/quasi-random seeds may cause inconsistencies at the boundaries resulting in a poor approximation of the boundaries of the design domain. To introduce some regularity in the Voronoi meshes, we construct centroidal Voronoi tessellations (CVT) using a modified Lloyd's algorithm [130]. For a large number of iterations, CVT cells tend to be uniform in size [77]. To generate CVT meshes, we replace seeds $\mathbf{Q}$ with centroids $\mathbf{Q}_c$ of the Voronoi cells. We compute the polyhedron centroid by partitioning it into tetrahedrons and determining the weighted mean of the centroids of the tetrahedrons. The weights are the volumes of the tetrahedrons.

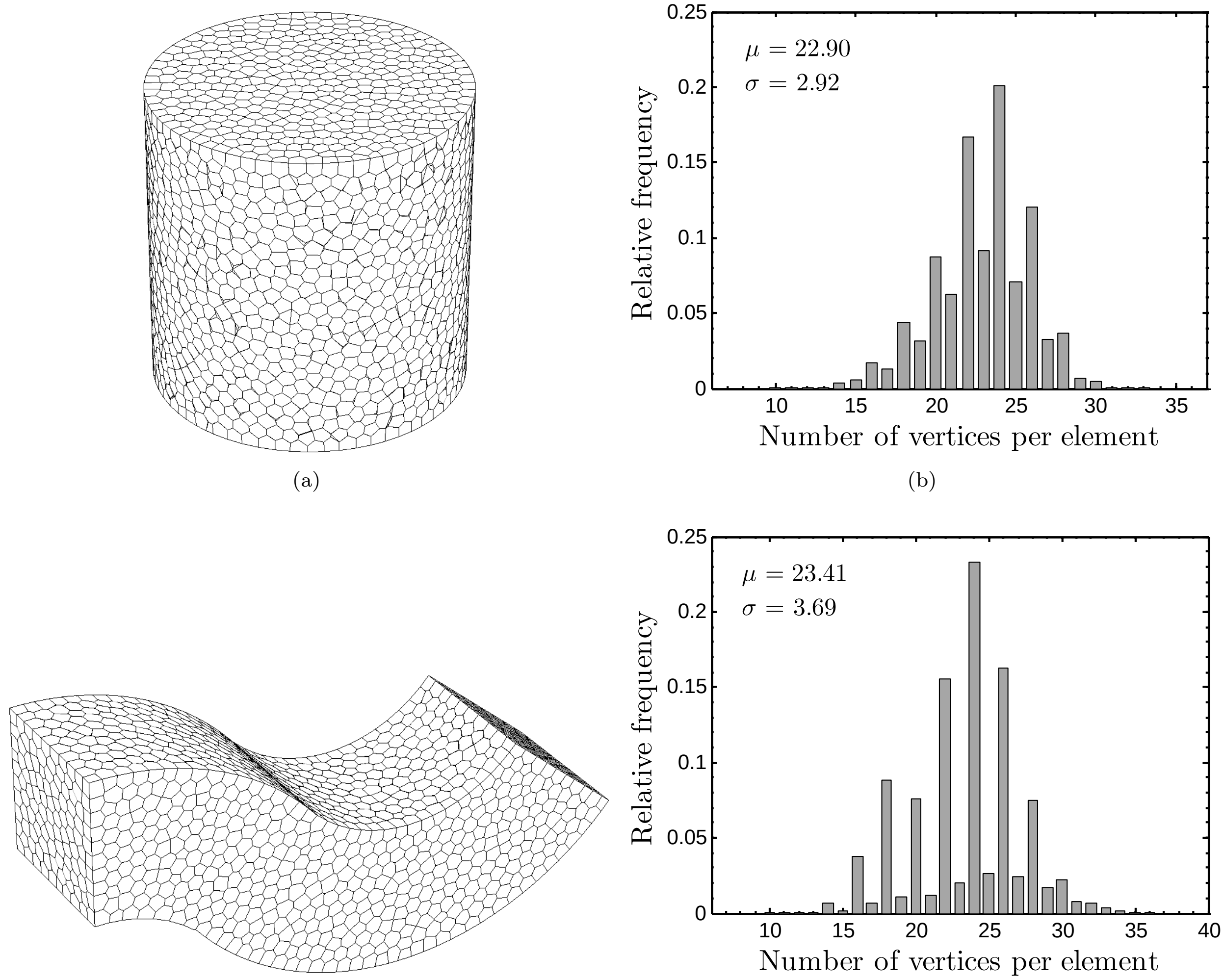
To construct the element stiffness matrix using the approach discussed in Section 4.2, along with vertices location and element connectivity information, we need to know the nodes on each face of each polyhedral element. The pseudo-code listed in Figure 5.2 summarizes the approach to obtain facial information in MATLAB. The inputs to the algorithm are the element connectivity and the coordinates of the vertices and seeds. For each seed of the mesh, the algorithm computes the convex hull of the set $\mathbf{S}$, which consists of the vertices of the corresponding element. The convex hull of $\mathbf{S}$ is a matrix $\mathbf{H}$ with as many rows as the number of triangles present in the convex hull, and three columns containing the indexes of the vertices of the triangles. Then, we iterate over all triangles and unite those that are co-planar to obtain the polygonal faces of the polyhedron. The resulting array, called **Elem**, contains the faces and vertices of each polyhedral element. Figure 5.3 shows some of the sample meshes and their statistics, obtained from the current algorithm.

```
input: $\mathbf{E}, \mathbf{N}_D, \mathbf{Q}, n$     %% Voronoi cells E, vertices coords $\mathrm{N}_D$, seeds coords Q, and number of seeds $n$
    Elem ← 0                 %% initialize an array with size of $n$
    for $i = 1$ to $n$ do
        let $\mathbf{S} = \mathbf{N}_D(\mathbf{E}(i))$
        construct convex hull $\mathbf{H} \leftarrow \mathcal{H}(\mathbf{S}; \mathbb{R}^3)$
        $m \leftarrow |\mathbf{H}|$          %% number of triangles $m$ obtained from the convex hull
        ElementFaces ← 0
        for $j = 1$ to $m$ do
            let $\mathbf{T} = \mathbf{E}(\mathbf{H}(j))$          %% vertices of a triangle
            $\mathbf{T} = OrderVertices(\mathbf{N}_D, \mathbf{T}, \mathbf{Q}(\mathbf{i}))$
            ElementFaces ← $\mathbf{T}$       %% create/unite faces
        end for
        $\mathbf{Elem}(i) \leftarrow$ ElementFaces
    end for
output: Elem
```

Figure 5.2: Construction of final mesh consisting of vertices, elements and faces in MATLAB.

5.3 Numerical examples

In this section, we show some numerical examples to demonstrate the effectiveness of the current approach. First, the results from the cantilever beam problem are solved on a box domain using different mesh discretizations, followed by problems on non-Cartesian design



(a) (b)

(c) (d)

Figure 5.3: Sample meshes. (a) Cylinder. (b) Cylinder mesh statistics. On average, the elements in the mesh have approximately 23 vertices with a standard deviation of 2.92. (c) Curved cantilever beam. (d) Curved cantilever mesh statistics. On average, the elements in the mesh also have approximately 23 vertices with a standard deviation of 3.69.

domains. For all the examples, the Young's modulus, E, and Poisson's ratio, ν, are taken as $10,000$ and 0.3, respectively. The scaling factor α is chosen as $\alpha^*\ trace(\boldsymbol{K}_1)$, where α^* is a scaling coefficient chosen as 0.05 based on a parametric study conducted in [91]. As mentioned earlier, Optimality Criteria (OC) is used as the optimization algorithm. The optimization is terminated when either the maximum of the change in element densities is less than 0.01 or the number of iterations exceed 300.

5.3.1 Cantilever beam problem on box domain

We begin with the benchmark cantilever beam problem for a design domain of dimension $2\times1\times1$. The left face of the box is fixed and a point load is applied in the middle of the right face (refer Fig. 5.4). The problem is solved on both brick and polyhedral element meshes. Taking advantage of symmetry, only half of the domain is optimized, which is discretized using 54,872 brick elements (60,060 nodes) and 10,000 polyhedrons (58,601 nodes). A linear filter of radius 5% of the maximum domain dimension is used and a volume fraction of 0.1 is prescribed.

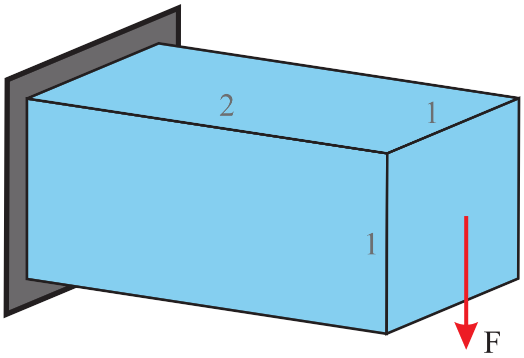
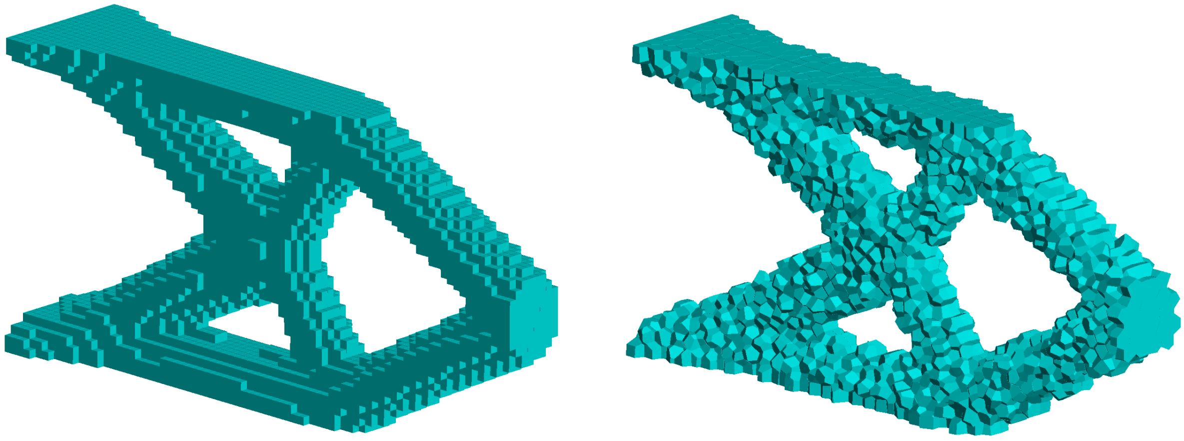


Figure 5.4: Cantilever beam problem.

With the present approach, both mesh discretizations produce similar optimization results (Fig. 5.5). Note that the optimized topologies shown in Fig. 5.5 (also all subsequent results) show only the elements whose density exceeds 0.5. The final compliance values are 0.1098 and 0.1082 for the topologies on brick and polyhedral meshes, respectively.

For comparison, the same problem is solved using the finite element method on a brick mesh. Topology similar to the present method is obtained. The convergence history for all three cases are illustrated in Fig. 5.6. As expected, a smooth monotonic convergence is obtained for all three cases and they all converge to similar final compliance values.


(a)

(b)

Figure 5.5: Converged topologies for the cantilever beam problem using the present method. (a) Brick mesh of 54,872 elements, 60,060 nodes ($J = 0.1098$). (b) Polyhedral mesh of 10,000 elements, 58,601 nodes. The average number of vertices per polyhedron is, $\mu = 22.85$, with standard deviation, $\sigma = 3.80$, ($J = 0.1082$).

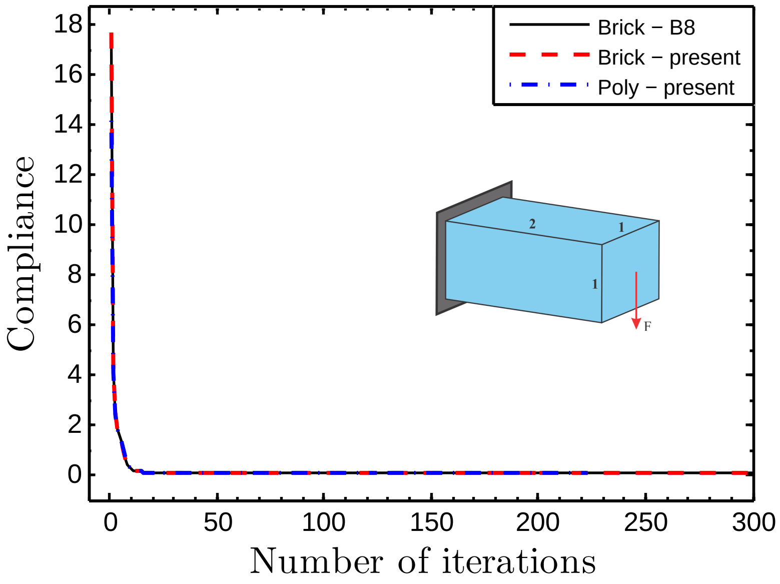


Figure 5.6: Convergence history for the cantilever beam problem.

5.3.2 Curved cantilever beam problem

Next, we explore problems on non-Cartesian design domains. The cantilever beam problem discussed in the previous section, with a point load applied in the middle of the right face, is now optimized on a curved design domain (c.f. Fig. 5.7). A volume fraction of 0.1 is prescribed, and a linear filter with radius equal to 4% of the maximum domain dimension is used.

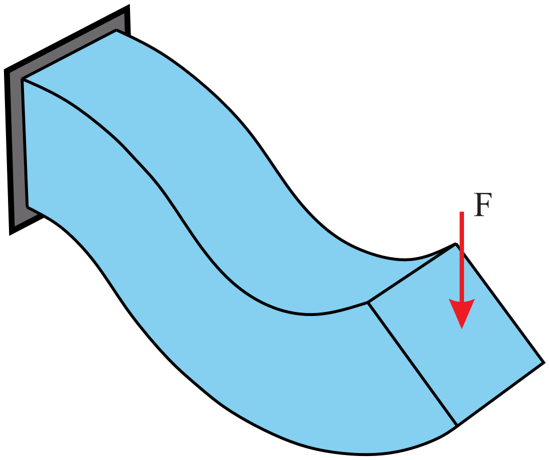


Figure 5.7: Curved cantilever beam problem. The left face is fixed and a point load is applied in the middle of the right face.

As illustrated in Fig. 5.8, the shape of the domain influences the final topology. The member orientations are similar to the ones in the two-dimensional version of the problem (c.f. [87, 211]). The optimization algorithm converged to a final compliance value of 0.0796.

5.3.3 Shear loaded thin disc

We next investigate the shear loaded thin disc problem. The thin disc domain has an external radius of 6 units with an internal cylindrical hole of radius 1 unit and has a thickness of 0.5 units (Fig. 5.9). Eight equidistant shear loads are applied along the circumference of the disc and all the nodes along the cylindrical hole are fixed. A polyhedral mesh of 10,000 elements (55,810 nodes) is used to discretize the design domain. A filter radius of 4% of the outer diameter is selected and a volume fraction of 0.2 is enforced.

According to Michell layout theory [143, 103], an optimum structural layout is one in which the tension and compression members meet orthogonally. Such a set of orthogonal curves are known as Hencky nets [103]. The tension and compression members in topology optimization solutions should adhere to this principle. The converged topology for the shear loaded thin disc problem, obtained from our algorithm, is shown in Fig. 5.10 (resembles a flower). The members of the structure intersect nearly at right angles, even for a coarse polyhedral mesh, indicating that mesh bias is alleviated with polyhedral elements. The compliance of the final topology is 0.5850.

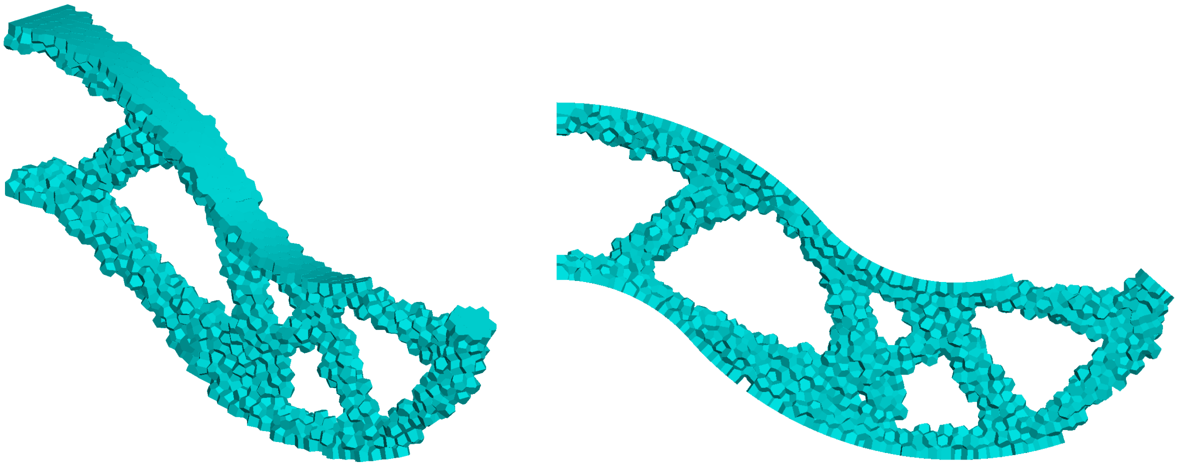


Figure 5.8: Converged topology for the curved cantilever beam problem using the present method on a 10,000 elements, 63,626 nodes polyhedral mesh. The average number of vertices per polyhedron is, $\mu = 23.38$, with standard deviation, $\sigma = 3.38$, ($J = 0.0796$).

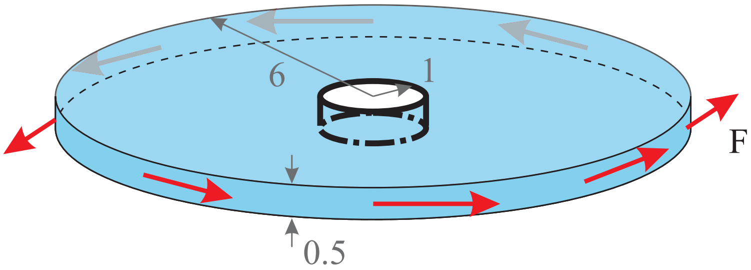


Figure 5.9: Problem description for thin disc. Eight equidistant shear loads are applied along the circumference of the disc and all the nodes along the central cylindrical hole are fixed.

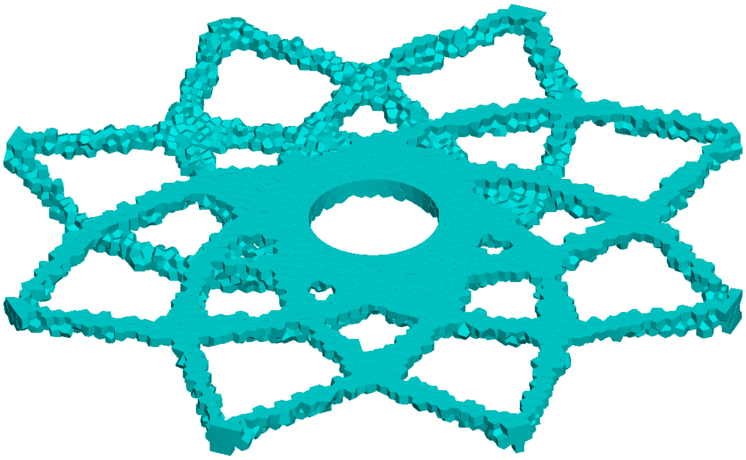
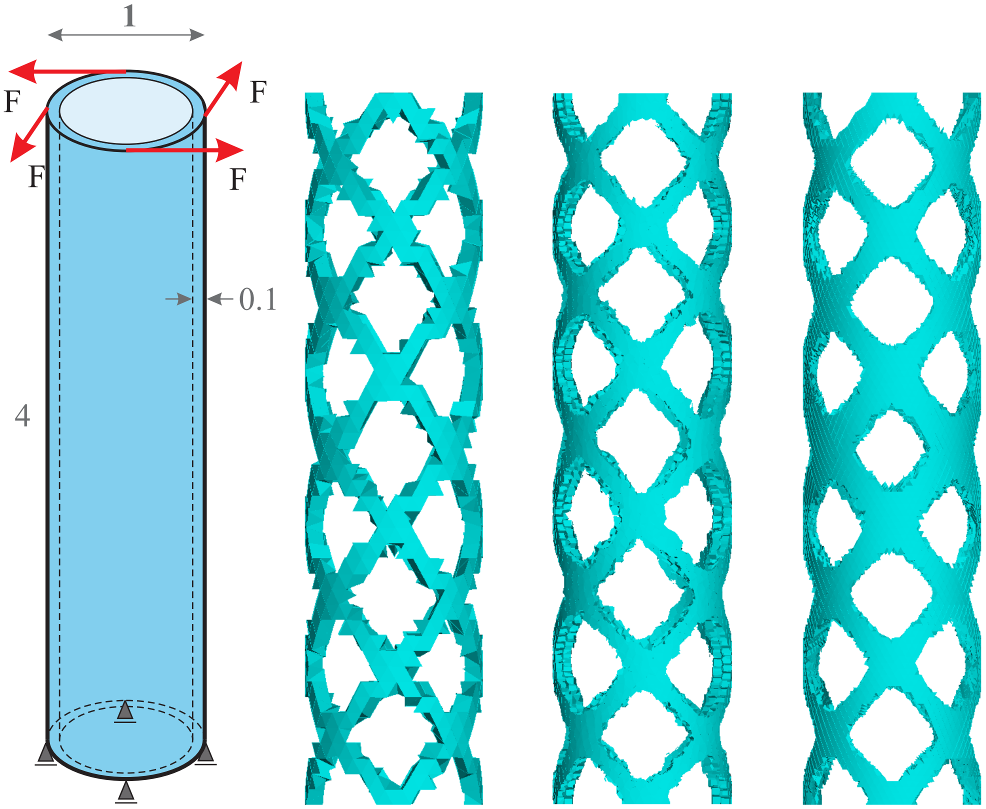


Figure 5.10: Final topology for shear loaded thin disc on a 10,000 element, 55,810 nodes polyhedral mesh. The average number of vertices per element are, $\mu = 20.53$, with standard deviation of, $\sigma = 3.71$, ($J = 0.5850$).

