Structural Topology Optimization Based on Level Set Method for Multidisciplinary Design

September 2009

Lee Chen Jian

A Thesis for the Degree of Ph.D. in Engineering

Structural Topology Optimization Based on Level Set Method for Multidisciplinary Design

September 2009

Graduate School of Science and Technology
Keio University

Lee Chen Jian

DISSERTATION

Submitted to the School of Science for Open and Environmental Systems, Keio University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

To my family and dear friends,

in memory of Professor Hirohisa Noguchi (1959∼2008)

Abstract

This dissertation contributes a novel concept for structural topology optimization for use in multidisciplinary design. The concept is to allow the structural optimization process to be influenced by specific designer inputs regarding topological properties and preferences, such as boundary curvature and location of voids. Although such inputs are specified, they are not strictly enforced as the proposed concept aims to find a balance between engineering and design objectives. Within the scope of this dissertation, engineering objectives are associated with metrics such as compliance and design objectives are associated with topological properties that add aesthetic and/or functional value to the structure. For example, design objectives can include aspects of architectural design.

There are many challenges to overcome in order to turn this concept into a reality. A technique to express these topological properties and their preferences has to be defined. Multi-objective methods have to be used to solve this multidisciplinary design problem. In addition, the proposed methods and techniques have to be fast.

Although there are many different approaches to topology optimization, level set based methods are recommended and heavily utilized in this dissertation for three main reasons:

1. Topological boundaries are very clearly defined by the contours of the level set function.

2. Local topological differences can be easily measured.

3. Designer inputs specifying topological properties can be created very naturally, and with relative ease, within the level set framework.

To allow the topology optimization process to be influenced by specified designer inputs, an objective function that measures deviation from those topological properties is proposed. This objective function suggests preference for specified topological properties such as boundary curvature and/or placement of voids in the structure. The specification for these preferences can be derived from different disciplines. For example, while boundary curvature can be linked to aesthetics, manufacturing constraints might require specific locations for voids.

Improving the efficiency of any optimization method is always a challenging task and for structural problems, Finite Element Method (FEM) computation is the bottleneck. In this dissertation, a high pass filter and an adaptive scheme for removing elements are proposed to tackle the issue of making the level set based topology optimization fast. The high pass filter increases time efficiency by reducing the number of iterations required to reach convergence. The introduced adaptive scheme takes a different approach and reduces the number of finite element equations in the system. A major advantage of this adaptive scheme is its generality; it can be applied to any level set based topology optimization method with relative ease and reduce the computational time required by as much as 70%.

The numerical examples shown in this dissertation substantiate the innovative use of the level set based methods and fast techniques described above to solve multidisciplinary structural topology optimization involving topological preferences. When aesthetics are considered an objective in structural topology optimization, the interpretation of the multidisciplinary optimization problem becomes one that searches for beauty and performance in topology.

Acknowledgements

Although it was only three years ago that I started my PhD, it has been an eventful short three years. This journey, in pursuit of a *philosophiædoctor*, has also become somewhat of a rite of passage for me. It was filled with happiness, excitement, gloom and heartache. It wasn't easy. But I had ample support and encouragement along the way.

I owe much to my current supervisor Professor Naoki Takano. For a student who entered his research lab very late in the academic year, he was very patient and tried his best to integrate me into his lab culture. Throughout the course of writing this dissertation, he provided excellent support. With all the associated administrative paperwork, it mustn't have been easy and I sincerely thank him for ensuring a smooth and relatively painless process. His research advice as well as advice on structuring this dissertation was much appreciated.

I would like to express my gratitude and heartfelt thanks to the committee members, Professor Akira Mita, Professor Yoshiyuki Matsuoka and Associate Professor Kenji Oguni who went through my dissertation, dissected it, put it under their microscope, prodded it and made me sweat under my shirt during my public defense. As a result of their contributions, I managed to add a final shine to this dissertation. I am also very grateful to Professor Akiko Matsuo for providing a listening ear whenever I was feeling down. She also provided facilities for me to print this dissertation.