5.3.4 Hollow cylinder under torsional load

We now explore the torsionally loaded hollow cylinder. The design domain is in the shape of a hollow cylinder of thickness 0.1 units, height 4 units and outer diameter of 1 unit (c.f. Fig. 5.11(a)). Four equidistant nodes along the bottom face are fixed and four tangential point loads are applied to corresponding nodes on the top face, effectively acting as a torsional load. A filter radius of 3% of the height of the cylinder and a volume fraction of 0.3 are prescribed. The problem is solved on three sets of meshes - two tetrahedral meshes of 9,977 elements (3,349 nodes); 451,584 elements (85,320 nodes) and a polyhedral mesh of 10,000 elements (79,925 nodes).



(a) (b) (c) (d)

Figure 5.11: Hollow cylinder under torsional load. (a) Problem description. Converged topologies for (b) Linear tetrahedral mesh of 9,977 elements, 3,349 nodes ($J = 1.1397$); (c) Polyhedral mesh of 10,000 elements, 79,925 nodes. On average, polyhedral elements have, $\mu = 22.57$, vertices with standard derivation of, $\sigma = 2.88$, ($J = 1.6005$); (d) Linear tetrahedral mesh of 451,584 elements, 85,320 nodes ($J = 1.2064$).

Using the polyhedral mesh, our optimization algorithm yields an elegant spiraling structure with nearly orthogonally oriented crossing members (Fig. 5.11(c)). For a tetrahedral mesh, with a similar number of elements as the polyhedral mesh (9,977), we still obtain a similar spiraling structure (c.f. Fig. 5.11(b)), but the members orientation is affected by the mesh geometry and the intersecting members are not fully orthogonal. This might be because although the tetrahedral mesh has similar number of elements as polyhedral mesh, the degrees of freedom in tetrahedral mesh is far less than the polyhedral mesh. So we solved the hollow cylinder problem on a fine tetrahedral mesh, such that the number of nodes in the mesh are comparable to that of the polyhedral mesh. Note that, in the fine tetrahedral mesh, the number of elements has risen to 451,584. Although the fine tetrahedral mesh (Fig. 5.11(d)) rectifies the lack of member orthogonality in the optimization result, the fine mesh considerably increases the computational cost associated with operations, such as creation and storage of the filter matrix, compared to the polyhedral and coarse tetrahedral meshes. If other length scale control approaches are adopted, for example enforcing perimeter constraint, in place of filters, the the computational cost associated with fine tetrahedral meshes can be considerably reduced. But the downside to this approach on tetrahedral meshes is that single node connections may arise in the designs. In terms of the cost associated with solving the governing elasticity problem, polyhedral meshes are marginally expensive than tetrahedral meshes of comparable total degrees of freedom. This is due to that fact that polyhedrons on an average have higher number of vertices (approximately 23 for our meshes) than tetrahedrons. Finally, note that the lower compliance for the optimization result on a coarse tetrahedral mesh (Fig. 5.11(b)) can be attributed to the fact that tetrahedral meshes experience artificial stiffness due to shear locking phenomenon, which reduces with mesh refinement.

We would like to point out that the element stiffness matrix obtained using the linear tetrahedral finite element approach is identical to the one obtained using the current approach. The reason being that, in the present method, for elements in the shape of a tetrahedron, the contribution of the stability term is zero, because the approximate space, $\boldsymbol{\mathcal{V}}_h^E$, is identical to the space of linear deformations, $\mathbb{P}_1(E)^3$. So, the only contribution to the element stiffness matrix comes from the consistency term which is the same as the one obtained from finite element analysis. Thus, the final topologies obtained from the current approach and the linear tetrahedral finite element approach should be identical, along with the convergence history. the results are in agreement with the above statement.

5.3.5 Hook domain under line load

For the final compliance minimization problem, we investigate the hook domain subjected to a uniformly distributed line load along the negative z-direction (Fig. 5.12). A volume fraction of 0.1 is prescribed and a linear filter with radius equal to 2% of the maximum domain dimension is used. Using symmetry, we optimize only half the hook domain. The polyhedral mesh contains 10,000 elements (67,893 nodes). The converged topology, Fig. 5.13, has a compliance of 7.0484 and resembles the structure of a fan. The two dimensional version of the problem [211] has similar member orientations as the current three-dimensional result.

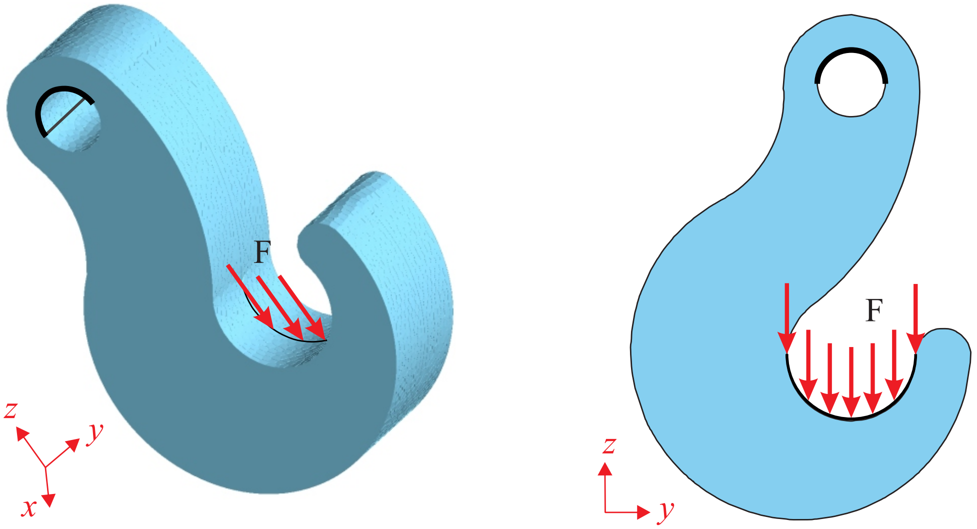


Figure 5.12: Hook domain problem description. All the nodes along the top half of the upper cylindrical hole are fixed and line load is applied along the circular arc in the negative z-direction.

In topology optimization, filters are used to enforce a length-scale in the problem and to ensure mesh-independency. Filtering implies that stiffness at any point in the design domain depends on the densities of all the points in the neighborhood of that point. Filtering imposes a limitation on the allowable density distribution in any optimized solution. We studied the effect of filters on the optimization results for polyhedral meshes. Keeping all the parameters the same as before, we solved the hook problem without using any filter. As expected, without any length-scale control, more structural members, including some thin ones, appear in the solution (Fig. 5.14). It is interesting to note that even without filtering no single node connections were observed in the design. Other filters, such as sensitivity filters may also be used.

5.3.6 Displacement inverter

Apart from compliance minimization, we also investigate a compliant mechanism problem, specifically the displacement inverter. The problem description is shown in Fig. 5.15. The domain is of dimension 1 × 1 × 1 and is fixed at the bottom four corners. The objective of optimization is to maximize the output displacement u_{out} on a workpiece modeled by a spring of stiffness k_{out}. The input and output spring stiffnesses, k_{in} and k_{out}, are taken to be the same as the components of the global stiffness matrix at the degrees of freedom corresponding to the input and output nodes. Taking advantage of symmetry, only a quarter of the domain is optimized and is discretized using 10,000 polyhedral elements (58,785 nodes). A volume

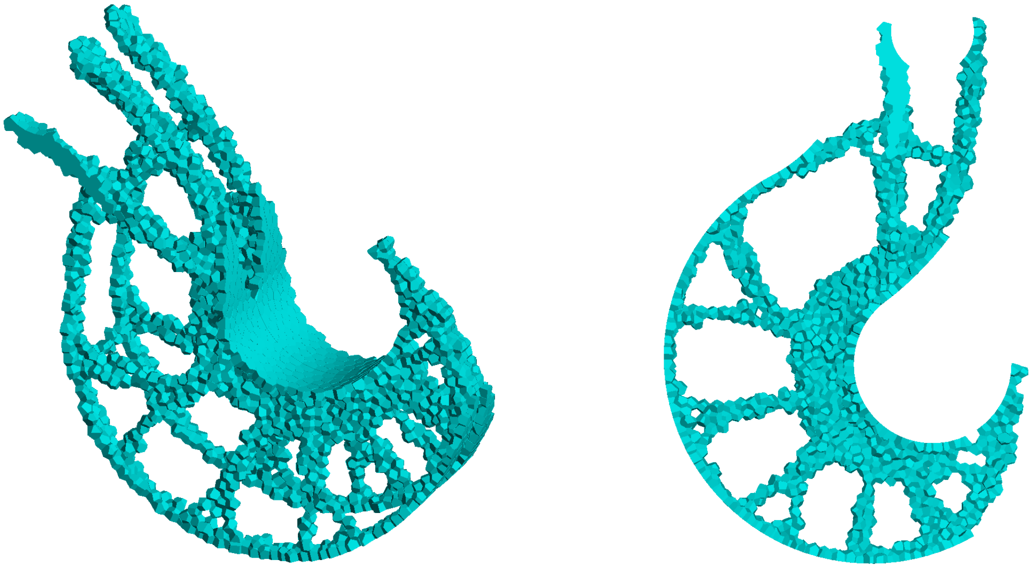


Figure 5.13: Final topology for the hook domain under line load using the present method on a 10,000 elements, 67,893 nodes polyhedral mesh. The average number of vertices per polyhedron is, $\mu = 23.97$, with standard deviation, $\sigma = 4.19$ ($J = 7.0484$). A linear filter with radius of 2% of the maximum domain dimension is used.

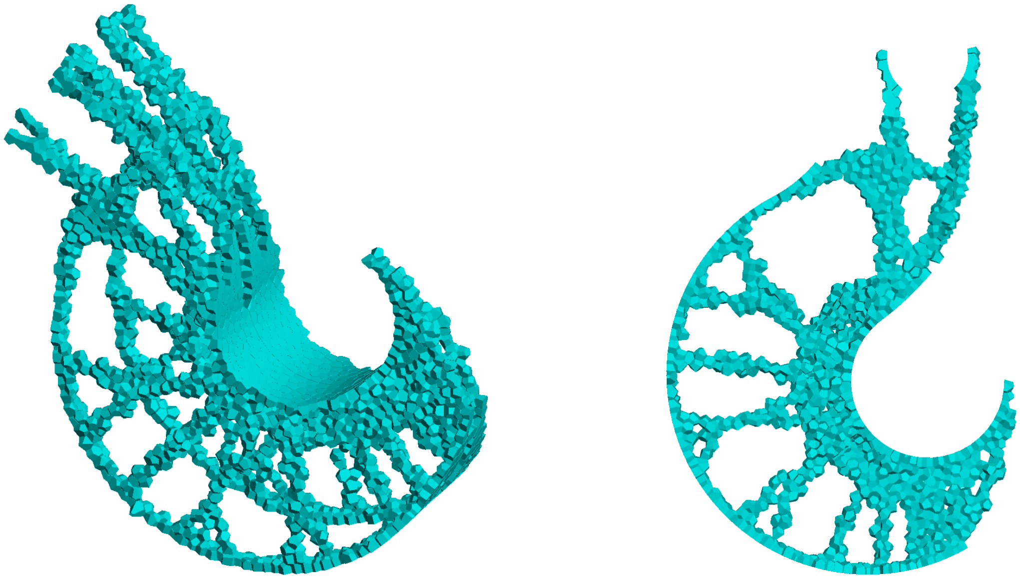


Figure 5.14: Hook problem without any filter.

fraction of 0.05 is prescribed. Our final topology (Fig. 5.16) is similar to the ones available in the literature [14]. Visually, the three-dimensional optimization result (Fig. 5.16) resembles to the two-dimensional optimization result [39].

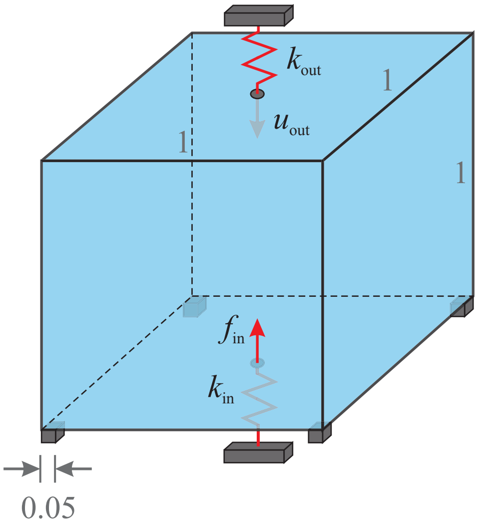


Figure 5.15: Displacement inverter problem.

The output spring stiffness, k_{out}, controls the displacement amplification obtained from the inverter. A high value of the spring stiffness produces small output displacement and vice versa. To verify this for our polyhedral topology optimization implementation, we ran the inverter problem using k_{out} of magnitude 0.01 times the global stiffness matrix component at the degree of freedom corresponding to the output direction. The current design (Fig. 5.17) is clearly different from the one obtained before (Fig. 5.16) towards the top. The four members at the top become narrower as they approach the knot (compared to Fig. 5.16), making the design more compliant.

5.3.7 Gripper

A gripper, as the name suggests, is a complaint mechanism suitable for gripping objects. The input actuator, modeled as a spring with stiffness k_{in}, and a force f_{in}, applies a horizontal load as indicated in Fig. 5.18(a) and the goal of optimization is to maximize the vertical output displacement u_{out} on a workpiece modeled by a spring of stiffness k_{out}. The problem dimensions and boundary conditions are indicated in Fig. 5.18(a). Nodes on the top and bottom section on the right face, indicated by gray color, are fixed. Using the passive element concept the orange box is modeled as void. Before each optimization cycle, the elements lying inside the orange box are identified and are assigned the minimum density corresponding to voids. The spring stiffnesses, k_{in} and k_{out} are taken to be the same as the components of the global stiffness matrix at the degrees of freedom corresponding to

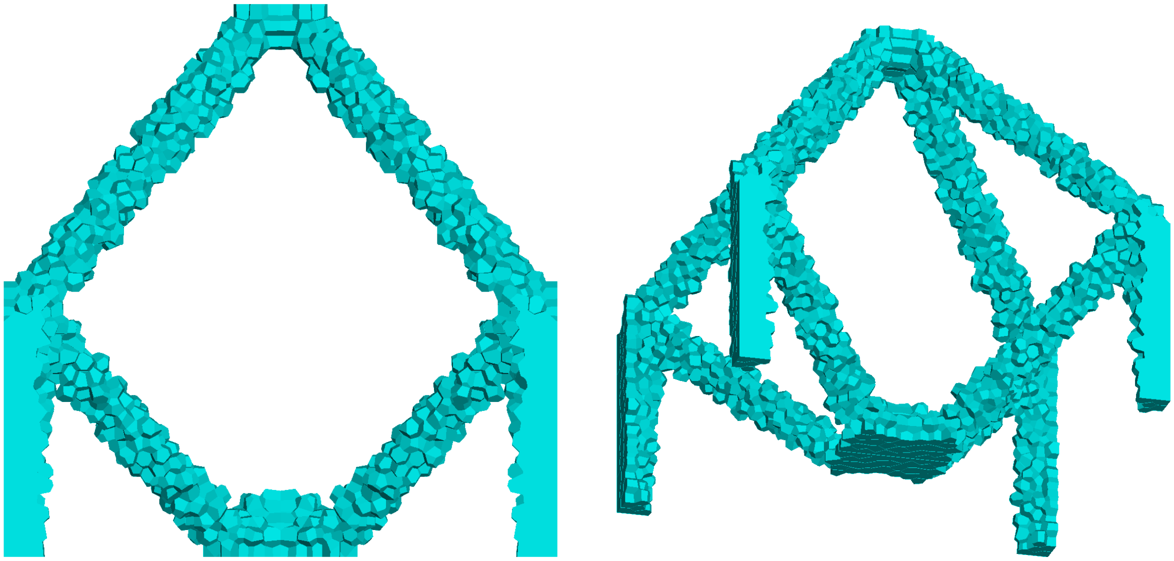


Figure 5.16: Converged topology for the displacement inverter problem. Due to symmetry, only a quarter of the domain is optimized and is discretized using 10,000 polyhedral elements containing 58,785 nodes. The average number of vertices per polyhedron is, $\mu = 22.98$, with standard deviation of, $\sigma = 3.75$.

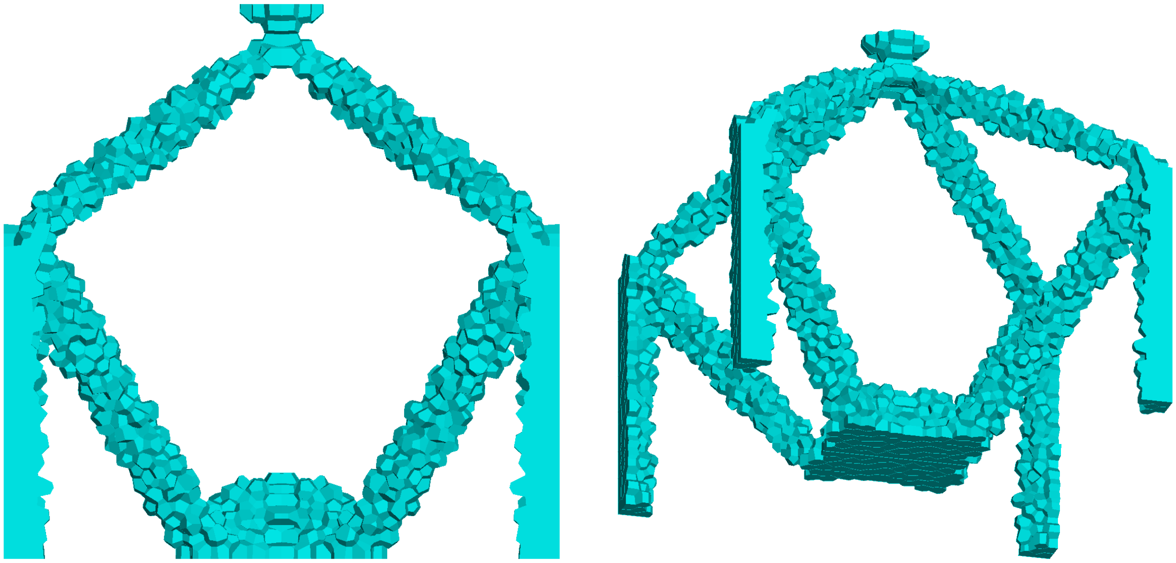


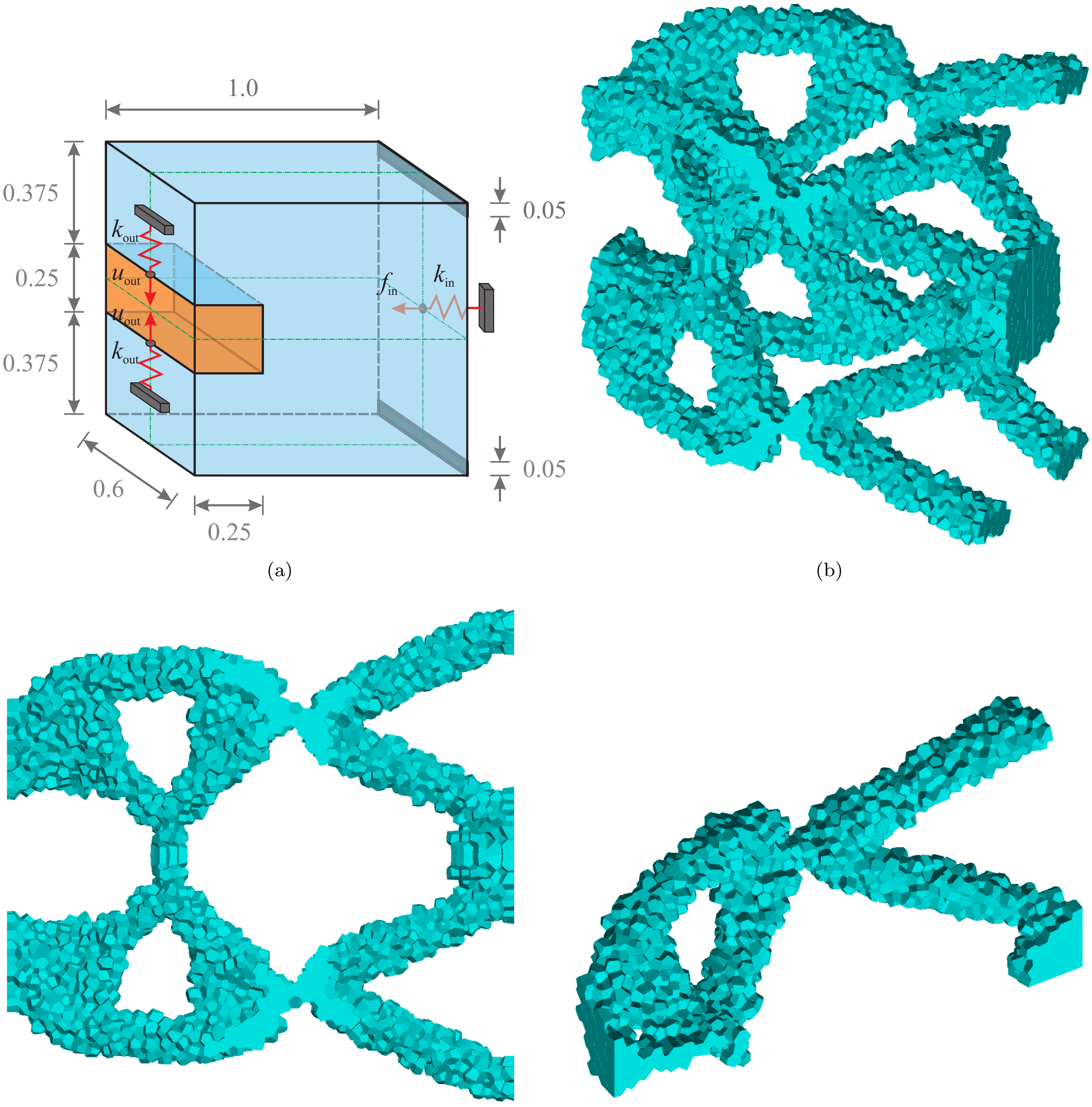
Figure 5.17: Converged topology for the displacement inverter problem using the present method on a 10,000 elements, 58,785 nodes polyhedral mesh.

the input and output nodes. Due to symmetry, only a quarter of the domain is optimized and is discretized using 10,000 polyhedral elements (58,785 nodes) with a prescribed volume fraction of 0.10. The converged topology (Fig. 5.18) for three-dimensional optimization is a clear extension of the two-dimensional results available in reference [39]. The combination of gripping jaws and hinge mimicking narrow necks, along the middle of the design, resemble a pair of scissors. The convergence history for the gripper problem, shown in Fig. 5.19, indicates stable convergence.

5.4 Concluding remarks

In this work, we explore polyhedral elements for three-dimensional topology optimization. Polyhedral elements provide flexibility in discretizing complicated design domains often encountered in engineering applications. We use a Virtual Element Method (VEM) to solve the elastic problem which is the governing state equation in our simulations. The VEM is a Galerkin scheme, similar to classical finite element approaches, which begins with transforming a continuous problem into its weak form and then solving the resulting equations on approximation spaces spanned by a finite set of interpolation functions. However, VEM does not require explicit computation of these interpolation functions. In VEM, the approximation space is first systematically decomposed into rigid body, linear and higher order spaces. Using the bases which span the rigid body and constant strain spaces, two projection maps are defined which extract the rigid body and constant strain modes of the deformation. These projection maps facilitate the construction of stiffness matrix without the need for the evaluation of canonical basis functions and their derivatives inside the element. All the computations are reduced to the evaluation of purely geometric quantities pertaining to the faces of the polyhedrons.

We investigate the topology optimization of compliance minimization and compliant mechanism problems using polyhedrons. A centroidal Voronoi tessellation (CVT) based meshing approach is implemented to discretize complicated design domains employing the concept of signed distance functions. Compliance minimization problems such as the cantilever beam, curved cantilever beam, shear loaded disc, hollow cylinder under torsion, hook problem subjected to line load; and compliant mechanism problems such as the displacement inverter and gripper are explored. As indicated by our thin disc and hollow cylinder problem solutions, unstructured polyhedral elements capture member orthogonality and alleviate mesh bias in the design. Single node connections often arise in topology optimization designs and techniques such as filtering are used to tackle them. Due to their geometry, polyhedral elements naturally alleviate such numerical anomalies. Current work of three-dimensional topology optimization with polyhedrons paves the way to future applications in the field of



(a) (b)

(c) (d)

Figure 5.18: Topology optimization design of gripper. A quarter of the problem is solved with a polyhedral mesh of 10,000 elements and 59,194 nodes. The average number of vertices per polyhedron is, $\mu = 23.11$, with standard deviation, $\sigma = 3.85$. Region in orange is assigned as voids using passive element concept. (a) Problem description. (b) Complete design. (c) Front view. (d) Quarter section view.

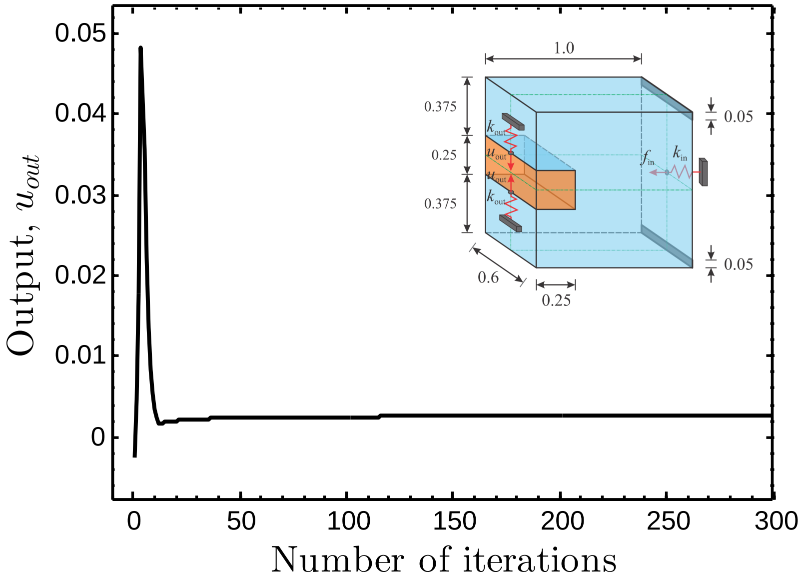


Figure 5.19: Convergence history for the gripper problem.

multiphysics designs [183, 56] and biomedical engineering [203]. The complicated designs resulting from polyhedrons-based three-dimensional topology optimization can be rapidly prototyped using additive manufacturing/3D printing.

As observed in typical topology optimization solutions, the ones presented in this chapter are governed by the chosen initial optimization parameters, such as the prescribed volume fraction and filter radius. Design loads and boundary conditions, which have been assumed deterministic in our analysis, often have uncertainties embedded in them. Reliability-based topology optimization approaches, to obtain robust designs, accounting for uncertainties, have been explored in the past [141, 118, 150]. There are also uncertainties associated with the manufacturing of these topology optimization designs using processes such as milling and etching. Misaligned, malcalibrated machine tools and under/over etching may degrade the performance of carefully optimized topology optimization designs. Robust design against such manufacturing imprecisions have been proposed [184, 170]. Development of a robust topology optimization scheme for polyhedrons will be explored in the future.

Chapter 6

BEAM - Bridging Engineering, Art and Mathematics

Generally, two-dimensional finite element analysis utilizes triangular and quadrilateral elements. The pioneering work of Wachspress laid the foundation for polygonal interpolants which has made polygonal elements popular. The next stage of element shapes, after polygonal elements, can be considered as tessellations. In this work, we investigate the topology optimization of linear elastic systems on tessellations as a means to coalesce art and engineering. We mainly focus on M.C. Escher's tessellations using recognizable figures. To solve the elasticity problem, we utilize a Mimetic Finite Difference-inspired approach, known as the Virtual Element Method. In this approach, the stiffness matrix is constructed to pass the displacement patch test and ensure optimum numerical convergence rates. Numerical verification studies such as the displacement patch test and shear loaded cantilever beam bending problem are solved to demonstrate the accuracy of the present approach in two-dimensions. The artistic aspects of topology optimization designs are explored using three numerical examples with the design objective of compliance minimization [88].

6.1 Introduction

Having explored the utility of polygonal elements in diverse fields such as fracture mechanics, topology optimization, micromechanical analysis, computer graphics and image processing, researchers have often pondered where do we go next? What is the next evolutionary stage of element geometries? One potential direction is *tessellations.* A tessellation refers to the arrangement of one or more geometric shapes to completely cover the plane, without overlapping and gaps. Tessellations have the potential to bridge diverse fields such as engineering, art and mathematics, which is the topic of investigation of the current work.

Tessellations divide a plane using closed regular as well as irregular shapes. There have been many pioneers in this field. Dutch graphic artist Maurits Cornelis Escher was a prolific creator of visual riddles, impossible structures, tessellations and patterns. He is well known for his, often mathematically-inspired, lithographs and woodcuts. Some of his famous works include impossible constructions and art works such as *Relativity*, *Waterfall* and *Hand with Reflecting Sphere*. Electrical engineer Robert Fathauer is another well known tessellation artist. He is one of the first artists to use computers to generate tessellations. He is famous for his artwork involving fractals, knots and links. Mathematics professor Roger Penrose is an expert in recreational math and has made significant contributions in the field of geometric puzzles and tessellations. Penrose tiles are one of his pioneering works. Penrose tiles are self-similar quasicrystals which possess reflective and rotational symmetry but lack translational symmetry (Fig. 6.1).

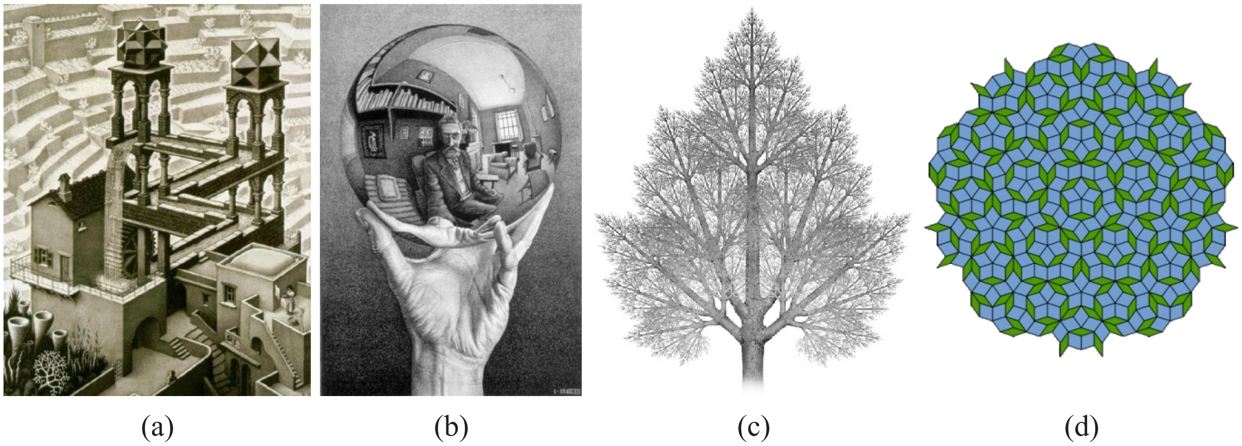

(a) (b) (c) (d)

Figure 6.1: Artworks of famous artists. M.C. Escher's (a) Waterfall, (b) Hand with Reflecting Sphere (www.wikipedia.org). (c) Robert Fathauer's fractal tree (http://mathartfun.com). (d) Roger Penrose's penrose tiling (www.wikipedia. org).

In order to numerically solve governing partial differential equations on arbitrary polygonal meshes, specialized approaches are available. Utilizing polygonal shape functions, one such approach, has been discussed in previous chapters. In polygonal shape functions-based numerical simulations, an isoparametric mapping scheme for numerical integration is used [200]. To pass the patch test using previously discussed shape functions, a high quadrature rule is needed which is computationally expensive. For tessellations, with a large number of vertices, numerical simulations can become even more cumbersome. A recently proposed approach, known as the Virtual Element Method (VEM), addresses the issue of both accuracy as well as efficiency and has been discussed in detail in Chapter 4. The Virtual Element Method provides an elegant way to handle any non self-intersecting closed polygon. In this work, to solve the linear elasticity problem, we use an approach derived from the Virtual Element Method.

Tessellations have been explored in computational mechanics in the past in the form of tiling. The terms *tiling* and *tessellations* are often used interchangeably. Tiling refers to patterns of polygons with straight boundaries. Tilings, such as the *pinwheel*, have been used in fracture mechanics [155, 163] because pinwheel tiling produces tiles in infinitely many orientations which is favorable for crack propagation. In this work, we investigate topology optimization on tessellated meshes as a means to coalesce engineering and art. Topology optimization aims to obtain an optimal material distribution in a domain in order to satisfy certain prescribed design objectives. Since the early works of Bendsøe and Kikuchi [37], Suzuki and Kikuchi [204], Bendsøe [36] and Rozvany et al. [172], the field of topology optimization has grown rapidly. Here, we mainly concentrate on the density-based linear elastic topology optimization problem of compliance minimization (1.3). The domain description for the optimization problem is shown in Fig. 6.2. Refer to Section 5.1 for more details on the problem formulation, sensitivity analysis and filtering scheme.

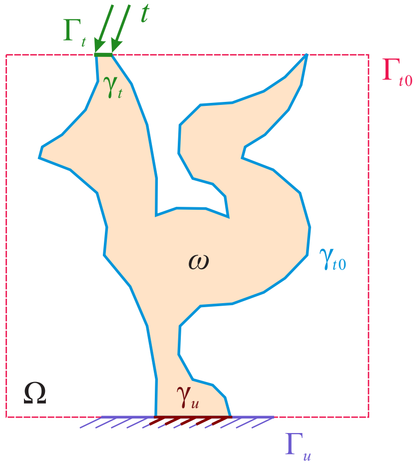


Figure 6.2: Illustration of the design domain, Ω. Its boundary, $\partial\Omega$, contains three disjoint segments, $\partial\Omega = \Gamma_u \cup \Gamma_{t0} \cup \Gamma_t$, corresponding to displacement, homogeneous traction, and non-homogeneous traction boundary conditions, respectively. Any design $\omega \subseteq \Omega$, with boundary $\partial\omega = \gamma_u \cup \gamma_{t0} \cup \gamma_t$, is constrained to satisfy $\gamma_u \subseteq \Gamma_u$ and $\gamma_t = \Gamma_t$. Boundaries γ_u, γ_{t0}, and γ_t correspond to displacement, homogeneous traction, and non-homogeneous traction boundary conditions on $\partial\omega$, respectively.

The remainder of this chapter is organized as follows. In Section 6.2, we discuss the generation of tessellations based on basic two-dimensional shapes such as triangles and quadrilaterals. In Section 6.3, we briefly discuss the Mimetic Finite Difference-inspired approach to solve two-dimensional linear elasticity equation on meshes with arbitrary shaped elements. In Section 6.4, we show some numerical verification studies to illustrate the accuracy of the current numerical approach. Section 6.5, shows some designs obtained from topology optimization on tessellations. Finally, we conclude with some remarks in Section 6.6.

6.2 Tessellation generation

In this section, we discuss the generation of tessellations which we use to explore art in engineering. All the tessellations discussed here use basic two-dimensional shapes, such as triangles and quadrilaterals, as the background shape which is then modified to obtain the motifs[1] used to generate the tessellations. These tessellations are developed using the basic concepts of translation and rotation. We divide the tessellations discussed in this work into

---

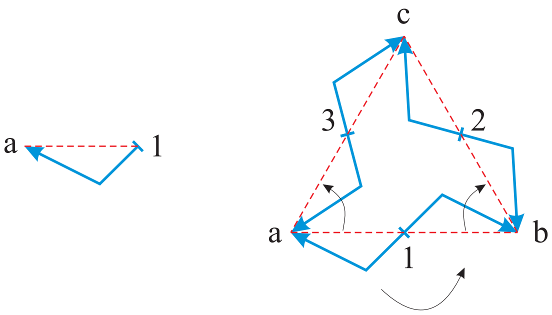
[1] M.C. Escher defines motif as a certain polygonal form that repeats itself in congruent shapes to form a tessellation [175].

two categories. The first category of tessellations are created from minor modifications of the basic shapes. The second category of tessellations are based on recognizable figures such as images of birds, animals, people and other day to day objects which one could relate to. Such tessellations use motifs with very intricate curved lines to resemble a recognizable object. M.C. Escher was a great exponent of such tessellations [175]. We use some of his designs in this work and develop some of our own based on the concepts discussed in [175].

6.2.1 Tessellations of polygons

Using a triangle as the building block and using translation and rotation operations, we create four sets of tessellations. Figure 6.3 illustrates the sequence of steps to create triangle-based tessellations. We start with the red dashed line **a1** and bend it into the blue solid line. The deformed line **a1** is rotated by 180° about **1** to obtain line **ab** which is then rotated by ±60° to obtain lines **ac** and **bc**, respectively.

Figure 6.3: The sequence of steps to create a triangle-based tessellations. We start with the red dashed line **a1** and deform it into the blue solid line. Deformed line **a1** is rotated by 180° about **1** to obtain line ab which is then rotated by ±60° to obtain lines **ac** and **bc**, respectively.



By horizontally and vertically translating the patches created using the technique illustrated in Fig. 6.3, the tessellations in Fig. 6.4 are created. Each patch/element in tessellations *Tess T1* (Fig. 6.4(a)), *Tess T2* (Fig. 6.4(b)), *Tess T3* (Fig. 6.4(c)), *Tess T4* (Fig. 6.4(d)) contains 9, 15, 15 and 27 nodes, respectively. The nodes in each elements are numbered counterclockwise and we conduct checks to ensure there are no duplicate nodes in the mesh.

The sequence of steps to generate tessellations using a quadrilateral base is similar to that for the triangule base discussed previously. We start from the red dashed line **a1** and deform it into the blue solid line. Deformed line **a1** is rotated by 180° about **1** to obtain line **ab**, which is then rotated by 90° to obtain line **ad**. Deformed blue solid lines **ab** and **ad** are translated vertically and horizontally to obtain **dc** and **bc**, respectively (Fig. 6.5).

Tessellations in Fig. 6.6 are created using the technique illustrated in Fig. 6.5. Patches in tessellations shown in Figs. 6.6(a) and 6.6(b) contain 12 and 20 nodes each, respectively.

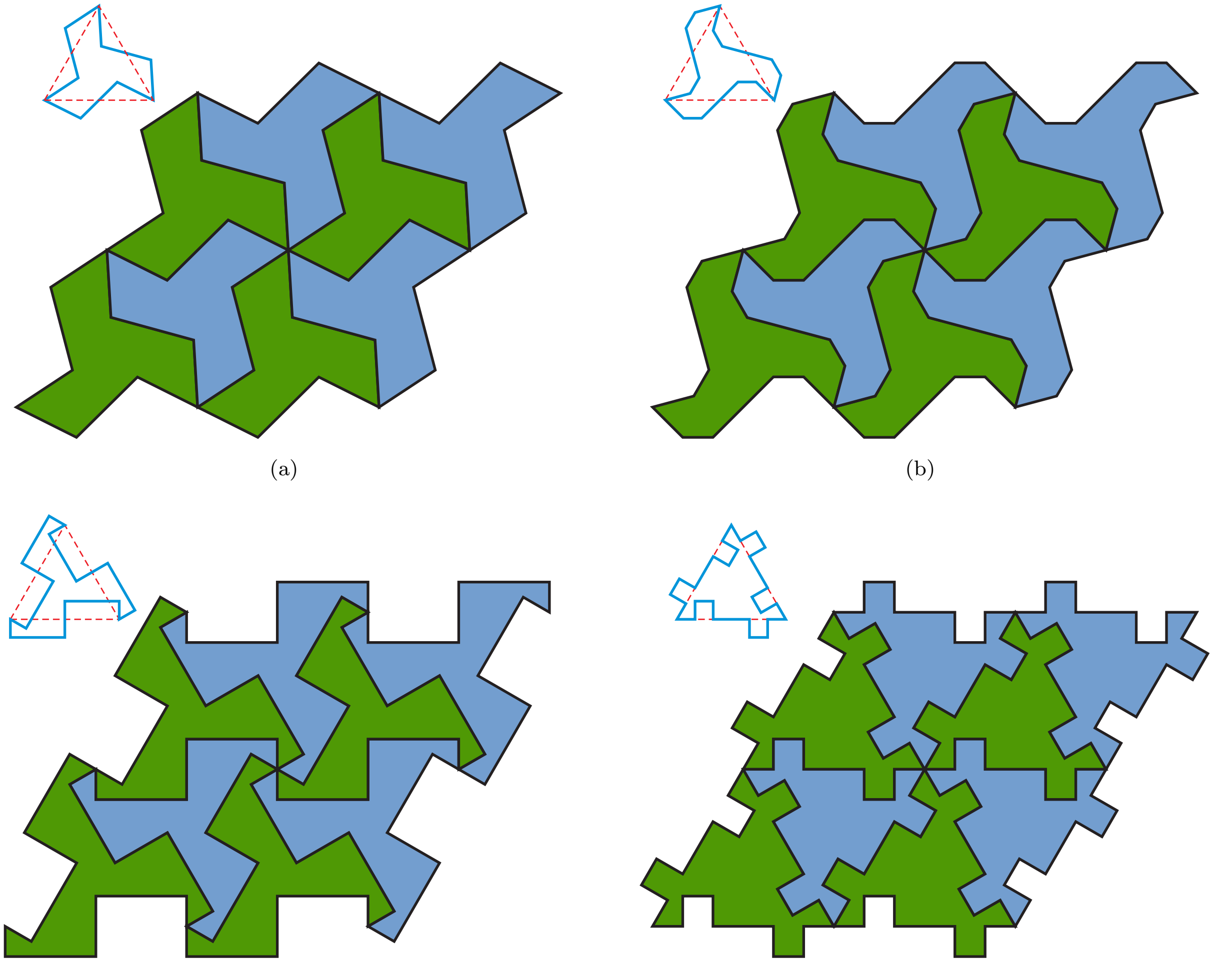

(a) (b)
(c) (d)

Figure 6.4: Tessellations generated using the patches shown in the corresponding insets. (a) *Tess T1*. (b) *Tess T2*. (c) *Tess T3*. (d) *Tess T4*.