I am indebted to Professor Hirohisa Noguchi (1959∼2008) for taking me under his wing. He is the reason I managed to study in Japan as well as develop this research. While

providing supervision, Prof. Noguchi was extremely patient and gave me a great deal of freedom to pursue my own research interests. I am also very grateful to him for sending me to many conferences both domestic and abroad. These trips have certainly widened my horizons academically as well as affected my overall outlook on life in a positive way. Some might say that I'm the first PhD graduate of Takano Lab (in Keio), but I also like to think of myself as the last PhD graduate of Noguchi Lab. I took for granted the time we had together and I miss him dearly.

A great source of inspiration and motivation for me throughout my time as a PhD candidate has been from my circle of PhD colleagues. Dr. Masato Tanaka, Dr. Zhang Zhiqian, Dr. Leonardo Patacchini, Dr. Gaku Hashimoto and Dr. Hiroshi Kawai, all of whom I spent a great deal of time with while in Noguchi Lab. I wish to thank them for the encouragement and support they provided throughout the duration of my PhD career.

Life is not all work and no play. There are numerous individuals who have contributed their smiles and provided a source of moral support along the way. I would like to thank my colleagues, friends and acquaintances that have kept me going with comfort and solace in times of hardship. When I was hungry, Alex was just a phone call away. Art kept me company online on numerous late nights spent writing this dissertation. Mamiko for providing a smile whenever one was needed. My wonderful juniors Wataru, Mari and Mizuki deserve a mention as they took good care of me and gave in to all my impossible demands.

The Japanese Government MEXT provided me with this excellent opportunity in pursuing my graduate studies at Keio University. Without their financial support, this research would not have been possible.

My deepest gratitude goes to my family who have always supported me in all my undertakings. Their love and faith in me throughout my life have empowered me to overcome the countless obstacles and challenges that have arisen over the years.

Last, but not least, to HE who watches over me.

Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In practical product design, there are many separate design teams involved. Two of these teams are creative design teams and engineering teams. The process is usually sequential with the creative design team laying out the topology and the engineering team working to realize it. In this process, communication is a one-way street. The engineering team is usually not allowed to alter, no matter how slight, any aspects put in place by the creative design team. However, sometimes a slight alteration might increase the robustness of the design dramatically. The question then becomes where to alter the design and by how much, such that the creative design team's intentions for the design are still visible.

This dissertation presents a novel concept for allowing the structural optimization process to be influenced by specific designer inputs regarding topological properties and preferences, such as boundary curvature and location of voids. Although such inputs are specified, they are not strictly enforced as the proposed concept is posed as a multidisciplinary problem that finds a balance between engineering and design objectives. Within the scope of this dissertation, engineering objectives are associated with metrics such as compliance and design objectives are associated with topological properties that add aesthetic and/or functional

value to the structure. For example, design objectives can also include aspects of architectural design.

There are many challenges to overcome in order to turn this concept into a reality. Multi-objective methods have to be used to solve this multidisciplinary problem. The concept of multi-objective optimization is a powerful tool which can be used to describe most, if not all, practical design and engineering problems. Such problems are fundamentally multi-objective in nature because of the existence of inherent tensions between objectives and expectations of design teams involved. A technique to express the above-mentioned topological properties and their preferences has to be defined. This has to be established in a straightforward manner that will facilitate compatibility when computer-aided design (CAD) programs are used to create those topological properties. Moreover, preferences have to be expressed in a style that is both simple and intuitive. Unlike single-objective problems with a single outcome, multi-objective problems deal with many possible outcomes and hence involves a great deal of computation. In addition to solving the problem, the proposed methods and techniques have to be optimized in terms of time efficiency.