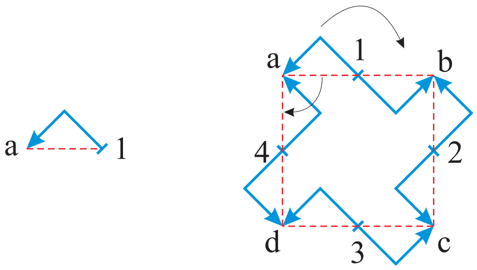
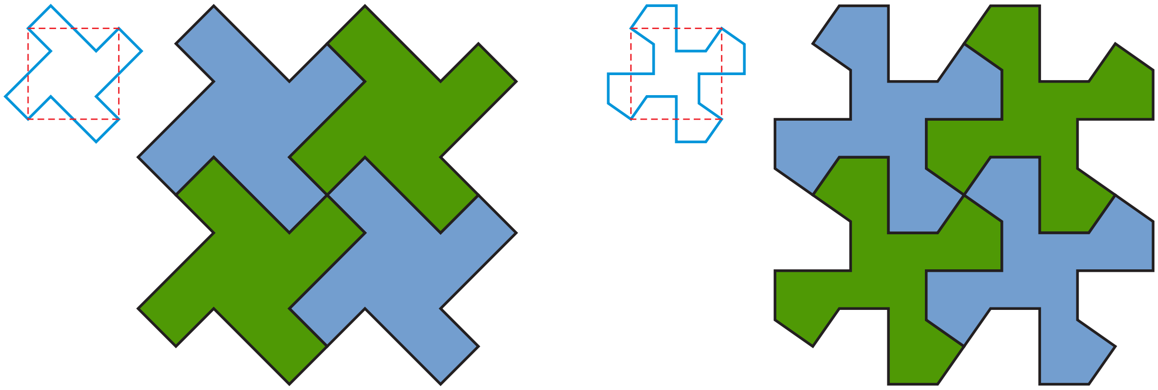


Figure 6.5: The sequence of steps to create a quadrilateral-based tessellations. We start from the red dashed line **a1** and deform it into the blue solid line. Deformed line **a1** is rotated by 180° about **1** to obtain line **ab**, which is then rotated by 90° to obtain line **ad**. Deformed blue solid lines **ab**, ad are translated vertically and horizontally to obtain **dc** and **bc**, respectively.



(a) (b)

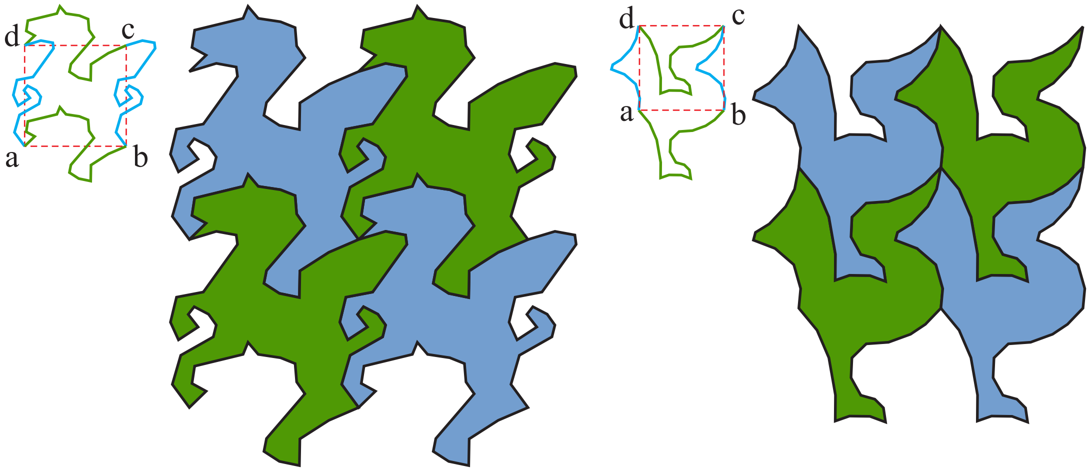
Figure 6.6: Tessellations generated using the patches shown in the corresponding insets. (a) *Tess Q1*. (b) *Tess Q2*.

6.2.2 Tessellations of recognizable figures

M.C. Escher was one of the pioneers of creating tessellations using recognizable figures such as birds, horses, fishes and lizards [175]. Escher classified his tessellations into quadrilateral systems and triangle systems. In his work, tessellations based on quadrilaterals are more common as they are easier to work with. He further subcategorizes his quadrilateral systems depending on two characteristics: the type of underlying polygon (represented by letters A, B, C, D, E) and the symmetry present in the motif (represented by Roman numerals I through X). The letters A, B, C, D and E denote parallelogram, rhombus, rectangle, square and isosceles right triangle, respectively. In this work, we will explore a few tessellations of type I^D, V^C and IV^B.

We start with Pegasus and Bird tessellations which belong to the system of type I^D. Class I^D tessellations use a square as the underlying polygon (square **abcd** in Fig. 6.7) and use translation in both transversal and diagonal directions. Edge **dc** and **da** are replaced by the curved lines as shown in Fig. 6.7. Then curved edges **dc** and **da** are translated vertically and horizontally to **ab** and **cb** to complete the motif. In our tessellations, we use a total of 66 and 52 nodes per Pegasus and Bird motif, respectively. Final tessellations are created by simply translating the motifs, obtained earlier, vertically and horizontally.

A Bulldog is a type V tessellation. Parallelogram **abcd** (Fig. 6.8) is used as the background shape. Edge **dc** represented by the red dashed line is deformed into a curved line (green colored) and is then glide-reflected (translation of the reflected image) to the bottom edge **ba**. The Bulldog motif is completed by translating the blue curved edge **da** horizontally to edge **cb**. By repeating identical copies of this motif horizontally and mirror copies vertically a Bulldog tessellation can be obtained. Escher classified this design as type V^C [175]. We have 72 nodes per Bulldog element.



(a)

(b)

Figure 6.7: Escher's tessellations of type I^D. (a) Pegasus. (b) Bird.

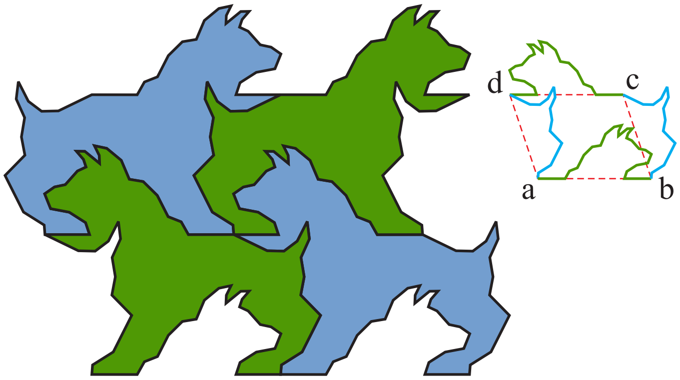


Figure 6.8: Bulldog - Escher's tessellations of type V^C

As an example of type IV^B, we look at the Winged Lion tessellation (Fig. 6.9). A quadrilateral shaped like a dart or kite is used as the starting polygon. Deformed edges **dc** and **ad** are glide-reflected to edges **cb** and **ba**, respectively. Each Winged Lion motif has 76 nodes.

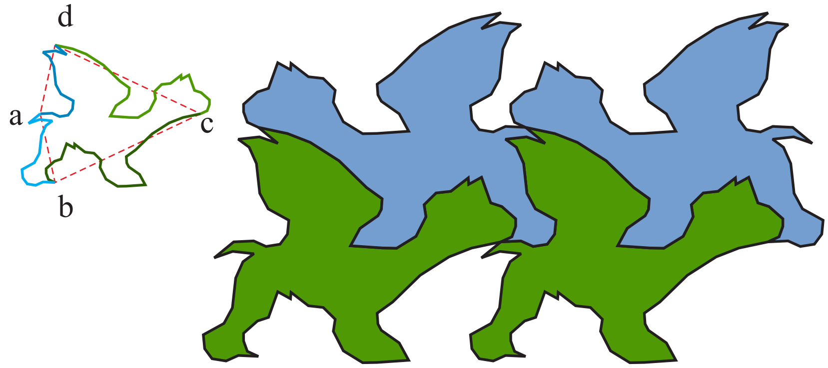


Figure 6.9: Winged Lion - Escher's tessellations of type IV^B

6.3 A MFD-inspired approach for two-dimensional linear elasticity

In order to numerically solve the linear elasticity problem (1.1, 1.2) we use a Mimetic Finite Difference-inspired approach, known as the Virtual Element Method [26, 27, 91, 173]. The details of the method for the three-dimensional linear elasticity problem were presented in Chapter 4. In this section, for the two-dimensional problem, we summarize the construction of the discrete bilinear form, also known as the stiffness matrix, and the force vector which are needed to solve the optimization problem (1.4).

Let the continuous bilinear form be split over polygonal elements, E, as:

$$a(\boldsymbol{u}, \boldsymbol{v}) = \sum_E a^E(\boldsymbol{u}, \boldsymbol{v}), \quad \forall \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{\mathcal{V}} \tag{6.1}$$

In the Virtual Element Method, the bilinear form is constructed in such a way that the displacement patch test is satisfied, i.e:

$$\begin{gathered} a_h^E(\boldsymbol{p}_1, \boldsymbol{v}_h) = a^E(\boldsymbol{p}_1, \boldsymbol{v}_h) \\ \forall E, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E, \quad \forall \boldsymbol{p}_1 \in \mathbb{P}_1(E)^2 \end{gathered} \tag{6.2}$$

Equation (6.2) implies that, when the trial function belongs to linear space $\mathbb{P}_1(E)^2$, the discrete bilinear form is exactly the same as the continuous bilinear form. Here $\boldsymbol{\mathcal{V}}_h^E \subset \boldsymbol{\mathcal{V}}$, and from the viewpoint of Galerkin approximations, represents the space spanned by the smooth shape functions which are non-negative inside the domain, form a partition of unity, satisfy the Kronecker-delta property, are linearly precise and C^0 continuous on the edges. For the discrete solution to be stable, we also need to ensure that the discrete bilinear form is continuous and bounded, i.e., we require that there exists constants $\alpha^*, \alpha_* > 0$ such that:

$$\alpha_* a^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \le a_h^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \le \alpha^* a^E(\boldsymbol{v}_h, \boldsymbol{v}_h) \quad \forall E, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h^E \tag{6.3}$$

The discrete bilinear form is constructed to have two terms, consistency and stability, which satisfy the displacement patch test and stability requirements, respectively. In order to derive the expressions for the stiffness matrix, the linear space $\mathbb{P}_1(E)^2$ is selected as the span of basis vectors $\hat{\boldsymbol{p}}_1 = [1,\ 0]^T,\ \hat{\boldsymbol{p}}_2 = [0,\ 1]^T,\ \hat{\boldsymbol{p}}_3 = [y,\ -x]^T,\ \hat{\boldsymbol{p}}_4 = [x,\ 0]^T,\ \hat{\boldsymbol{p}}_5 = [0,\ y]^T,\ \hat{\boldsymbol{p}}_6 = [y,\ x]^T$. The consistency term is taken as $a_h^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = a^E(\Pi^E\boldsymbol{\phi}_i, \Pi^E\boldsymbol{\phi}_j)$, where the projection $\Pi^E : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1(E)^2$ needs to be constructed. For discussion on the rationale for such a choice, refer to [26, 27, 91, 173]. Since the discrete bilinear form involves calculating the strains of the arguments, the contribution of the rigid body terms is lost as they have zero strains. So we treat the rigid body motions separately and split the linear space, $\mathbb{P}_1(E)^2$, and projection Π^E as follows:

$$\mathbb{P}_1(E)^2 = \mathbb{P}_1^0(E)^2 \oplus \mathbb{P}_1^1(E)^2, \qquad \Pi^E = \bar{\Pi}^E + \Pi^0(\mathbf{I} - \bar{\Pi}^E) \tag{6.4}$$

Here, $\mathbb{P}_1^0(E)^2$ and $\mathbb{P}_1^1(E)^2$ represent the rigid body and constant strain spaces, respectively. And the projections are defined as, $\bar{\Pi}^E : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1^1(E)^2$ and $\Pi^0 : \boldsymbol{\mathcal{V}}_h^E \to \mathbb{P}_1^0(E)^2$. Note that $a^E(\Pi^E\boldsymbol{\phi}_i, \Pi^E\boldsymbol{\phi}_j) = a^E(\bar{\Pi}^E\boldsymbol{\phi}_i, \bar{\Pi}^E\boldsymbol{\phi}_j)$. Thus, the consistency term requires determining the projection $\bar{\Pi}^E$. After some algebraic manipulations, the consistency term, represented by $\boldsymbol{K}_1$, can be written as [91, 173]:

$$\boldsymbol{K}_1 = a^E(\bar{\Pi}^E\boldsymbol{\phi}_i, \bar{\Pi}^E\boldsymbol{\phi}_j) = \left[\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\mathbf{R}^T\right]_{ij} \tag{6.5}$$

where $\mathbf{N}$ is the nodal coordinate matrix of dimension $2N \times 3$, N being the total number of nodes in an element and is expressed as:

$$\mathrm{N}_{2i-1:2i,:} = \begin{bmatrix} x_i & 0 & y_i \\ 0 & y_i & x_i \end{bmatrix}, \qquad i = 1, ..., N \tag{6.6}$$

Matrix $\mathbf{R}$ is also of dimension $2N \times 3$ and is calculated as (c.f. Fig. 6.10):

$$\begin{aligned} \mathrm{R}_{2i-1:2i,:} &= \int_{\partial E} \boldsymbol{\phi}_i \cdot (\mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\alpha))\, \mathbf{n}\, d\boldsymbol{s}, \qquad i = 1, ..., N \\ &= \sum_{j=1}^{N_{neighbor}} \left((\mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\alpha))\, \mathbf{n}_j \cdot \int_{\partial E} \boldsymbol{\phi}_i \, d\boldsymbol{s}_j \right) \\ &= \mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\alpha) \left(e_{i-1}\mathbf{n}_{i-1} + e_i \mathbf{n}_i \right)/2 \end{aligned} \tag{6.7}$$

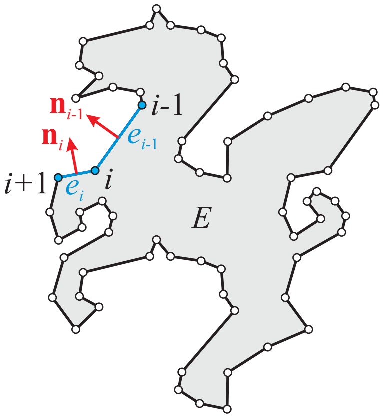


Figure 6.10: Illustration of the adjacency information needed in evaluation of line integral in the consistency term. Lengths of the sides adjacent to node i are represented by e_{i-1}, e_i. The arrows represent the outward pointing normals, $\mathbf{n}_{i-1}$, $\mathbf{n}_i$.

Here, lengths of the sides adjacent to node i are represented by e_{i-1} and e_i. The arrows represent the outward pointing normals, $\mathbf{n}_{i-1}$ and $\mathbf{n}_i$. The discrete bilinear term is constructed to contain a second term, $\boldsymbol{K}_2 = S^E((\mathbf{I} - \Pi^E)\boldsymbol{\phi}_i, (\mathbf{I} - \Pi^E)\boldsymbol{\phi}_j)$, which imparts stability to the solution. The second term is chosen to be a symmetric positive definite matrix which is continuous and bounded. It is also constructed in such a way that the appended discrete bilinear form still satisfies the displacement patch test. Using (6.4), $\boldsymbol{K}_2$ can be simplified as:

$$\boldsymbol{K}_2 = (\mathbf{I} - \mathbf{P}_M^T)(\mathbf{I} - \mathbf{P}_1^T) S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)(\mathbf{I} - \mathbf{P}_1)(\mathbf{I} - \mathbf{P}_M) \tag{6.8}$$

where $\mathbf{P}_M$ is the matrix representation of the projection $\bar{\Pi}^E$ and is equivalent to $\left[\mathbf{N}(\mathbf{R}^T\mathbf{N})^{-1}\mathbf{R}^T\right]$. Also, $\mathbf{P}_1$ is the matrix representation of the projection operator Π^0 and we define it as:

$$\Pi^0 \boldsymbol{v} = \bar{\boldsymbol{v}} + \left\langle \frac{1}{|E|} \int_E \nabla \boldsymbol{v} d\boldsymbol{x} \right\rangle (\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{6.9}$$

The projection Π^0 in (6.9) is chosen such that, if $\boldsymbol{v} \in \mathbb{P}_1^0(E)^2$ (i.e. $\boldsymbol{v} = \sum_{i=1}^3 c_i \hat{\boldsymbol{p}}_i$ where c_i's are constants), then $\Pi^0 \boldsymbol{v} = \boldsymbol{v}$. Finally, $S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$ needs to be chosen such that it is a positive definite matrix and should scale like the consistency term. Let, $S^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$ be $\alpha\delta_{ij}$, where α is a scaling factor. So, the final stiffness matrix expression looks like,

$$a_h^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \boldsymbol{K}_{ij} = \left[\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\mathbf{R}^T\right]_{ij} + \left[\alpha(\mathbf{I} - \mathbf{P}_M^T)(\mathbf{I} - \mathbf{P}_1^T)(\mathbf{I} - \mathbf{P}_1)(\mathbf{I} - \mathbf{P}_M)\right]_{ij} \tag{6.10}$$

In this work, the scaling factor α is chosen as $\alpha^* trace(\boldsymbol{K}_1)$, where α^* is a scaling coefficient chosen based on a parametric study conducted in Section 6.4.2.

For the numerical examples in Section 6.4, we only consider concentrated point loads. The global force vector $\mathbf{F}$, corresponding to those loads consists of all zeros except at the degrees of freedom corresponding to the point loads, where it takes the value of the loads.

6.4 Numerical verification studies

Here, we demonstrate the accuracy of the present numerical approach using the displacement patch test and shear loaded cantilever beam bending problems. The purpose of the numerical verification studies is to show that the current approach produces numerically convergent and stable results for tessellations of arbitrary shape including concave elements. The accuracy and convergence of the numerical results are verified in terms of relative L^2 error norms. The displacement error norm is the first metric we use and is expressed as:

$$U_{Err} = \frac{|\mathbf{U}_{exact} - \mathbf{U}_{VEM}|}{|\mathbf{U}_{exact}|} \tag{6.11}$$

where $\mathbf{U}_{exact}$ and $\mathbf{U}_{VEM}$ are the exact and VEM solutions, respectively. We also utilize a stress error norm to measure the accuracy of the method. The error measure that we use for the stresses, $\boldsymbol{\sigma}$ (written in Voigt notation), is defined as:

$$\bar{\sigma}_{Err} = \frac{|\boldsymbol{E}_1|_2}{|\boldsymbol{E}_2|_2} \tag{6.12}$$

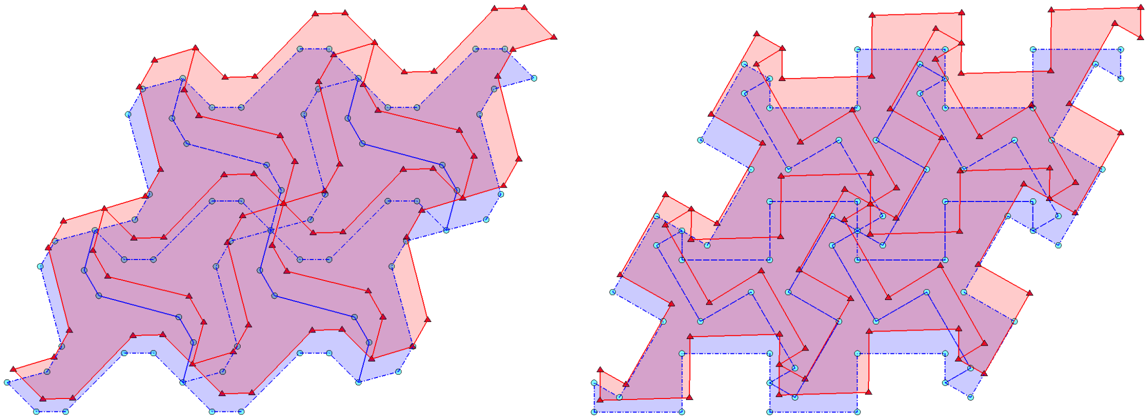
Matrices $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ are of dimension (Number of elements) × 3 and constructed such that each row of $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ corresponds to $\int_E (\boldsymbol{\sigma}_{exact} - \bar{\boldsymbol{\sigma}}_{VEM,e})^2\, d\boldsymbol{x}$ and $\int_E (\boldsymbol{\sigma}_{exact})^2\, d\boldsymbol{x}$, respectively, which are numerically integrated over each element. Here, $\boldsymbol{\sigma}_{exact}$ is the exact stress and $\bar{\boldsymbol{\sigma}}_{VEM,e}$ is the VEM average element stress calculated as [91]:

$$\bar{\boldsymbol{\sigma}}_{VEM,e} = \boldsymbol{u}_e^T \left(\mathbf{R}(\mathbf{N}^T\mathbf{R})^{-1}\right) \mathbf{C}\boldsymbol{\varepsilon}(\hat{\boldsymbol{p}}_\beta) \tag{6.13}$$

For numerical integration, we triangulate each element and use the standard integration rules of linear T3 element to obtain the location of Gauss points and their corresponding weights. The relative L^2 error norms are plotted against the maximum of the element diameter in the mesh. An element diameter is defined as the maximum pairwise distance of all the vertices of the element.

6.4.1 Displacement patch test

We conduct the displacement patch tests for all the tessellations shown in Figs. 6.4, 6.6 6.7, 6.8 and 6.9. An arbitrary linear displacement of the form $\mathbf{U} = \mathbf{AX} + \mathbf{B}$, where $\mathbf{A} = 1/100\,[2\ 1;\ 3\ 4]$ and $\mathbf{B} = 1/100\,[1\ 2]^T$, is applied to all the nodes on the boundary of the mesh and no forces are applied to the system. Scaling factors of the form $\alpha^* trace(\boldsymbol{K}_1)$, with $\alpha^* \in [10^{-3} - 10^3]$ are investigated. For all tessellations, the relative L^2 displacement error and relative L^2 average stress errors are in the machine precision range (10^{-16}) indicating that the VEM passes the displacement patch test exactly. The tests indicate that the choice of the scaling factor has no impact on the patch test results. Figure 6.11 illustrates the deformed configurations from the patch test for the triangle-based tessellations *Tess T2* and *Tess T3*.