There are various different formulations describing objective functions. When provided with a number of objectives, the most basic of numerical approaches to multi-objective optimization methods is to combine them with a scalar function by using aggregating weights and, depending on the problem, minimize or maximize this scalar function. A deterministic multi-objective method using a real valued function was developed to solve multi-objective optimization problems. Since most objective functions are not valued in the same scale, this real valued function involves a normalizing term. The proposed multi-objective method was first validated by solving various sizing and shape optimization problems. The solutions to these problems exhibit clear Pareto fonts, detailing the trade-offs between the various objectives.

Within sizing optimization problems, additional objective functions were introduced to reflect preference towards certain size values. Topology optimization in this dissertation is

performed using level set based methods. The level set framework provides a very natural description for topology that can be created with relative ease. To allow the topology optimization process to be influenced by specified designer inputs, an objective function that measures deviation from those topological properties is proposed. This objective function suggests preference for the specified topological properties; preferences for boundary curvature and/or placement of voids in the structure. Weighting factors used when combining the multiple objective functions determine how strictly these desired topological properties are adhered to in the final structure.

Multi-objective problems are computationally intensive. In multi-objective structural size and shape optimization, sensitivity analyses involving FEM amount to a bulk of the computational cost. For these problems, it was found that the real valued function and the adjoint variable method can be combined for efficient sensitivity analysis, especially when the number of design variables greatly outnumbers the number of objective functions. In order to improve the time efficiency of the level set method used in topology optimization, a high pass strain energy filter and an adaptive scheme for removing elements determined to be of void material by the level set are also proposed to reduce computational cost by as much as 70%. The high pass strain energy filter reduces computational cost by aggressively removing material to meet the volume constraint. On the other hand, the adaptive scheme proposed was aimed at reducing computational cost by reducing the number of degrees of freedom in the FEM equations.

The numerical examples shown in this dissertation substantiate the innovative use of the methods and techniques described herein to solve multidisciplinary structural optimization problems involving topological preferences. Preferences for boundary curvature and the location and size of voids can be adjusted by adjusting their respective weight values. The resulting Pareto fronts show trade-offs between maintaining topological preferences and minimizing structural compliances. When aesthetics are considered an objective in structural topology optimization, the interpretation of the multidisciplinary optimization problem

becomes one that searches for beauty and performance in topology.

1.2 Outline

This dissertation is organized into eight chapters. Chapter 2 provides a literature review of multi-objective optimization methods and level set based topology optimization methods. Chapter 3 provides various background information which serves as a prelude to the topics covered in later chapters. Chapter 4 introduces the concept of multi-objective optimization and how it will be extended to structural optimization involving size, shape and topology. It further details techniques and methods used to accomplish multi-objective optimization in these three areas. A novel multi-objective topology optimization problem involving topological preferences is also proposed towards the end of Chapter 4. Chapter 5 discusses and proposes some techniques and numerical schemes aimed at making the technology discussed in Chapter 4 more efficient. Chapter 6 presents numerical examples for sizing and shape optimization problems while Chapter 7 will present those related to topology optimization. Numerical examples pertaining to the multi-objective topology optimization problem proposed in Chapter 4 are also presented and their results discussed in Chapter 7. Chapter 8 closes this dissertation with a conclusion. Supplementary material regarding the selection of variables to normalize objective functions in the real valued function is also provided in Appendix A.

Chapter 2

Literature Review

2.1 Multi-objective optimization

Engineering design by default is a multi-objective problem solving procedure. It involves finding a favorable and suitable solution that best achieves and satisfies all design objectives and constraints. Within the design objectives, the presence of several conflicting design objectives is typical of engineering design problems. For most cases, optimization techniques are used to determine the most favorable design outcome, an optimal design. Traditional optimization techniques only consist of scalar methods. This means that their application only solves for one design objective at a time. The workaround to this limitation was to aggregate multiple objectives into a single objective function. The most basic of which, is called the weighted sum method [1]. This single objective function is then subjected to optimization with the result of one optimal design outcome. This result is however, strongly dependent on the aggregating methods used. Different elements used in the aggregating methods yield different optimal outcomes.