(a)

(b)

Figure 6.11: Displacement patch test on triangle-based tessellations (a) *Tess T2*, (b) *Tess T3*. Blue colored mesh with circular nodes represents the initial configuration and red colored mesh with triangular nodes represents the deformed configuration.

6.4.2 Shear loaded cantilever beam bending

Next, we study the performance of the present numerical approach using the shear loaded cantilever beam bending problem. Consider a rectangular beam of length a and width $2b$, $0 < x < a$, $-b < y < b$, subjected to transverse shear load, $F = 0.1$, at end $x = 0$. For the current study, $a = 10$ and $2b = 1$. The expressions for stresses are available in [213, 23] and are repeated here for completeness.

$$\sigma_{xx} = \frac{3Fxy}{2b^3}, \qquad \sigma_{yy} = 0, \qquad \sigma_{xy} = \frac{3F\left(b^2 - y^2\right)}{4b^3} \tag{6.14}$$

Using the stress-strain relationships and definitions of displacement gradients, the displacement fields corresponding to the stresses (6.14) are written as:

$$\begin{aligned} u &= \frac{3Fx^2y}{4Eb^3} + \frac{3F(1+\nu)y}{2Eb} - \frac{F(2+\nu)y^3}{4Eb^3} + A - Cy, \\ v &= -\frac{3F\nu xy^2}{4Eb^3} - \frac{Fx^3}{4Eb^3} + B + Cx \end{aligned} \tag{6.15}$$

Here, u and v represent the displacements in the x and y directions, respectively. Constants A, B and C account for the three rigid body motions in two-dimensions.

In the VEM formulation, the choice of scaling factor for the stability term in the stiffness matrix (6.10) needs to be chosen carefully, as it affects the accuracy of results. We chose the scaling factor as $\alpha^*\ trace(\boldsymbol{K}_1)$. We conducted a parametric study to determine feasible choices of α^*, where α^* is varied in the range $[10^{-3} - 10^{3}]$. The study utilized two mesh discretizations of 490 and 810 elements of the tessellation *Tess Q1*. Figure 6.12 shows the relative L^2 displacement error (6.11) and relative L^2 average stress error (6.12) plotted against scaling coefficient α^*. Illustrated in Fig. 6.12, the relative L^2 error curves attain a minimum somewhere in the range $[0.0, 0.1]$. Similar results are obtained for other mesh discretizations and are not shown here. Based on this study, we chose $\alpha^* = 0.1$.

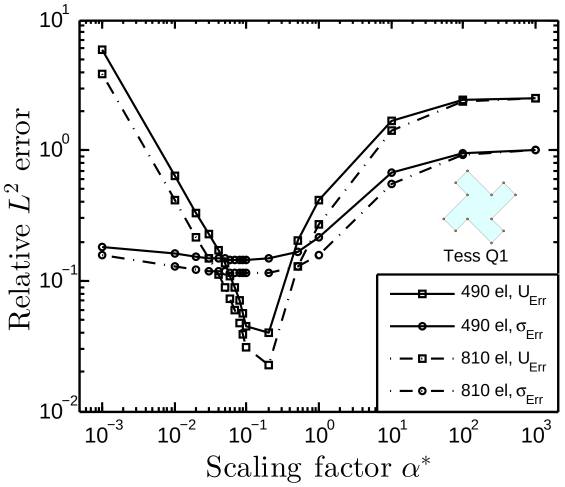
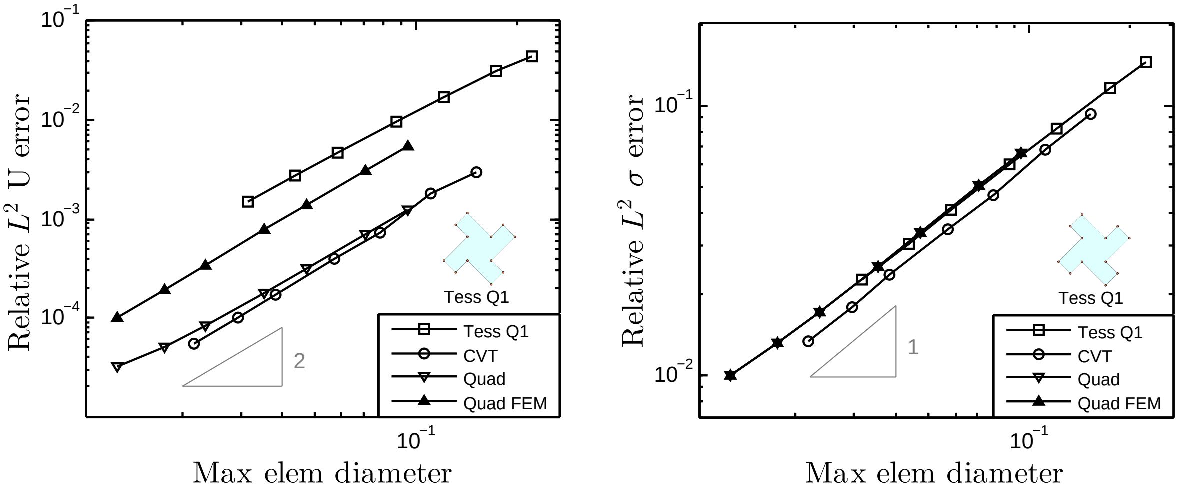


Figure 6.12: Parametric study of the scaling coefficient α^* for the shear loaded cantilever beam problem using two discretizations of 490, 810 elements of the tessellation *Tess Q1* shown in the inset.

The current approach can be applied to tessellations/polygons of any arbitrary shape, including concave elements, however the shape of the element dictates the accuracy of the numerical simulations. We evaluate the performance of the approach for tessellations *Tess Q1*, centroidal Voronoi tessellation (CVT) meshes and uniform quad meshes and compare it with FEM results on uniform quad meshes. The results of the study are shown in Fig. 6.13 where the relative L^2 errors are plotted against the maximum element diameter of the meshes. We note that, even for tessellation *Tess Q1*, which essentially contains concave elements, a second order of convergence is obtained for displacement errors (Fig. 6.13(a)) and a first order of convergence for average stress errors (Fig. 6.13(b)). We observe that, for a given maximum element diameter, tessellation *Tess Q1* produces the highest displacement error (Fig. 6.13(a)). On the other hand, the relative L^2 average stress errors showed almost no noticeable difference (Fig. 6.13(b)) among different meshes using the VEM and FEM.



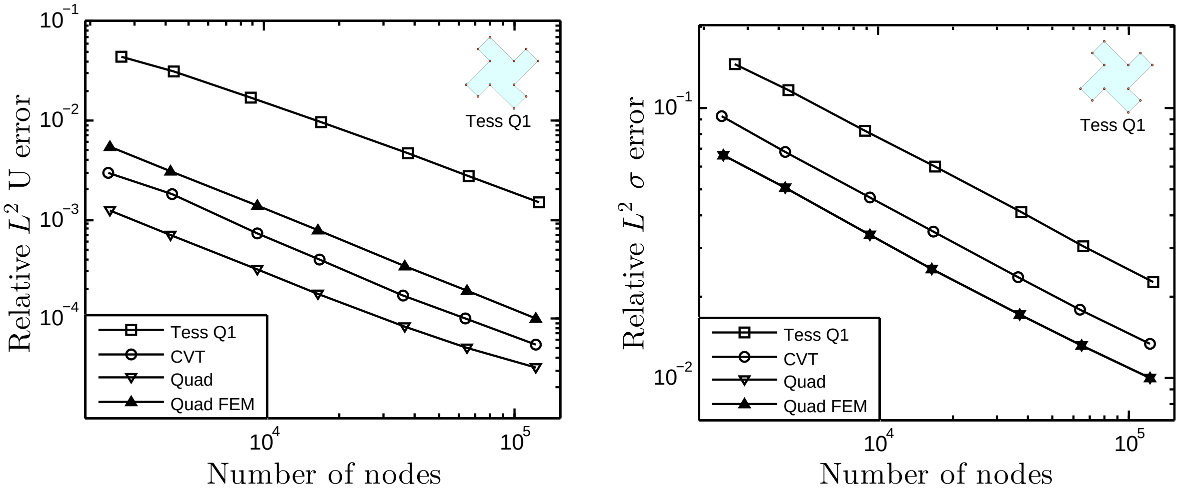
(a) (b)

Figure 6.13: Comparison of the performance of the VEM on different meshes (*Tess Q1*, CVT Voronoi mesh, uniform quad mesh) and the FEM on a uniform quad mesh. (a) L^2 displacement error. (b) L^2 average stress error.

Evaluating the accuracy of the numerical method on different meshes based on error estimates plotted against maximum element diameters may not be the most appropriate measure. Error estimates with respect to the total number of nodes in the mesh may provide better insight into the effectiveness of the method (Fig. 6.14). As a frame of reference, we included the relative L^2 errors computed from FEM on uniform quad meshes. In terms of displacement errors, the current numerical approach on uniform quad meshes seems to produce the lowest L^2 error. For the average stress errors, the difference is a bit more apparent. It is interesting to note that, the relative L^2 average stress errors are almost identical for analysis using the current approach and FEM on uniform quad meshes.

6.5 Generation of optimal and artistic designs

In the construction industry, a landmark design comes about as a consequence of a synergistic blend of architecture and engineering. In general, this is a difficult task to achieve. Engineers focus on objectives such as structural stability, efficiency and efficient load transfer. Architects, on the other hand, are concerned with the style and appearance, i.e. aesthetics of the design. More often than not, the goals of an engineer and architect conflict. Since both structural performance and aesthetics are crucial to the building design, often a compromise



(a) (b)

Figure 6.14: Comparison of the performance of VEM on different meshes (*Tess Q1*, CVT Voronoi mesh, uniform quad mesh) and FEM on uniform quad mesh with respect to total number of nodes in the mesh. (a) L^2 displacement error. (b) L^2 average stress error.

is arrived at. The goal of the current work of topology optimization on tessellated meshes is to provide a natural means to integrate artistic aspects in engineering, thus satisfying both engineers and architects. This is illustrated using three examples - a cantilever beam with circular support, a bracing system design and a bridge problem. The Young's modulus and Poisson's ratio are selected as 1.0 and 0.3, respectively and consistent units are employed. We chose Optimality Criteria (OC) as the optimization algorithm. Optimization is terminated when either the maximum of the change in element densities is less than 0.01 or the maximum iterations exceed 150.

6.5.1 Cantilever beam with circular support

We start with the benchmark problem of a cantilever beam with circular support [209]. The problem description is shown in Fig. 6.15. All the nodes along the circular segment are fixed and a point load, acting in the negative y-direction, is applied approximately in the middle of the right edge. The domain is discretized using Escher's type I^D tessellations - Pegasus and Bird (Fig. 6.7). The Pegasus tessellation contains 1,441 elements (47,603 nodes) and the Bird tessellation has 1,442 elements (37,261 nodes). The volume fraction and penalty factor for the SIMP model are chosen as 0.3 and 3.0, respectively.

For the first case, optimization is performed without any filters. As illustrated in Fig. 6.16 the overall converged topologies are similar to the ones available in the literature [209]. In this work, we are more interested in the details around the boundaries of the members. In Fig. 6.16, outlines of the Pegasus and Bird are clearly visible and their heads and feet

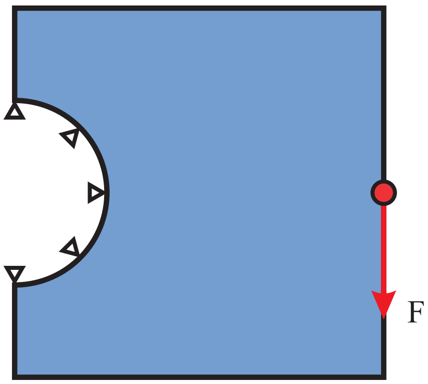
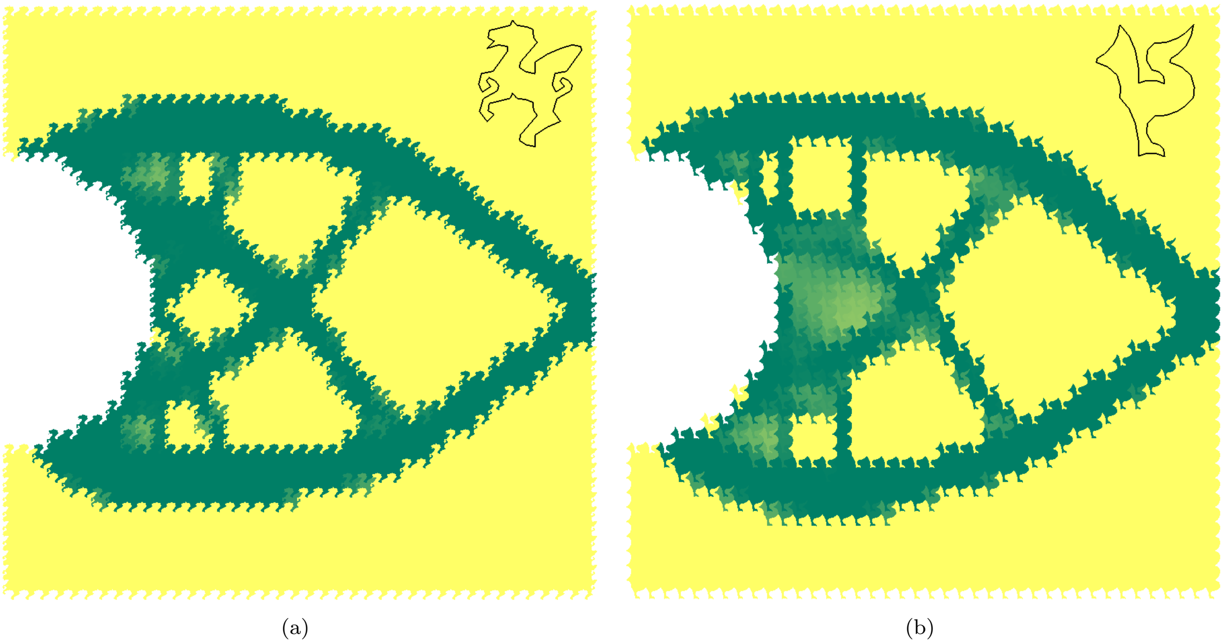


Figure 6.15: Cantilever beam with circular support.

are well defined. Aesthetically, in our opinion, such designs are much more appealing than the optimization designs on triangle and quadrilateral meshes.

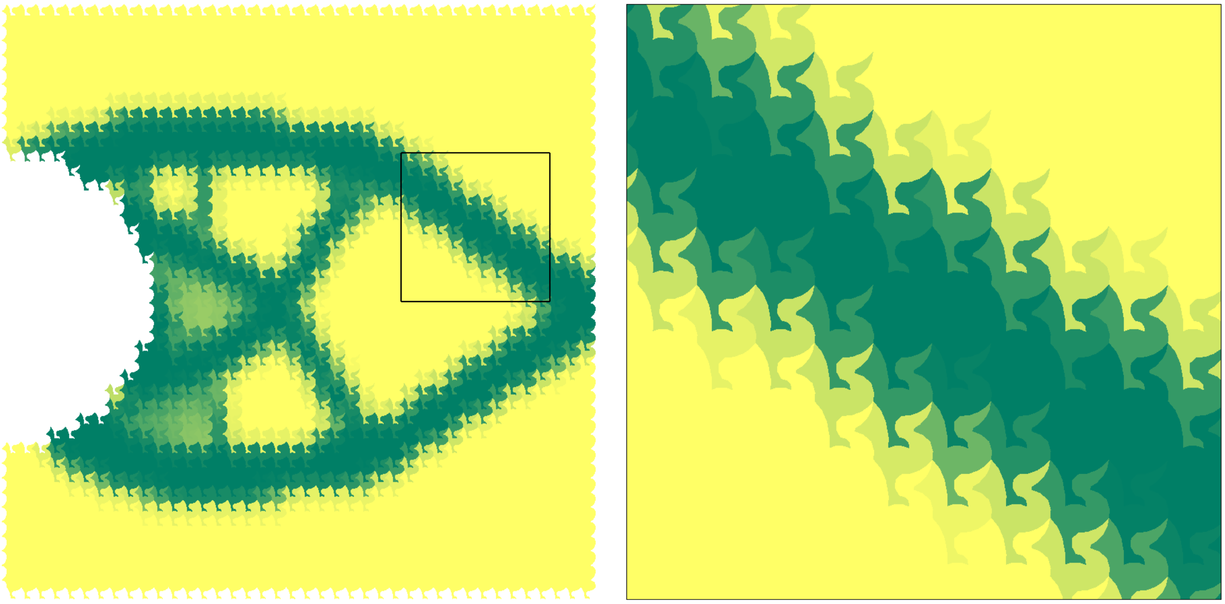

(a)

(b)

Figure 6.16: Converged topologies for the cantilever beam with circular support. The volume fraction and SIMP penalty factor are chosen as 0.3 and 3, respectively and no filter is used. (a) Pegasus tessellation. (b) Bird tessellation.

In the previous designs (Fig. 6.16), we see that, due to the absence of filters, the boundaries, in general, are distinct. Artistically it would be much more attractive if there were a gradation at the boundaries, with elements of varying densities. To obtain such a design, we perform optimization on the cantilever beam problem discussed earlier, employing a linear filter of radius equal to 3% of the maximum domain dimension. The other parameters are kept the same. In the zoomed in section (Fig. 6.17(b)) of the optimized design, Fig.

6.17(a), an elegant design pattern can be seen. In our opinion, such designs will not only be appreciated by an engineer for being structurally optimal and stable but also by architects due to their aesthetic appeal.


(a)

(b)

Figure 6.17: Converged design for cantilever beam with circular support problem solved on Bird tessellation, using a linear filter of radius 3% of the maximum domain dimension, is shown in (a). Volume fraction and SIMP penalty factor are taken as 0.3 and 3, respectively. (b) Zoomed in section.

6.5.2 Bracing system design

Next example focuses on the design of more practical systems, the bracings. In buildings, they provide lateral resistance to withstand loadings such as the wind and seismic forces. We consider a model problem shown in Fig. 6.18. Nodes along the bottom edge are fixed and lateral loads are applied at the top corners. The design domain is discretized using Escher's type V^C tessellation - Bulldog, consisting of 690 elements and 25,063 nodes. A linear filter of radius equal to 3% of the maximum domain dimension is adopted along with a volume fraction and SIMP penalty factor of 0.3 and 3, respectively.

As expected, the optimization algorithm produces an X-bracing system (Fig. 6.19) to resist lateral loads. As before, we can identify the outline of the Bulldogs and their graded pattern which add an artistic touch to an otherwise bland engineering design. To design bracing systems for high-rise buildings, manufacturing and layout constraints such as pattern repetition and pattern gradation can be employed [12, 193]. In pattern repetition manufacturing constraint utilizes the same design pattern for multiple stories to help increase the speed of construction and ensure high quality. Pattern gradation can be used for a smooth

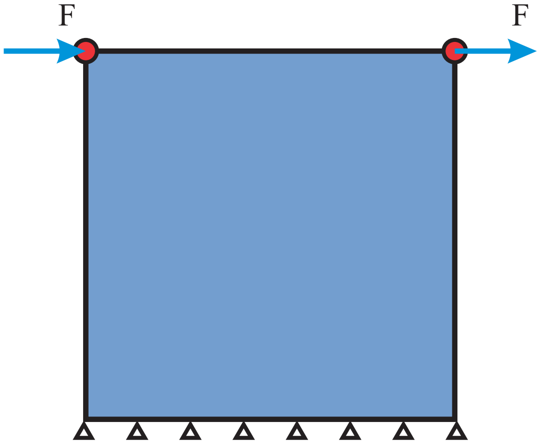


Figure 6.18: Problem description for bracing system design.

transition of the design between two dissimilar loading resistance requirements, such as for tall building where overturning moments are higher at the bottom and shear is dominant at the top.

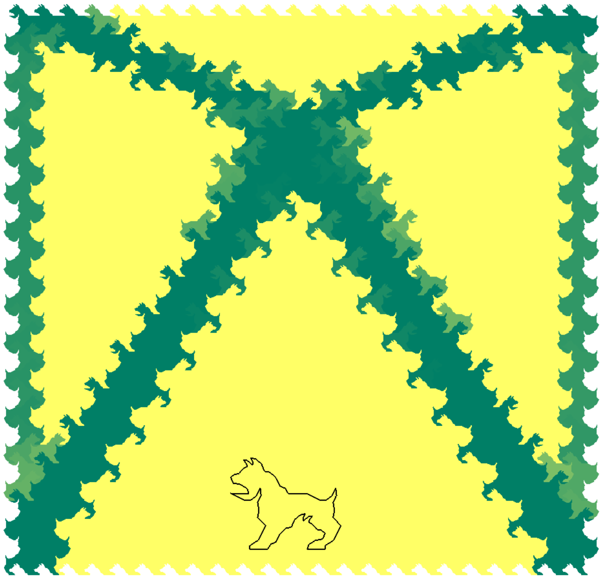


Figure 6.19: Structurally efficient and aesthetically pleasing bracing design using Bulldog tessellation.

6.5.3 Bridge problem

The final example is the optimization of the bridge shown in Fig. 6.20. The locations of the piers are indicated by pin and roller supports. Two point loads, representing dead weights, are applied at the locations shown in Fig. 6.20. Escher's type IV^B tessellation - Winged Lion, is used to discretize the design domain. The mesh consists of a total of 3,200 elements and 121,948 nodes. To obtain a graded pattern at the design boundaries, a linear filter of radius equal to 5% of the minimum domain dimension is used. The other parameters are kept the same as in previous examples.

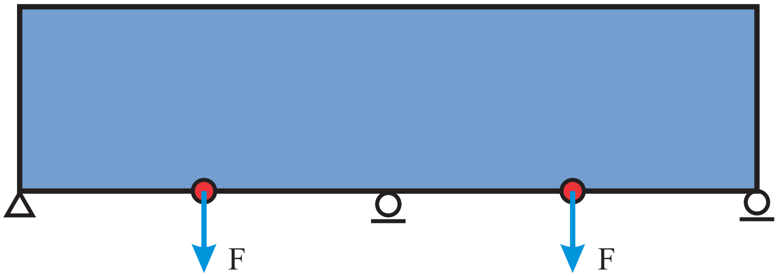
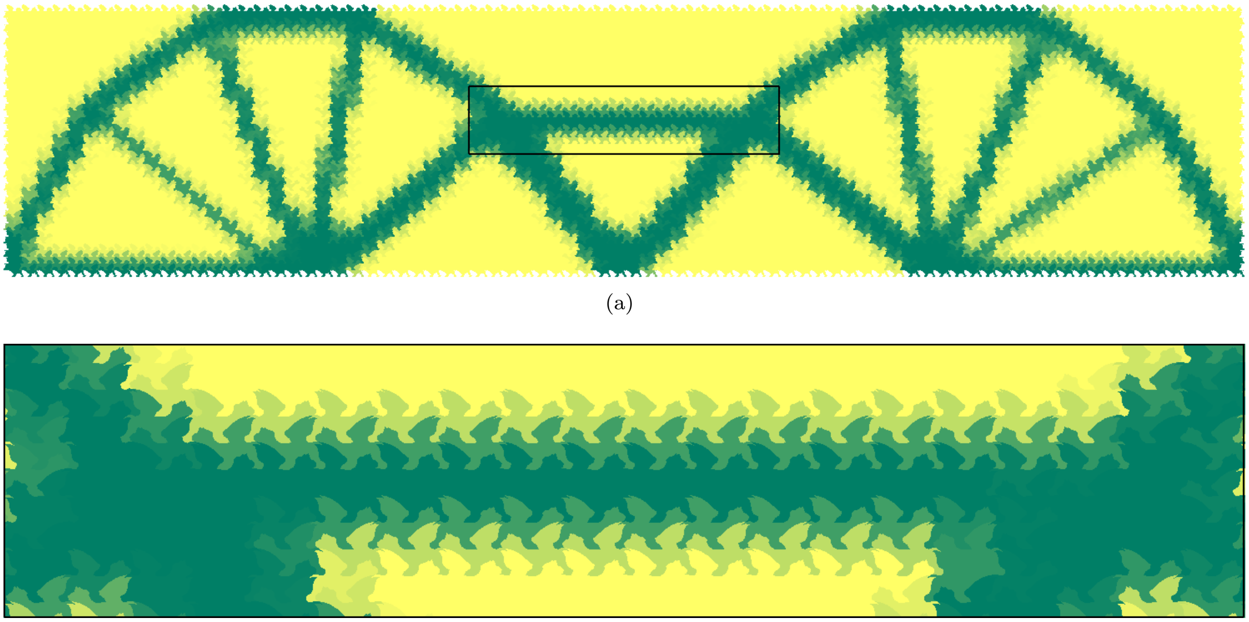


Figure 6.20: Problem description for the bridge design.

The converged design (Fig. 6.21(a)) resembles a typical bridge with two fan like structural segments. Figure 6.21(b) illustrates the captivating design details at the boundaries. In our opinion, an architect would more often than not choose this design over ones with zigzag boundaries that are typically obtained from topology optimization using typical finite elements such as quadrilaterals. To design bridges with multiple spans, the previously discussed pattern repetition concept can be adopted.


(a)

(b)

Figure 6.21: Converged bridge design solved on Winged Lion tessellation, using a linear filter of radius 5% of the minimum domain dimension, is shown in (a). Volume fraction and SIMP penalty factor are chosen as 0.3 and 3, respectively. (b) Zoomed in section.

6.6 Concluding remarks

In this work, we investigate topology optimization using tessellations as a means to bridge diverse fields such as engineering, art and mathematics. A special numerical approach is needed to handle tessellations; which can often be highly skewed, degenerate and non-convex in nature. The Mimetic Finite Difference method-inspired Virtual Element Method provides an elegant way to handle any non self-intersecting closed polygons. In this approach, the approximation space on each element is split into spaces spanned by linear and higher order polynomials. The discrete bilinear form is constructed to be continuous, bounded and pass the displacement patch test exactly.


(a) (b)

Figure 6.22: Architectural designs of Bjarke Ingels Group inspired by basic shapes such as circles and Voronoi polygons (www.big.dk). (a) REN building. (b) Warsaw museum of modern art.

We discuss the construction of tessellations, including ones created using recognizable figures, which are used as the background meshes to generate artistic designs. The current numerical approach to solve the linear elastic system is tested using the displacement patch test and shear loaded cantilever beam bending problems. All the tessellations used in this work pass the displacement patch test exactly. For the shear loaded cantilever beam bending problem, second and first order rates of convergence are obtained for relative displacement and average stress errors, respectively, for different tessellations with varying degrees of accuracy. We make use of compliance minimization (stiffest structure) topology optimization to obtain our artistic designs. Other objectives such as minimization of lateral drift, maximization of the fundamental eigenvalue and maximization of the minimum critical buckling load can be used [39]. We illustrate three optimization examples - a cantilever beam with circular support, a bracing system design and a bridge problem. We obtain structurally optimal, stable and aesthetically appealing designs which, we feel will be appreciated by both engineers and architects. Architectural firms such as Bjarke Ingels Group have been using basic shapes such as circles and Voronoi polygons (c.f. Fig. 6.22) in their designs. We hope that our approach of topology optimization using tessellations acts as a source of inspiration for architects and engineers alike who strive to create mechanically sound innovative designs.

Other potential applications of tessellation-based topology optimization is the design of high-strength composites. Tessellations such as Pegasus and Bulldog contain interlocking elements which provide natural resistance to shear and axial load. These self-penetrating elements posses higher edge length to face area ratio than typical finite elements such as triangles and quads, thus are better suited to model grains for high toughness composite design. The design of masonry structures could also benefit from tessellations. Masonry is a form of durable construction which has been around for centuries. Masonry construction utilizes bricks, stones, concrete blocks as the construction units laid in and bound together using mortar. These units could be shaped as tessellations, which provide natural toughness due to their self-penetrating nature. Tessellation-based masonry walls could be customized using different patterns which will not only be aesthetically pleasing but also structurally strong. Also, tessellated construction units will reduce the need of mortar as binding material due to their inherent interlocking nature.

Chapter 7

Conclusions and extensions

In this work, a framework for linear elastic topology optimization using polytopes is presented. A Mimetic Finite Difference-inspired approach is used to accurately and efficiently solve the governing linear elasticity problem. The current numerical approach is verified using numerical studies such as the displacement patch test and shear loaded cantilever beam bending problems. The topology optimization framework is illustrated using various compliance minimization and compliant mechanism problems. Also, recently emerged differential equation-driven methods for topology optimization, such as level-set and phase field, are investigated. To conclude the work, aesthetics in topology optimization is explored using M.C. Escher's tessellations of recognizable figures. In this chapter, the thesis and its contributions are summarized and suggestions for future work are provided.

7.1 Concluding remarks

Various topology optimization approaches and polytope-based numerical methods are explained in Chapter 1. Density-based and differential equation driven methods for topology optimization are elucidated and their literature background is provided. Chapter 1 also explains two main categories of polytope-based numerical methods - polygonal/polyhedral shape function-based methods and Mimetic Finite Difference methods. The statement of the topology optimization problem, explored in this thesis, is also discussed in this chapter.

In Chapter 2, a critical comparative study of the differential equation-driven methods for two-dimensional topology optimization is conducted. Four level-set methods and one phase-field method, which are representative of the literature, are exhaustively analyzed using compliance minimization topology optimization problems. The main differences between the two methods are as follows. In level-set methods, the design variables are level-set functions, which are generally updated using the Hamilton-Jacobi equation and later the design variables are mapped onto the density field for physical interpretation of the design. On the other hand, in phase-field methods, the design variables are the same as the density variables and the design is updated using the Allen-Cahn equation. The investigation demonstrates that the evolution equations for the level-set methods and the phase-field method, that were examined in this work, can be regarded as variations of the steepest decent method. For compliance minimization, all the methods strive to minimize the strain energy in the system by placing more material at high strain energy locations. Thus, both the differential equation-driven methods and the gradient-based density methods for topology optimization follow similar design philosophies. Most differential equation-driven methods show similar behavior when used for compliance minimization problems, but might exhibit drastically different behavior for objective functions such as compliant mechanism and non-linear elasticity. Thus, to ascertain the robustness of any method, it needs to be evaluated against all these objectives.

The Allen-Cahn evolution equation-based phase-field method for topology optimization in two-dimensions is further investigated in Chapter 3. Merits of polygonal elements are also explored in this chapter. A centroidal Voronoi tessellation (CVT) based finite volume scheme is used to solve the governing Allen-Cahn equation on unstructured polygonal meshes. The features of the approach are demonstrated using various compliance minimization and compliant mechanism problems on complicated domains. The phase-field method, implemented here, is similar to gradient-based density methods in many aspects. Both methods have the same domain representation and their optimization algorithms are primarily driven by sensitivity information. The main difference between the two methods is in the updating algorithms they employ. Mathematical programming algorithms such as, Method of Moving Asymptotes (MMA) and Optimality Criteria (OC) are used to update the designs in gradient-based density methods whereas, the phase-field method uses the Allen-Cahn evolution equation.

In Chapter 4, a Mimetic Finite Difference-inspired approach, known as the Virtual Element Method (VEM), for three-dimensional elasticity, is explored. Similar to the finite element method, the VEM is a Galerkin method with an underlying approximation space defined according to the partition of the domain. It is distinct from classical finite elements in that the approximation space is not computed explicitly. Rather the discrete bilinear form and load linear form are constructed directly based on the kinematic decomposition of the element deformation states into rigid body, constant strains and higher order modes. The construction captures the linear deformations exactly, guaranteeing satisfaction of the engineering patch test and optimum numerical convergence. Explicit expressions for the projection maps are derived to extract the linear displacement component and to subsequently construct the expressions for the stiffness matrix and load vector. Surface integrals encountered during the construction of the stiffness matrix are evaluated numerically using nodal quadrature and a conforming interpolant quadrature scheme. To verify the formulation and implementation, numerical studies such as the displacement patch test and shear loaded cantilever beam bending problem are presented. The Virtual Element Method is especially attractive, because under the same framework of splitting the approximation space into linear and higher order spaces, a higher order VEM can be formulated for any general polyhedral element. Appropriate degrees of freedom (DoF), such as DoFs defined on the edges and internal points, need to be chosen and suitable basis functions needs to be defined to span the required polynomial space.

Utilizing the numerical scheme presented in Chapter 4 for linear elasticity, a framework for three-dimensional topology optimization is developed in Chapter 5. The effectiveness of the optimization scheme and flexibility of polyhedral meshes to discretize non-Cartesian design domains is illustrated using various compliance minimization and compliant mechanism problems. The thin disc and hollow cylinder topologies indicate that the unstructured polyhedral elements capture member orthogonality and thus alleviate any mesh bias in the design. Numerical artifacts, such as single node connections, are also alleviated.

As discussed previously, a mimetic-inspired Virtual Element Method provides greater flexibility in the geometric shapes of the admissible elements. To take advantage of this fact, Chapter 6 explores two-dimensional topology optimization on tessellations as a means to coalesce engineering and art. M.C. Escher's tessellations of recognizable figures, such as birds and animals, are used to add aesthetics to otherwise bland engineering designs. The goal of the topology optimization examples shown in this chapter, such as the bracing system and bridge problems, is to provide a source of inspiration to architects and engineers to strive for innovative designs which are also mechanically sound.

7.2 Suggestions for future work

The unstructured nature of polyhedral elements offers many benefits in engineering applications such as topology optimization and fracture mechanics simulations. Some potential extensions of the current work are suggested below.

Large-scale three-dimensional topology optimization using polyhedrons

Application of polyhedral topology optimization to practical problems; such as, large scale structures or complex materials, requires three-dimensional models with hundreds of thousands of degrees of freedom. One option for large scale computations is the object oriented framework - TopFEM [24]. TopFEM is a computational framework designed to perform finite element analysis using a topological data structure, called TopS [58]. TopS and TopFEM are written in C++ to take advantage of the Object Oriented Programming (OOP) paradigm; which includes features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance, making the codes extensible and reusable. Large codes can be reduced to smaller, manageable ones using OOP, increasing the maintainability of the codes. Finite element meshes are represented using TopS; which is a compact adjacency-based data structure. TopS provides efficient handling of data by reducing the required storage space; while preserving the ability to retrieve all the adjacency information in a time proportional to the number of retrieved entries. This efficiency is due to the fact that nodes and elements are the only entities stored explicitly, while other topological entities such as facets, edges and vertices are implicitly represented.

Parallel computing is another option for large-scale topology optimization. Parallel computing for topology optimization on traditional Central Processing Unit (CPU) architecture has been explored by Borrvall and Petersson [44], Kim et al. [120], Vemaganti et al. [220], Mahdavi et al. [137], Aage et al. [3] and Aage and Lazarov [2]. Schmidt and Schulz [176] and Zegard and Paulino [247] used Graphics Processor Unit (GPUs) for parallelization. An efficient approach to deal with large topology optimization problems is to split it into smaller problems. These smaller problems can either be solved serially and assembled later or parallel computing can be used to analyze the individual pieces concurrently.

In structural topology optimization, the solution of the state equation is typically the most computationally expensive step. Iterative solvers; such as Krylov subspace recycling [174, 230], are especially attractive in this regard. In topology optimization, since the system changes slowly over optimization steps, Wang et al. [230] were able to reduce the number of iterations and runtime by recycling the information from previous steps. Alternatively, Amir et al. [13] integrated the Combined Approximation method into topology optimization. They showed that approximate solutions of system equations produced comparable designs with significant savings in runtime. Later, Amir and Sigmund [14], utilized a single factorization/preconditioner for the entire optimization process and combined it with the Preconditioned Conjugate Gradient (PCG) method to solve the system approximately with fairly accurate designs. All these methods are encouraging and can potentially be used for topology optimization using polyhedrons.

Polyhedral multiresolution topology optimization (PolyMTOP3D)

For large-scale optimization, to obtain high resolution designs, approaches other than the ones indicated in the previous section are available in the literature. One such approach is the multiresolution topology optimization developed by Nguyen et al. [148]. In their approach for two-dimensional optimization, Nguyen et al. use three different levels of mesh discretization. A coarse finite element mesh is used to solve the elasticity problem and the optimization is conducted on a fine design variable mesh and element densities are represented on a finer mesh. Thus, this approach reduces the computational cost compared to the case when both finite element analysis and optimization are performed on a fine mesh.

Later, Nguyen et al. [149] made their MTOP approach more efficient by introducing adaptive refinement of design and density meshes.

The two-dimensional MTOP approach can naturally be extended to three-dimensions using the polyhedral topology optimization approach discussed in this thesis. Two possible three-dimensional polyhedral multiresolution topology optimization (PolyMTOP3D) approaches are illustrated in Figs. 7.1 and 7.2.

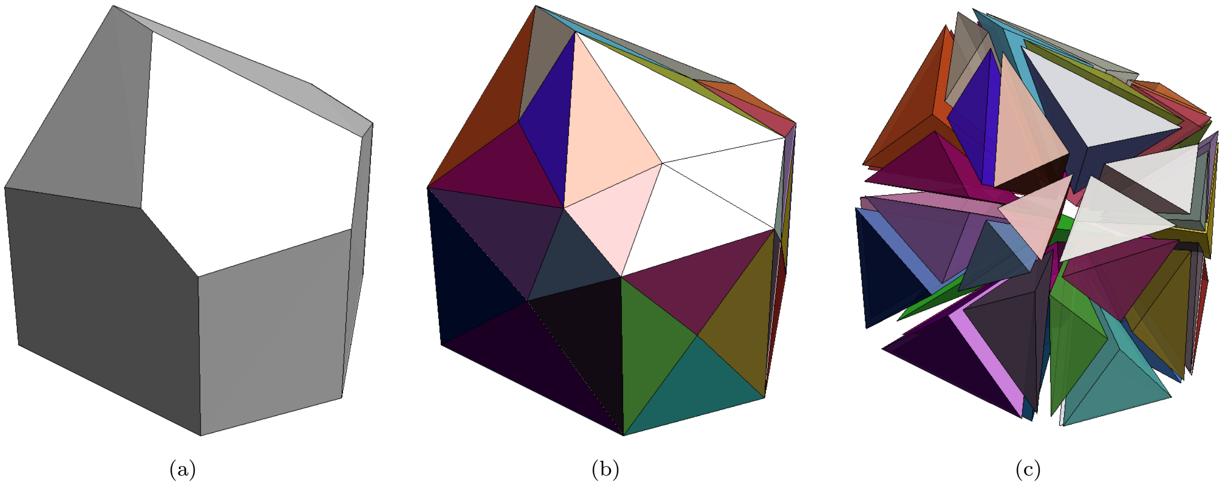

(a) (b) (c)

Figure 7.1: PolyMTOP3D using tetrahedral sub-elements as density elements. (a) Displacement element. (b) Tetrahedral density elements. (c) Split view of the density elements.

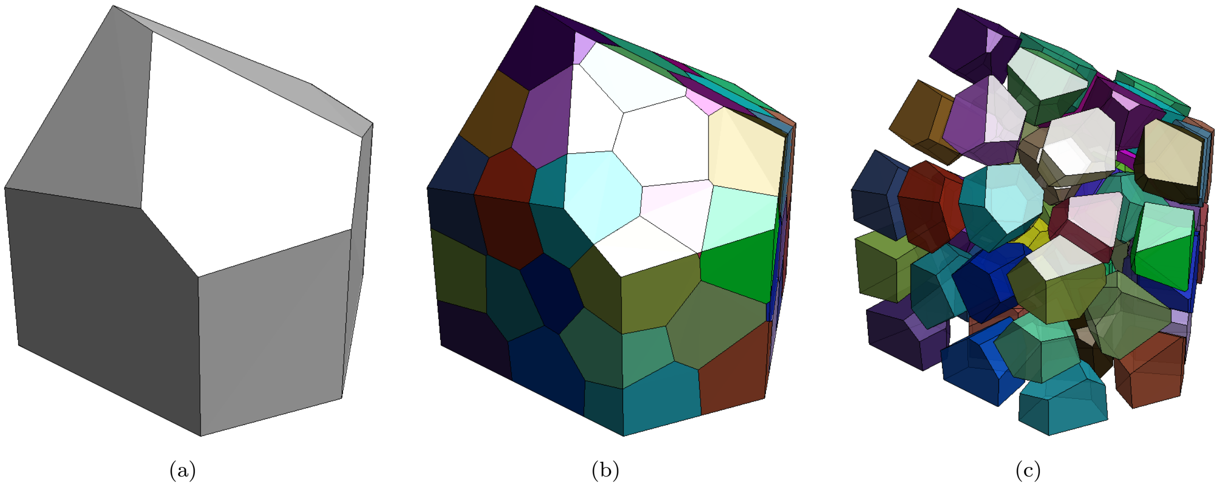

(a) (b) (c)

Figure 7.2: PolyMTOP3D using polyhedral sub-elements as density elements. (a) Displacement element. (b) Polyhedral density elements. (c) Split view of the density elements.

In the first approach, the finite element analysis is conducted on a polyhedral mesh (also referred to as the displacement mesh) and optimization is conducted on a tetrahedral mesh, obtained by subdividing the polyhedrons in the displacement mesh. Figure 7.1 shows one displacement polyhedral element with tetrahedral sub-elements as density elements. Another possible approach is to use polyhedrons as sub-divided density elements (Fig. 7.2). The main challenge in the implementation of PolyMTOP3D is to develop an efficient data structure which is needed to obtain element adjacency information. Perhaps the data structure TopS [58] could be used for this purpose. Capability to handle polyhedrons needs to be added to TopS.

Multi-objective topology optimization (MOTO)

In general, practical applications of topology optimization may require simultaneous optimization of multiple objective functions. In contrast to single-objective problems, multi-objective problems are more complex. For example, for minimization problems, the notion of minimum of a multiple functions is unclear. The features which minimize a particular objective may maximize, minimize or have no affect on other objective functions. To deal with multi-objective optimization, the concept of *Pareto optimality* [156] was introduced. It states that a point $\boldsymbol{x}^*$ is called a Pareto optimal point if there is no point $\boldsymbol{x}$ in the feasible design space that improves at least one objective function while keeping others unchanged. There are many multi-objective optimization methods available in the literature which are characterized by the nature of the solutions they provide. Some methods provide only Pareto optimal solutions but skip certain solution points in the Pareto optimal set. The weighted sum method is one such approach, in which the total objective is written as the weighted sum of individual objective functions with the weights chosen beforehand. By varying the weights systematically, different Pareto optimal points can be obtained. There are other methods which capture the entire Pareto optimal set but may also encompass non-Pareto optimal points, such as the weighted min-max method. Other multi-objective optimization methods are multi-objective genetic algorithms, weighted global criterion, goal programming and the lexicographic method [18]. Depending on the application, the relevant objective functions can be chosen for a multi-objective topology optimization. For example, for the conceptual design of craniofacial bone replacements, optimum load transfer and vascularization of the replacement are important considerations [203] which can be explored using polyhedral topology optimization on arbitrary non-Cartesian domains.

Other applications of polyhedral topology optimization

In this thesis, a computationally efficient topology optimization approach using polyhedrons is presented. The unstructured characteristic of the mesh alleviates any mesh bias in the designs. These features make the present approach attractive for applications in multi-physics problems [183, 187], photonic crystal structure design [113], phononic band gap material design [185], design of functionally graded materials [162, 194] and piezoelectric actuators [56]. In their work, Pereira et al. [166] present a topology optimization implementation for fluid flow problems in two-dimensions. They use polygonal elements to obtain a stable low-order discretization of the governing incompressible Stokes flow problem. In a separate work [212], the same authors show that their formulation satisfies the Babuska-Brezzi stability conditions for polygonal meshes such as Voronoi tessellations. The two-dimensional formulation of Pereira et al. [166] may conceivably be extended to three-dimensions using the approach discussed in this thesis.