The prerequisite for decision making is the availability of choices[2, 3]. Therefore, the nature of a multi-objective problem prescribes it with a choice of various optimal solutions instead of a single optimal solution. The concept of Pareto optimality is used in order

to evaluate such solutions [1]. The term Pareto optimality is credited to Vilfredo Pareto (1848-1923), an economist who introduced the concept of nondominance in the context of economics. Since then, it has been applied to solve multi-objective problems. In a population of solutions, an individual solution is Pareto optimal if and only if there are no other solutions which best it in any one of its attributes. Due to the Pareto optimality of individual solutions, the outcome of a multi-objective optimization problem is a choice of solutions that represent the trade-offs between the various conflicting objectives. This trade-off is usually referred to as the Pareto front. As the development processes of complex engineering systems such as structures, aircrafts, vehicles etc. continue to improve and shorten in time, the consideration of multiple design objectives is necessary. The promise of multi-objective optimization methods is that they provide the required mathematical framework to numerically aid the development and understanding of such design problems and solving them[4, 5, 6, 7, 8] .

Kim and de Weck presented an adaptive weighted-sum method for problems with two objective functions. In their approach, the respective weights are adaptively determined and additional inequality constraints are specified in order to focus the search on unexplored regions in the solution space[9, 10]. The results they presented showed great promise as their method could produce well-distributed Pareto optimal solutions even in non-convex regions of the Pareto front.

A different approach to solving multi-objective problems by using evolutionary algorithms have been gaining popularity. Multi-objective evolutionary algorithms (MOEA) are considered robust[11] because of their ability to search for Pareto optimal solutions without the need for differentiable objective and constraint functions[12, 13]. However, frequently noted in recent literature is the fact that the convergence rate of such algorithms can be relatively slow. Furthermore, there is a limitation to the number of constraints that these algorithms can solve for.

Obayashi et al.[14] worked on using evolutionary algorithms to solve various multi-

objective problems relating to aircraft wing design. In [15], Obayashi showed that evolutionary algorithms are more robust and are more likely to find Pareto optimal solutions that lie on non-convex and discontinuous Pareto fronts than gradient-based methods. However they also did mention that gradient-based methods are more efficient and suggested that an initial search could be conducted with gradient-based methods and the results seeded into the initial population for evolutionary algorithms, in the hope that such a hybrid approach could inherit advantages from both methods.

Coello and Christiansen proposed using genetic algorithms (GA) to solve multi-objective truss optimization problems. They used the min-max optimum concept and formulated their GA in such a way that infeasible solutions determined by constraint conditions are never generated[16]. This approach was quite novel as most GAs at that time did not consider any constraint conditions.

A recent extension to multi-objective design optimization is multidisciplinary design optimization. Most current multi-objective design optimization problems are based on design objectives from a single discipline. This restricts the design to the component design level[1]. Multidisciplinary design concerns design endeavors which incorporate design objectives from a number of disciplines. To account for this extension in multi-objective optimization methods is not difficult as the key is simply to introduce additional design objectives from other disciplines. The crux is the formulation of the respective design objectives. Although the simultaneous consideration of design objectives from all relevant disciplines significantly raises the bar on the complexity of the design problem, the design from such a process is superior to that created from single discipline multi-objective optimization because it exploits the interactions between the respective disciplines. Furthermore, it allows the design process to be expanded from the component level to the system level[1].

2.2 Level set based topology optimization

Bendsoe and Sigmund[17] described topology optimization in structural design as the process of finding the most efficient placement of voids in a material domain. Topology optimization differs from sizing and shape optimization in the sense that it involves the determination of topological features such as the number, location and shape of voids as well as the connectivity of the structure in the domain. In sizing or shape optimization, some of the previously mentioned features are known a priori and are often kept constant throughout the optimization process.