The manufacturing industry could benefit from three-dimensional topology optimization using polyhedrons. There are two types of processes which are suitable for manufacturing topology optimization designs - subtractive manufacturing (cutting or drilling) and additive manufacturing (3D printing). Subtractive manufacturing refers to the removal of unwanted material to achieve a desired shape, for example, sculptures. Topology optimization has a similar design philosophy to subtractive manufacturing in terms of eliminating undesired material. Operations such as casting and milling have been included in the optimization process as manufacturing constraints [165, 177, 93]. In additive manufacturing or 3D printing, as the name suggests, a solid object is manufactured by successively laying down layers of material in different shapes. Additive manufacturing is advantageous because objects of almost any complexity can be manufactured using this technique. It has potential applications in the field of manufacturing scientific equipment [164], printing human organs [1], customizing food [69] and in chemical compounds [206]. The process starts with the creation of computer aided design (CAD) of the object and digital slices of cross-sections. The CAD files are then fed into a printing machine using STL file formats. The machine reads each cross-section data and lays down layers of powder or liquid which are automatically fused to create the final product. Although 3D printing can generate high print resolutions, sometimes an oversized version of the object is created and the unwanted material is removed using a high resolution subtractive process to obtain more precise designs. 3D printing is an attractive tool for rapid prototyping complicated polyhedrons-based three-dimensional topology optimization designs which might be difficult for subtracting manufacturing processes to handle. To generate 3D models, first the converged polyhedral topology optimization design needs to be post-processed and a corresponding CAD model needs to be generated.

Dynamic fracture simulations using polyhedral elements

A natural extension of the current work would be for dynamic problems such as dynamic crack propagation simulations. In fracture simulations, mesh orientation adversely affects the quality of computational solutions in simulations where crack propagation is confined to element interfaces. To alleviate mesh bias, techniques such as adaptive mesh refinement [157], nodal perturbation and edge-swap operators [161] have been used in the past. Mesh bias can also be reduced by using unstructured meshes such as Voronoi polygonal meshes [43, 197, 41, 128, 191]. Other special meshes such as pinwheel tilings have also been used [155, 163]. It should be noted that approaches such as mesh refinement, nodal perturbation and edge-swap often result in elements which are polygonal/polyhedral. Thus, use of numerical approaches based on polytopes not only reduces mesh bias naturally but can also handle the elements resulting from the bias reduction approaches discussed earlier. In this regard, the Mimetic Finite Difference-inspired approach discussed in this work can be beneficial for fracture simulations. For dynamic fracture simulation, other than the stiffness matrix, expressions for the mass and damping matrices need to be derived using the current approach. Other components such as the time integration schemes (for example Newmark algorithm [147]), cohesive zone models (for example PPR [158]) are well established.

Appendix A

Nomenclature

a, a_h	continuous and discrete bilinear form
ℓ, ℓ_h	continuous and discrete load linear form
$\boldsymbol{\varepsilon}$	linearized strain field
$\boldsymbol{b}$	applied body force
$\boldsymbol{t}$	surface loads
J	objective function
$\bar{J}$	effective objective function
$\mathbf{C}$	elasticity tensor
ω	admissible design for the optimization problem
Ω	working domain consisting of all admissible shapes ω
$\lvert\cdot\rvert$	measure (area or volume) of a set. Also, Euclidean norm of vector
$\mathbf{K}$	global stiffness matrix
$\mathbf{F}$	discretized global force vector
$\mathbf{U}$	discretized global displacement vector
E	Young's modulus
ν	Poisson's ratio
ρ, ρ_e	continuous density function and discrete element density
$\boldsymbol{V}$	array of the fractional areas/volumes of elements
V_f	prescribed volume fraction
$\phi^n_{i,j}$	design function, ϕ, level-set or phase-field, for nth iteration at location $\boldsymbol{x}_{i,j}$
$v_{i,j}$	Hamilton-Jacobi advection velocity, v, for nth iteration at location $\boldsymbol{x}_{i,j}$
r_{min}	filter radius
a^E, a^E_h	restriction of a, a_h to polyhedron E

$\boldsymbol{\mathcal{V}}_h$	approximate solution space spanned by smooth shape functions
$\boldsymbol{\mathcal{V}}_h^E$	restriction of $\boldsymbol{\mathcal{V}}_h$ to polyhedral element E
$\mathbb{P}_1(E)^n$	linear space
$\mathbb{P}_1^0(E)^n$	rigid body space
$\mathbb{P}_1^1(E)^n$	constant strain space
Π^E	projection operator to extract $\mathbb{P}_1(E)^n$
Π^0	projection operator to extract $\mathbb{P}_1^0(E)^n$
$\bar{\Pi}^E$	projection operator to extract $\mathbb{P}_1^1(E)^n$
$\mathbf{R}$	surface integration matrix
$\mathbf{N}$	nodal coordinate matrix
ϕ_i	Lagrange basis function
$\mathbf{P}_1$, $\mathbf{P}_M$	matrix representation of projection Π^0, $\bar{\Pi}^E$
K_1, K_2	consistency, stability terms of the stiffness matrix
α	scaling factor for the stability term

References

[1] Building body parts with 3D printing. The Engineer, May 2010.

[2] N. Aage and B. S. Lazarov. Parallel framework for topology optimization using the method of moving asymptotes. *Structural and Multidisciplinary Optimization*, 2013. Article in Press.

[3] N. Aage, T. H. Poulsen, A. Gersborg-Hansen, and O. Sigmund. Topology optimization of large scale Stokes flow problems. *Structural and Multidisciplinary Optimization*, 35(2):175–180, 2008.

[4] G. Allaire, F. de Gournay, F. Jouve, and A.-M. Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics*, 34(1):59–80, 2005b.

[5] G. Allaire and F. Jouve. A level-set method for vibration and multiple loads structural optimization. *Computer Methods in Applied Mechanics and Engineering*, 194:3269–3290, 2005a.

[6] G. Allaire and F. Jouve. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 32:909–918, 2008.

[7] G. Allaire, F. Jouve, and A.-M. Toader. A level set method for shape optimization. *Comptes Rendus Mathematique*, 12:1125–1130, 2002.

[8] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194:363–393, 2004.

[9] G. Allaire, A. Karrman, and G. Michailidis. Scilab code manual. `http://www.cmap.polytechnique.fr/~allaire/levelset_en.html`, January 2012.

[10] G. Allaire and O. Pantz. Structural optimization with freefem++. *Structural and Multidisciplinary Optimization*, 32(3):173–181, 2006.

[11] S. M. Allen and J. W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27:1085–1095, 1979.

[12] S. R. M. Almeida, G. H. Paulino, and E. C. N Silva. Layout and material gradation in topology optimization of functionally graded structures: A global-local approach. *Structural and Multidisciplinary Optimization*, 42(6):855–868, 2010.

[13] O. Amir, M. P. Bendsøe, and O. Sigmund. Approximate reanalysis in topology optimization. *International Journal for Numerical Methods in Engineering*, 78:1474–1491, 2009.

[14] O. Amir and O. Sigmund. On reducing computational effort in topology optimization: how far can we go? *Structural and Multidisciplinary Optimization*, 44:25–29, 2011.

[15] O. Amir, M. Stolpe, and O. Sigmund. Efficient use of iterative solvers in nested topology optimization. *Structural and Multidisciplinary Optimization*, 42(1):55–72, 2010.

[16] S. Amstutz and H. Andrä. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216:573–588, 2006.

[17] I. S. Aranson, V. A. Kalatsky, and V. M. Vinokur. Continuum field description of crack propagation. *Physics Review Letters*, 85(1):118–121, 2000.

[18] J. S. Arora. *Introduction to Optimum Design.* Elsevier Inc., 2012.

[19] M. Arroyo and M. Ortiz. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65(13):2167–2202, 2006.

[20] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[21] F. Aurenhammer and R. Klein. Voronoi diagrams. pages 201–290, 2000. Handbook of computational geometry, North-Holland, Amsterdam.

[22] I. Babuska and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.

[23] J. R. Barber. *Elasticity.* Springer, 3rd edition, 2010.

[24] L. L. Beghini, A. Pereira, R. Espinha, I. F. M. Menezes, W. Celes, and G. H. Paulino. An object-oriented framework for finite element analysis based on a compact topological data structure. Submitted for publication., 2013.

[25] L. Beirão Da Veiga. A mimetic discretization method for linear elasticity. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(2):231–250, 2010.

[26] L. Beirão Da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(1):199–214, 2013.

[27] L. Beirão Da Veiga, F. Brezzi, and L. D. Marini. Virtual Elements for linear elasticity problems. *SIAM Journal on Numerical Analysis*, 2013.

[28] L. Beirão Da Veiga, V. Gyrya, K. Lipnikov, and G. Manzini. Mimetic finite difference method for the Stokes problem on polygonal meshes. *Journal of Computational Physics*, 228(19):7215–7232, 2009.

[29] L. Beirão Da Veiga, K. Lipnikov, and G. Manzini. Convergence analysis of the high-order mimetic finite difference method. *Numerische Mathematik*, 113(3):325–356, 2009.

[30] L. Beirão Da Veiga, K. Lipnikov, and G. Manzini. Error analysis for a mimetic discretization of the steady stokes problem on polyhedral meshes. *SIAM Journal on Numerical Analysis*, 48(4):1419–1443, 2010.

[31] L. Beirão Da Veiga and G. Manzini. A higher-order formulation of the mimetic finite difference method. *SIAM Journal on Scientific Computing*, 31(1):732–760, 2008.

[32] V. V. Belikov, V. D. Ivanov, V. K. Kontorovich, S. A. Korytnik, and A. Yu. Semenov. The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics*, 37(1):9–15, 1997.

[33] T. Belytschko and J. M. Kennedy. Computer models for subassembly simulation. *Nuclear Engineering and Design*, 49(1-2):17–38, 1978.

[34] T. Belytschko, S. P. Xiao, and C. Parimi. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering*, 57:1177–1196, 2003.

[35] T. B. Belytschko, W. K. Liu, and J. M. Kennedy. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43(3):251–276, 1984.

[36] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1:193–202, 1989.

[37] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[38] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archives of Applied Mechanics*, 69(9-10):635–654, 1999.

[39] M. P. Bendsøe and O. Sigmund. *Topology Optimization - Theory, Methods and Applications.* Springer: New York, 2003.

[40] J. Bishop. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *International Journal for Numerical Methods in Engineering*, 2013. Accepted for publication.

[41] J.E. Bishop. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics*, 44(4):455–471, 2009.

[42] J. E. Bolander and S. Saito. Fracture analysis using spring networks with random geometry. *Engineering Fracture Mechanics*, 61:569–591, 1998.

[43] J. E. Bolander and N. Sukumar. Irregular lattice model for quasistatic crack propagation. *Physical Review B - Condensed Matter and Materials Physics*, 71(9), 2005. art. no. 094106.

[44] T. Borrvall and J. Petersson. Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6201–6229, 2001.

[45] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(8):2143–2158, 2001.

[46] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM - Control, Optimisation and Calculus of Variations*, 9:19–48, 2003.

[47] F. Brezzi, A. Buffa, and K. Lipnikov. Mimetic finite differences for elliptic problems. *Mathematical Modelling and Numerical Analysis*, 43(2):277–295, 2009.

[48] F. Brezzi, K. Lipnikov, M. Shashkov, and V. Simoncini. A new discretization methodology for diffusion problems on generalized polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40 SPEC. ISS.):3682–3692, 2007.

[49] F. Brezzi, K. Lipnikov, and V. Simoncini. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences*, 15(10):1533–1551, 2005.

[50] F. Brezzi and L. D. Marini. Virtual Element Methods for plate bending problems. *Computer Methods in Applied Mechanics and Engineering*, 253:455–462, 2013.

[51] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations.*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, New York, N.Y., 2004.

[52] M. Burger, B. Hackl, and W. Ring. Incorpotating topological derivatives into level set methods. *Journal of Computational Physics*, 194:344–362, 2004.

[53] M. Burger and R. Stainko. Phase-field relaxation of topology optimization with local stress constraints. *SIAM Journal on Control and Optimization*, 45(4):1447–1466, 2006.

[54] J. W. Cahn and J. E. Hillard. Free energy of a nonuniform system. I. Interfacial energy. *Journal of Chemical Physics*, 28:258–267, 1958.

[55] J. C. Campbell and M. J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. *Journal of Computational Physics*, 172(2):739–765, 2001.

[56] R. C. Carbonari, E. C. N. Silva, and Paulino. G. H. Topology optimization design of functionally graded bimorph-type piezoelectric actuators. *Smart Materials & Structures*, 16(6):2605–2620, 2007.

[57] J. Céa, S. Garreau, P. Guillaume, and M. Masmoudi. The shape and topological optimizations connection. *Computer Methods in Applied Mechanics and Engineering*, 188(4):713–726, 2000.

[58] W. Celes., G. H. Paulino, and R. Espinha. A compact adjacency-based topological data structure for finite element mesh representation. *International Journal for Numerical Methods in Engineering*, 64:1529–1556, 2005.

[59] V. J. Challis. A discrete level-set topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 41(3):453–464, 2010.

[60] V. J. Challis and J. K. Guest. Level-set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering*, 79(10):1284–1308, 2009.

[61] V. J. Challis, J. K. Guest, J. F. Grotowski, and A. P. Roberts. Computationally generated cross-property bounds for stiffness and fluid permeability using topology optimization. *International Journal of Solids and Structures*, 49(23-24):3397–3408, 2012.

[62] V. J. Challis, A. P. Roberts, and A. H. Wilkins. Design of three dimensional isotropic microstructures for maximized stiffness and conductivity. *International Journal of Solids and Structures*, 45(14-15):4130–4146, 2008.

[63] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.

[64] C. D. Chapman, K. Saitou, and M. J. Jakiela. Genetic algorithms as an approach to configuration and topology design. *Journal of Mechanical Design, Transactions Of the ASME*, 116(4):1005–1012, 1994.

[65] J.-S. Chen, M. Hillman, and M. Rüter. An arbitrary order variationally consistent integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418, 2013.

[66] J.-S. Chen, C.-T. Wu, and You Y. Yoon, S. Stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001.

[67] D. L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106:77–91, 1993.

[68] N. H. Christ, R. Friedberg, and Lee T. D. Weights of links and plaquettes in a random lattice. *Nuclear Physics B*, 210(3):337–346, 1982.

[69] D. L. Cohen, J. Lipton, M. Cutler, D. Coulter, A. Vesco, and H. Lipson. Hydrocolloid printing: A novel platform for customized food production. In *Solid Freeform Fabrication Symposium (SFF09)*, Austin, TX, USA., August 2009.

[70] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J Witt. *Concepts and applications of finite element analysis.* John Wiley and Sons, Inc, 4 edition, 2002.

[71] R. Courant, K. O. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[72] A. L. Cunha. *A fully Eulerian method for shape optimization with application to Navier-Stokes flows.* PhD thesis, Carnegie Mellon University, 2004.

[73] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 24th National Conference*, pages 157–172. ACM Press: New York, NY, U.S.A, 1969.

[74] M. J. De Ruiter and F. Van Keulen. Topology optimization using a topology description function. *Structural and Multidisciplinary Optimization*, 26(6):406–416, 2004.

[75] A. R. Diaz and N. Kikuchi. Solutions to shape and topology eigenvalue optimization problems using a homogenization method. *International Journal for Numerical Methods in Engineering*, 35(7):1487–1502, 1992.

[76] A. R. Diaz and O. Sigmund. Checkerboard patterns in layout optimization. *Structural and Multidisciplinary Optimization*, 10(1):40–45, 1995.

[77] Q. Du and D. Wang. The optimal centroidal Voronoi tessellations and the gersho's conjecture in the three-dimensional space. *Computers and Mathematics with Applications*, 49(9-10):1355–1373, 2005.

[78] X. B. Duan, Y. C. Ma, and R. Zhang. Shape-topology optimization for Navier-Stokes problem using variational level set method. *Journal of Computational and Applied Mathematics*, 222(2):487–499, 2008.

[79] P. Dvorak. New element lops time off CFD simulations. *Machine Desig*, 78(5):154–155, 2006.

[80] H. Eschenauer and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural Optimization*, 8(1):42–51, 1994.

[81] X. Feng and H. Wu. A posteriori error estimates for finite element approximations of the Cahn-Hilliard equation and the Hele-Shaw flow. *Journal of Computational Mathematics*, 26(6):767–796, 2008.

[82] D. P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17(5):679–706, 1981.

[83] C. Fleury and V. Braibant. Structural optimization: a new dual method using mixed variables. *International Journal for Numerical Methods in Engineering*, 23(3):409–428, 1986.

[84] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[85] M. S. Floater, K. Hormann, and G. Kòs. A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*, 24(1-4):311–331, 2004.

[86] M. S. Floater, G. Kòs, and M. Reimers. Mean value coordinates in 3D. *Computer Aided Geometric Design*, 22(7):623–631, 2005.

[87] A. L. Gain and G. H. Paulino. Phase-field based topology optimization with polygonal elements: A finite volume approach for the evolution equation. *Structural and Multidisciplinary Optimization*, 46(3):327–342, 2012.

[88] A. L. Gain and G. H. Paulino. BEAM - Bridging Engineering, Art and Mathematics. 2013. To be submitted for publication.

[89] A. L. Gain and G. H. Paulino. A critical comparative assessment of differential equation-driven methods for structural topology optimization. *Structural and Multidisciplinary Optimization*, 48(4):685–710, 2013.

[90] A. L. Gain, G. H. Paulino, D. Leonardo, and I. F. M. Menezes. Topology optimization using polytopes. 2013. Submitted.

[91] A. L. Gain, C. Talischi, and G. H. Paulino. On the virtual element method for three-dimensional elasticity problems on arbitrary polyhedral meshes. 2013. Submitted.

[92] S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for PDE systems: The elasticity case. *SIAM Journal on Control and Optimization*, 39(6):1756–1778, 2001.

[93] A. R. Gersborg and C. S. Andreasen. An explicit parameterization for casting constraints in gradient driven topology optimization. *Structural and Multidisciplinary Optimization*, 44(6):875–881, 2011.

[94] S. Ghosh. *Micromechanical Analysis and Multi-Scale Modelling using the Voronoi Cell Finite Element Method.* CRC Press, 2011.

[95] S. Ghosh and S. N. Mukhopadhyay. A two-dimensional automatic mesh generator for finite element analysis for random composites. *Computers and Structures*, 41(2):245–256, 1991.

[96] A. K. Gillette. *Stability of Dual Discretization Methods for Partial Differential Equations.* PhD thesis, University of Texas at Austin, 2011.

[97] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing, Reading, MA, 1989.

[98] J. K. Guest, J. H. Prevost, and T. Belytschko. 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, 2004.

[99] V. Gyrya and K. Lipnikov. High-order mimetic finite difference method for diffusion problems on polygonal meshes. *Journal of Computational Physics*, 227(20):8841–8854, 2008.

[100] S. H. Ha and S. Cho. Topological shape optimization of heat conduction problems using level set approach. *Numerical Heat Transfer, Part B: Fundamentals*, 48(1):67–88, 2005.

[101] E. Haber. A multilevel, level-set method for optimizing eigenvalues in shape design problems. *Journal of Computational Physics*, 198(2):518–534, 2004.

[102] L. He, C. Y. Kao, and S. Osher. Incorporating topological derivatives into shape derivatives based level set methods. *Journal of Computational Physics*, 225:891–909, 2007.

[103] W. S. Hemp. *Optimum Structures.* Clarendon Press, Oxford, 1973.

[104] A. Hirani. *Discrete Exterior Calculus.* PhD thesis, California Institute of Technology, 2003.

[105] H. Hiyoshi and K. Sugihara. Two generalizations of an interpolant based on Voronoi diagrams. *International Journal of Shape Modeling*, 5(2):219–231, 1999.

[106] H. S. Ho, M. Y. Wang, and M. Zhou. Parametric structural optimization with dynamic knot RBFs and partition of unity method. *Structural and Multidisciplinary Optimization*, 47(3):353–365, March 2013.

[107] J. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[108] K. Hormann and N. Sukumar. Maximum entropy coordinates for arbitrary polytopes. In *Eurographics Symposium on Geometry Processing*, volume 27, pages 1513–1520, 2008.

[109] K. Hormann and M. Tarini. A quadrilateral rendering primitive. In *Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 7–14, 2004.

[110] J. Hyman, M. Shashkov, and S. Steinberg. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. *Journal of Computational Physics*, 132(1):130–148, 1997.

[111] J. M. Hyman and M. Shashkov. Mimetic discretizations for Maxwell's equations. *Journal of Computational Physics*, 151(2):881–909, 1999.

[112] A. Iga, S. Nishiwaki, K. Izui, and M. Yoshimura. Topology optimization for thermal conductors considering design-dependent effects, including heat conduction and convection. *International Journal of Heat and Mass Transfer*, 52(11-12):2721–2732, 2009.

[113] J. S. Jensen and O. Sigmund. Topology optimization of photonic crystal structures: A high-bandwidth low-loss T-junction waveguide. *Journal of the Optical Society of America B: Optical Physics*, 22(6):1191–1198, 2005.

[114] P. Joshi, M. Meyer, T. Derose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics*, 26(3), 2007. Art. no. 1276466.

[115] C. Y. Kao, S. Osher, and E. Yablonovitch. Maximizing band gaps in two-dimensional photonic crystals by using level set methods. *Applied Physics B: Lasers and Optics*, 81(2):235–244, 2005.