The purpose of structural topology optimization is to find the optimal layout of a structure within a specified region. In a topology optimization problem, the physical size, shape and connectivity of the structure are the unknowns. The known quantities are the applied loading conditions, possible support conditions and permissible volume of the structure. Additional design requirements such as the location and size of prescribed voids or solid areas can also be considered.

In his pioneering work, Bendsoe introduced the density or Solid Isotropic Material with Penalization (SIMP) method[17]. The density method has since then been used to solve many topology optimization and homogenization problems. Although rather recent when compared to the density method, the level set method has been gaining popularity with solving topology optimization problems.

To summarize the main advantages and disadvantages of level set based methods, Table 2.1 compares them to the respective pros and cons of density-based methods. Although the level set method requires more effort to implement (calculating spatial derivatives, time derivatives, reinitialization etc), its solutions possesses smooth boundaries that clearly distinguishes between material and non-material domains. On the other hand, the density method has a simpler implementation but its solutions are pixelated and the distinction between material and non-material boundaries are not as evident as the level set method's solutions; a phenomenon referred to as grayscale, see Figure 2.1(a). Moreover, checkerboard solutions

Table 2.1: Comparing Topology Optimization Methods

Topology optimization		
	Density-based methods	**Level set based methods**
Pros	• Relatively easier to implement • Procedure is intuitive; layout optimization	• Clear boundaries; clear distinction between material and non-material domains; no grayscale • No checkerboarding
Cons	• Pixelated topology • Grayscale areas • Checkerboard solutions • High computational cost	• Slow to converge • Slightly more effort required to implement • Procedure involves the evolution of material boundaries
Notable aspects	• Variables: density of elements • Fine mesh = smooth boundaries, but higher computational cost	• Variables: level set function defined at the nodes • Smooth boundaries can also be obtained on a coarse grid
Implementing topological properties	• Density of elements describing said topological properties are fixed at either 1 (material) or 0 (non-material)	• Topological properties are initially set in and velocity of the nodes describing said topological properties are set to 0; no movement
Expressing topological preferences	• Difficult, maybe even impossible • Boundaries are not clear; grayscale areas • Difficult to extract data from CAD	• Boundaries are clear; clear distinction between material and non-material. • Extracting topological data from CAD is straightforward

illustrated in Figure 2.1(b) are also sometimes obtained. Specified topological properties, such as voids, can be implemented by simply fixing the respective "density" parameter or level set function.

(a) Grayscale (b) Checkerboarding (c) Clear boundaries

Figure 2.1: Density method and level set method examples

However, when it comes to expressing topological preferences, which is one of the proposals in this dissertation, the level set method's advantages outweigh its disadvantages as well as the advantages of density-based methods. Particularly, the clear distinction between material and non-material domains shown in Figure 2.1(c) give level set based methods a huge advantage over density methods. Furthermore, there exist straightforward methods to

create implicit level set functions from commercial CAD programs that possesses clear and distinct boundaries.

Unlike the density method, level set based methods do not have a "density" parameter for each grid cell which defines the material domain in a Cartesian grid. Level set based methods define material domains by embedding their boundaries in an implicit function one dimension higher than the material domain. For example, given a 2D topology, the corresponding level set function would be in three dimensions. Interpolation methods are usually used to determine the position of the boundary within a grid cell from the values of the level set function defined at the nodes of that particular grid cell.

The level set method was developed by American mathematicians Stanley Osher and James Sethian in the 1980s. Since then, it has become popular in many disciplines, such as image processing, computer graphics, computational geometry, optimization, and computational fluid dynamics. The foundation of the level set method was laid out in the works of Osher[18] and Sethian[19]. The level set method is an initial value problem. For boundary value problems, Sethian[19] proposed a fast marching method. The review here is restricted to level set implementations in optimization, particularly in the field of topology optimization.