[116] A. Kawamoto, T. Matsumori, S. Yamasaki, T. Nomura, T. Kondoh, and S. Nishiwaki. Heaviside projection based topology optimization by a PDE-filtered scalar function. *Structural and Multidisciplinary Optimization*, 44(1):19–24, 2011.

[117] S. W. Key. A finite element procedure for the large deformation dynamic response of axisymmetric solids. *Computer Methods in Applied Mechanics and Engineering*, 4(2):195–218, 1974.

[118] G. Kharmanda, N. Olhoff, A. Mohamed, and M. Lemaire. Reliability-based topology optimization. *Structural and Multidisciplinary Optimization*, 26(5):295–307, 2004.

[119] M. G. Kim, S. H. Ha, and S. Cho. Level set-based topological shape optimization of nonlinear heat conduction problems using topological derivatives. *Mechanics Based Design of Structures and Machines*, 37(4):550–582, 2009.

[120] T. S. Kim, J. E. Kim, and Y. Y. Kim. Parallelized structural topology optimization for eigenvalue problems. *International Journal of Solids and Structures*, 41(9-10):2623–2641, 2004.

[121] R. Kobayashi. Modeling and numerical simulations of dendritic crystal growth. *Physica D: Nonlinear Phenomena*, 63(3-4):410–423, 1993.

[122] I. Kosaka and C. C. Swan. A symmetry reduction method for continuum structural topology optimization. *Computers and Structures*, 70(1):47–61, 1999.

[123] D. Kosloff and G. A. Frazier. Treatment of hourglass patterns in low order finite element codes. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2(1):57–72, 1978.

[124] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, 2012. DOI 10.1007/s00158-012-0782-8.

[125] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalized shape optimization of fluids with lattice Boltzmann method. *International Journal for Numerical Methods in Fluids*, 65:496–519, 2011.

[126] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. http://control.ee.ethz.ch/ mpt/.

[127] D. Lavender, A. Bowyer, J. Davenport, A. Wallis, and J. Woodwark. Voronoi diagrams of set-theoretic solid models. *Computer Graphics and Applications*, 12(5):69–77, 1992.

[128] S. E. Leon, D. W. Spring, and G. H. Paulino. Reduction of mesh bias for dynamic fracture using adaptive splitting of polygonal finite elements. 2013. Submitted for review.

[129] K. Lipnikov, J. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes. *Journal of Computational Physics*, 199(2):589–597, 2004.

[130] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[131] Z. Lui, J. G. Korvink, and R. Huang. Structure topology optimization: fully coupled level set method via FEMLAB. *Structural and Multidisciplinary Optimization*, 29(6):407–417, 2005.

[132] J. Luo, Z. Luo, L. Chen, L. Tong, and M. Y. Wang. A semi-implicit level set method for structural shape and topology optimization. *Journal of Computational Physics*, 227(11):5561–5581, 2008a.

[133] J. Luo, Z. Luo, L. Chen, L. Tong, and M. Y. Wang. A new level set method for systematic design of hinge-free compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 198(2):318–331, 2008b.

[134] Z. Luo, L. Tong, M. Y. Wang, and S. Wang. Shape and topology optimization of compliant mechanisms using a parameterization level set method. *Journal of Computational Physics*, 227(1):680–705, 2007.

[135] J. N. Lyness and G. Monegato. Quadrature rules for regions having regular hexagonal symmetry. *SIAM Journal on Numerical Analysis*, 14(2):283–295, 1977.

[136] G. Maenchen and S. Sack. The TENSOR code. *Methods in Computational Physics*, 3:181–210, 1964.

[137] A Mahdavi, R. Balaji, M. Frecker, and E. M. Mockensturm. Topology optimization of 2D continua for minimum compliance using parallel computing. *Structural and Multidisciplinary Optimization*, 32(2):121–132, 2006.

[138] E. A. Malsch, J. J. Lin, and G. Dasgupta. Smooth two dimensional interpolants: a recipe for all polygons. *Journal of Graphics Tools*, 10:2, 2005.

[139] R. March. Visual reconstructions with discontinuities using variational methods. *Image and Vision Computing*, 10:30–38, 1992.

[140] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, and M. Gross. Polyhedral finite elements using harmonic basis functions. *Computer Graphics Forum*, 27(5):1521–1529, 2008.

[141] K. Maute and D. M. Frangopol. Reliability-based design of MEMS mechanisms by topology optimization. *Computers and Structures*, 81(8-11):813–824, 2003.

[142] K. Maute, S. Kreissl, D. Makhija, and R. Yang. Topology optimization of heat conduction in nano-composites. In *9th World Congress on Structural and Multidisciplinary Optimization*, Shizuoka, Japan, 2011.

[143] A. G. M. Michell. The limits of economy of material in frame-structures. *Philosophical Magazine*, 8(47):589–597, 1904.

[144] J. E. Morel, R. M. Roberts, and M. J. Shashkov. A local support-operators diffusion discretization scheme for quadrilateral r - z meshes. *Journal of Computational Physics*, 144(1):17–51, 1998.

[145] S. E. Mousavi, H. Xiao, and N. Sukumar. Generalized Gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering*, 82(1):99–113, 2010.

[146] S. Natarajan, S. Bordas, and D. R. Mahapatra. Numerical integration over arbitrary polygonal domains based on SchwarzChristoffel conformal mapping. *International Journal for Numerical Methods in Engineering*, 80(1):103–134, 2009.

[147] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.

[148] T. H. Nguyen, G. H. Paulino, J. Song, and C. H. Le. A computational paradigm for multiresolution topology optimization (MTOP). *Structural and Multidisciplinary Optimization*, 41(4):525–539, 2010.

[149] T. H. Nguyen, G. H. Paulino, J. Song, and C. H. Le. Improving multiresolution topology optimization via multiple discretizations. *International Journal for Numerical Methods in Engineering*, 92(6):507–530, 2012.

[150] T. H. Nguyen, J. Song, and G. H. Paulino. Single-loop system reliability-based topology optimization considering statistical dependence between limit-states. *Structural and Multidisciplinary Optimization*, 44(5):593–611, 2011.

[151] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams.* John Wiley, 2nd edition, 2000.

[152] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces.* Springer-Verlag, 2003.

[153] S. Osher and F. Santosa. 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, 2001.

[154] S. Osher and J. A. Sethian. Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[155] K.D. Papoulia, S.A. Vavasis, and P. Ganguly. Spatial convergence of crack nucleation using a cohesive finite-element model on a pinwheel-based mesh. *International Journal for Numerical Methods in Engineering*, 67(1):1–16, 2006.

[156] V. Pareto. *Manuale di Economia Politica Societa Editrice Libraria.* Milan, 1906.

[157] K. Park, G. H. Paulino, W. Celes, and R. Espinha. Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture. *International Journal for Numerical Methods in Engineering*, 92(1):1–35, 2012.

[158] K. Park, G. H. Paulino, and Roesler J. R. A unified potential-based cohesive model for mixed-mode fracture. *Journal of the Mechanics and Physics of Solids*, 57(6):891–908, 2009.

[159] G. H. Paulino, I. F. M. Menezes, M. Gattass, and S. Mukherjee. Node and element resequencing using the Laplacian of a finite element graph. Part I: General concepts and algorithm. *International Journal for Numerical Methods in Engineering*, 37(9):1994, 1994a.

[160] G. H. Paulino, I. F. M. Menezes, M. Gattass, and S. Mukherjee. Node and element resequencing using the Laplacian of a finite element graph. Part II: Implementation and numerical results. *International Journal for Numerical Methods in Engineering*, 37(9):1531–1555, 1994b.

[161] G. H. Paulino, K. Park, W. Celes, and R. Espinha. Adaptive dynamic cohesive fracture simulation using nodal perturbation and edge-swap operators. *International Journal for Numerical Methods in Engineering*, 84(11):1303–1343, 2010.

[162] G. H. Paulino and E. C. N Silva. Design of functionally graded structures using topology optimization. *Materials Science Forum*, 492-493:435–440, 2005.

[163] G.H. Paulino, K. Park, W. Celes, and R. Espinha. Adaptive dynamic cohesive fracture simulation using nodal perturbation and edge-swap operators. *International Journal for Numerical Methods in Engineering*, 84(11):1303–1343, 2010.

[164] J. M. Pearce. Building research equipment with free, open-source hardware. *Science*, 337(6100):1303–1304, 2012.

[165] C. B. W. Pedersen and P. Allinger. Industrial implementation and applications of topology optimization and future needs. In *IUTAM symposium on topological design optimization of structures, machines and materials: status and perspectives. Solid mechanics and its applications*, volume 137, pages 229–238. Springer, 2006. ISBN 1-4020-4729-00.

[166] A. Pereira, C. Talischi, G. H. Paulino, I. F. M. Menezes, and M. S. Carvalho. Implementation of fluid flow topology optimization in PolyTop. *Structural and Multidisciplinary Optimization*, 2013. Submitted.

[167] M. Peric. Flow simulation using control volumes of arbitrary polyhedral shape. *ERCOFTAC Bulletin*, 62, September 2004.

[168] P.-O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345, 2004.

[169] G. Pingen, M. Waidmann, A. Evgrafov, and K. Maute. A parametric level-set approach for topology optimization of flow domains. *Structural and Multidisciplinary Optimization*, 41(1):117–131, 2010.

[170] X. Qian and O. Sigmund. Topological design of electromechanical actuators with robustness toward over- and under-etching. *Computer Methods in Applied Mechanics and Engineering*, 253:237–251, 2013.

[171] G. I. N. Rozvany, O. M. Querin, Z. Gaspar, and V. Pomezanski. Weight-increasing effect of topology simplification. *Structural and Multidisciplinary Optimization*, 25(5-6):459–465, 2003.

[172] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural and Multidisciplinary Optimization*, 4(3-4):250–252, 1992.

[173] A. Russo. Virtual Element Methods II. In *Workshop on discretization methods for polygonal and polyhedral meshes.*, Milan, Italy, 2012. http://k.matapp.unimib.it/WSVEM-2012/index.shtml.

[174] Y. Saad. *Iterative methods for sparse linear systems.* SIAM, 2nd edition, 2003.

[175] D. Schattschneider. *M.C. Escher: Visions of Symmetry.* Harry N. Abrams, 2nd edition, 2004.

[176] S. Schmidt and V. Schulz. A 2589 line topology optimization code written for the graphics card. *Computing and Visualization in Science*, 14(6):249–256, 2011.

[177] U. Schramm and M. Zhou. Recent developments in the commercial implementation of topology optimization. In *IUTAM symposium on machines and materials: status and perspectives. Solid mechanics and its applications.*, volume 137, pages 239–248. Springer, 2006. ISBN 1-4020-4729-0.

[178] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999a.

[179] J. A. Sethian. *Level-Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics,Computer Vision and Materials Science.* Cambridge University Press, Cambridge, UK, 1999b.

[180] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163(2):489–528, 2000.

[181] M. Sharir and P. Agarwal. *Davenport-Schinzel sequences and their geometric applications.* Cambridge University Press, 1995.

[182] R. Sibson. A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151–155, 1980.

[183] O. Sigmund. Design of multiphysics actuators using topology optimization - Part II: Two-material structures. *Computer Methods in Applied Mechanics and Engineering*, 190(49-50):6605–6627, 2001.

[184] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica/Lixue Xuebao*, 25(2):227–239, 2009.

[185] O. Sigmund and J. S. Jensen. Systematic design of phononic band-gap materials and structures by topology optimization. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 361(1806):1001–1019, 2003.

[186] O. Sigmund and J. Peterson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1):68–75, 1998.

[187] E.C.N. Silva, R.C. Carbonari, and G.H. Paulino. On graded elements for multiphysics applications. *Smart Materials & Structures*, 16(6):2408–2428, 2007.

[188] D. Sohn, Y.-S. Cho, and S. Im. A novel scheme to generate meshes with hexahedral elements and poly-pyramid elements: The carving technique. *Computer Methods in Applied Mechanics and Engineering*, 201-204:208–227, 2012.

[189] J. Sokolowski and A. Zochowski. On the topological derivatives in shape optimization. *SIAM Journal on Control and Optimization*, 37:1251–1272, 1999.

[190] C. A. Soto and A. R. Diaz. Layout of plate structures for improved dynamic response using a homogenization method. *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, 65(1):667–674, 1993.

[191] D. W. Spring, S. E. Leon, and G. H. Paulino. Unstructured adaptive refinement on polygonal meshes for the numerical simulation of dynamic cohesive fracture. 2013. To be submitted for publication.

[192] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, 2001.

[193] L. L. Stromberg, A. Beghini, W. F. Baker, and G. H. Paulino. Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Structural and Multidisciplinary Optimization*, 43(2):165–180, 2011.

[194] F. V. Stump, E. C. N. Silva, and G. H. Paulino. Optimization of material distribution in functionally graded structures with stress constraints. *Communications in Numerical Methods in Engineering*, 23(6):535–551, 2007.

[195] V. Subramanian and J. B. Perot. Higher-order mimetic methods for unstructured meshes. *Journal of Computational Physics*, 219(1):68–85, 2006.

[196] N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal of Numerical Methods in Engineering*, 61(12):2159–2181, 2004.

[197] N. Sukumar and J. E. Bolander. Voronoi-based interpolants for fracture modelling. In *Tessellations in the Sciences: Virtues, Techniques and Applications of Geometric Tilings*. Springer Verlag, 2009.

[198] N. Sukumar and E. A. Malsch. Recent advances in the construction of polygonal finite element interpolations. *Archives of Computational Methods in Engineering*, 13(1):129–163, 2006.

[199] N. Sukumar, B. Moran, A. Y. Semenov, and V. V. Belikov. Natural neighbor Galerkin methods. *International Journal for Numerical Methods in Engineering*, 50:1–27, 2001.

[200] N. Sukumar and A. Tabarraei. Conforming polygonal finite elements. *International Journal of Numerical Methods in Engineering*, 61(12):2045–2066, 2004.

[201] Y. Sun and C. Beckermann. Sharp interface tracking using the phase-field equation. *Journal of Computational Physics*, 220(2):626–653, 2007.

[202] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[203] A. Sutradhar, G. H. Paulino, M. J. Miller, and T. H. Nguyen. Topology optimization for designing patient-specific large craniofacial segmental bone replacements. *Proceedings of the National Academy of Sciences*, 107(30):13222–13227, 2010.

[204] K. Suzuki and N. Kikuchi. A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 93(3):291–318, 1991.

[205] K. Svanberg. The method of moving asymptotoes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

[206] M. D. Symes, P. J. Kitson, J. Yan, C. J. Richmond, G. J. T. Cooper, R. W. Bowman, T. Vilbrandt, and L. Cronin. Integrated 3D-printed reactionware for chemical synthesis and analysis. *Nature Chemistry*, 4(5):349–354, 2012.

[207] K. Tai and T.H. Chee. Design of structures and compliant mechanisms by evolutionary optimization of morphological representations of topology. *Journal of Mechanical Design, Transactions of the ASME*, 122(4):560–566, 2000.

[208] A. Takezawa, S. Nishiwaki, and M. Kitamura. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 229:2697–2718, 2010.

[209] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 82:671–698, 2010.

[210] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. PolyMesher: A general-purpose mesh generator for polygonal elements written in Matlab. *Structural and Multidisciplinary Optimization*, 45(3):309–328, 2012.

[211] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Journal of Structural and Multidisciplinary Optimization*, 45(3):329–357, 2012.

[212] C. Talischi, A. Pereira, G. H. Paulino, I. F. M. Menezes, and M. S. Carvalho. Polygonal finite elements for incompressible flow. *International Journal for Numerical Methods in Engineering*, 2013. Accepted.

[213] S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity.* McGraw Hill, 3rd edition, 1970.

[214] R. Tsai and S. Osher. Level set methods and their application in image science. *Communications in Mathematical Sciences*, 1:623–656, 2003.

[215] N. P. Van Dijk. *Pushing the boundaries: Level-set methods and geometrical nonlinearities in structural topology optimization.* PhD thesis, Delft University of Technology, 2012b.

[216] N. P. Van Dijk, M. Langelaar, and F. van Keulen. A discrete formulation of a discrete level set method treating multiple contraints. In *8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, June 2009.

[217] N. P. Van Dijk, M. Langelaar, and F. Van Keulen. Explicit level-set based topology optimization using an exact Heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 91(1):67–97, 2012a.

[218] N. P. Van Dijk, K. Maute, M. Langelaar, and F. Van Keulen. Level-set methods for structural topology optimization: A review. *Structural and Multidisciplinary Optimization*, March 2013. DOI: 10.1007/s00158-013-0912-y.

[219] J. F. V. Vasconcellos and C. R. Maliska. A finite-volume method based on Voronoi discretization for fluid flow problems. *Numerical Heat Transfer, Part B*, 45:319–342, 2004.

[220] K. Vemaganti and W. E. Lawrence. Parallel methods for optimality criteria-based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 194(34-35):3637–3667, 2005.

[221] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

[222] E. L. Wachspress. *A rational finite element basis.* Academic Press, New York, 1975.

[223] E. L. Wachspress. Rational bases for convex polyhedra. *Computers and Mathematics with Applications*, 59(6):1953–1956, 2010.

[224] M. Wallin, M. Ristinmaa, and H. Askfelt. Optimal topologies derived from a phase-field method. *Structural and Multidisciplinary Optimization*, 45(2):171–183, 2012.

[225] M. Y. Wang. Topology optimization with level set method incorporating topological derivatives. In *6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 30 May - 3 June 2005.

[226] M. Y. Wang, X. Wang, and D. Guo. A level-set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192:227–246, 2003.

[227] M. Y. Wang and X. M. Wang. Color level sets: a multi-phase method for structural topology optimization with multiple materials. *Computer Methods in Applied Mechanics and Engineering*, 193:469–496, 2004.

[228] M. Y. Wang and S. Zhou. Phase field: A variational method for structural topology optimization. *Computer Modeling in Engineering & Sciences*, 6(6):547–566, 2004a.

[229] M. Y. Wang and S. Zhou. Synthesis of shape and topology of multi-material structures with a phase-field method. *Journal of Computer-Aided Materials Design*, 11:117–138, 2004b.

[230] S. Wang, E. de Sturler, and G. H. Paulino. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69:2441–2468, 2007.

[231] S. Wang and M. Y. Wang. Radial basis functions and level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65:2060–2090, 2006.

[232] S. Y. Wang, K. M. Lim, B. C. Khoo, and M. Y. Wang. An extended level set method for shape and topology optimization. *Journal of Computational Physics*, 221(1):395–421, 2007a.

[233] S. Y. Wang, K. M. Lim, B. C. Khoo, and M. Y. Wang. On hole nucleation in topology optimization using the level set methods. *Computer Modeling in Engineering and Sciences*, 21(3):219–237, 2007b.

[234] J. Warren. Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics*, 6(1):97–108, 1996.

[235] J. A. Warren, R. Kobayashi, A. E. Lobkovsky, and W. C. Carter. Extending phase field models of solidification to polycrystalline materials. *Acta Materialia*, 51(20):6035–6058, 2003.

[236] P. Wei and M. Y. Wang. Parametric structural shape and topology optimization method with radial basis functions and level-set mehod. In *Proceedings of International Design Engineering Technical Conferences & Computer and Information in Engineering Conference*, 2006.

[237] Q. Xia and M. Y. Wang. Topology optimization of thermoelastic structures using level set methods. *Computational Mechanics*, 42(6):837–857, 2008.

[238] Y. M. Xie and G. P. Steven. Shape and layout optimization via an evolutionary procedure. In *Proceedings of the international Conference on Computational Engineering Science.*, page 421, Hong Kong, 1992.

[239] T. Yamada, K. Izui, and S. Nishiwaki. A level set-based topology optimization method for maximizing thermal diffusivity in problems including design-dependent effects. *Journal of Mechanical Design*, 133:031011–1–031011–9, 2011.

[240] T. Yamada, K. Izui, S. Nishiwaki, and A. Takezawa. A topology optimization method based on the level set method incorporating a fictitious interface energy. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2876–2891, 2010.

[241] S. Yamasaki, S. Nishiwaki, T. Yamada, K. Izui, and M. Yoshimura. A structural optimization method based on the level set method using a new geometriy-based re-initialization scheme. *International Journal for Numerical Methods in Engineering*, 83(12):1580–1624, 2010.

[242] J. C. Ye, Y. Bresler, and P. Moulin. A self-referencing level-set method for image reconstruction from sparse Fourier samples. *International Journal of Computer Vision*, 50:253–270, 2002.

[243] M. Yip, J. Mohle, and J. E. Bolander. Automated modeling of three-dimensional structural components using irregular lattices. *Computer Aided Civil and Infrastructure Engineering*, 20(6):393–407, 2005.

[244] G. H. Yoon, Y. S. Joung, and Y. Y. Kim. Optimal layout design of three-dimensional genmetrically non-linear structures using the element connectivity parameterization method. *International Journal for Numerical Methods in Engineering*, 69:1278–1304, 2007.

[245] G. H. Yoon and Y. Y. Kim. Element connectivity parameterization for topology optimization of geometrically nonlinear structures. *International Journal of Solids and Structures*, 42:1983–2009, 2005.

[246] G. H. Yoon, Kim. Y. Y., M. Langelaar, and F. van Keulen. Theoretical aspects of internal element connectivity parameterization approach for topology optimization. *International Journal for Numerical Methods in Engineering*, 76:775–797, 2008.

[247] T. Zegard and G. H. Paulino. Toward GPU accelerated topology optimization on unstructured meshes. *Structural and Multidisciplinary Optimization*, 2013. Article in Press.

[248] S. Zhou and Q. Li. A variational level set method for the topology optimization of steady-state Navier-Stokes flow. *Journal of Computational Physics*, 227(24):10178–10195, 2008.

[249] C. G. Zhuang, Z. H. Xiong, and H. Ding. A level set method for topology optimization of heat conduction problem under multiple load cases. *Computer Methods in Applied Mechanics and Engineering*, 196(4-6):1074–1084, 2007.