Allaire et al.[20] used a combination of the shape derivative and the level set method to solve topology optimization problems. In their implementation, the shape derivative is computed by an adjoint method. Furthermore, the velocity of the level set is computed from sensitivity analysis. Nonlinear problems as well as design of compliant mechanisms were also considered in later works by Allaire et al.[21]. Lagrange multiplier is a fixed positive value. While Allaire et al.'s earlier works consisted of using compliance and least square error for a target displacement, recent works have also introduced new objective functions for vibration eigenvalues and multiple loads[22].

Amstutz and Andr[23] proposed an evolution equation for the level set function based on a generalization of the topological gradient concept. In their implementation, they abandoned

the Hamilton-Jacobi equation in favor of one based on the generalized topological gradient. By using the topological gradient, Amstutz and Andr have been able to create new "voids" in the material domain that was not possible when using the Hamilton-Jacobi equation. However, no real voids are created; soft materials are inserted to simulate voids. Unlike the Hamilton-Jacobi equation, this evolution equation does not require time integration. It uses a step size that is determined adaptively.

Belytschko et al.[24] proposed a structured extended finite element method for solids defined by implicit surfaces. In their work, they described how to create implicit functions to describe those solids from commercial CAD programs. In a subsequent research, Belytschko et al.[25] described using a constrained implicit function for topology optimization. The constrained implicit function is only defined within a narrow band about the surface. The implicit function outside the narrow band is constrained, resulting in two plateaus in the implicit function. Instead of using the Hamilton-Jacobi equation to update the implicit function, a heuristic updating scheme is used instead. Bisection is used to determine the Lagrange multiplier. Belytschko et al. states even though the implicit function can be defined by a signed distance function, no effort is made during the optimization process to keep it a signed distance function.

Chen et al.[26] proposed a method to achieve parametric shape optimization using level sets. In their approach, the level set function using B-splines and parameterized primitives combined with R-functions: B-splines are used to allow free-form deformations and R-functions are to support desired parametric changes. The shape optimization problem is then parameterized and becomes a sizing problem with two sets of parameters: coefficients of the B-spline basis functions and geometric dimensions. Their examples show that the method offers great flexibility as it provides explicit parametric control of geometry and topology.

Enright et al.[27] proposed new numerical methods for improving the mass conservation properties of the level set method when the interface is passively advected in a flow field.

Their method uses Lagrangian marker particles to rebuild the level set in regions which are under-resolved. The overall method maintains a smooth geometrical description of the interface and achieves mass conservation of the level set function.

Recently, He et al.[28] proposed using a level set method based on the combination of the shape derivative and the topological derivative. Their approach results in greater flexibility when it comes to shape changes and they claim that convergence to local optimal solutions can be avoided with their method. Norato et al.[29] also proposed working with the topological derivative when using level set methods since it also supports the reintroduction of solid material in void regions as well as introducing voids in the material domain.

Wang et al.[30, 31, 32, 33] proposed various level set methods for structural topology optimization. In a recent publication [33], they used Radial Basis Functions (RBF) multi-quadric splines to construct the implicit level set function. They claimed that this led to a high level of accuracy and smoothness when used to discretize the original initial value problem into an interpolation problem. The motion of the dynamic interfaces is thus governed by a system of coupled ordinary differential equations (ODEs) and a relatively smooth evolution can be maintained without reinitialization. In addition, it includes a mechanism to generate voids in the material domain. Numerical examples show that their approach is insensitive to initial designs and has a fast convergence rate.

Despite new advances in the level set topology optimization research field, the optimization problem is the same. It is usually a single-objective optimization problem formulated to maximize structural performance based on a metric such as compliance or strain energy density in view of certain constraints and conditions. Multi-objective topology optimization problems are rare and, at least in recent literature, there has been no work done in allowing the structural topology optimization process to be influenced by designer inputs.

Chapter 3

Background

3.1 Single-objective optimization

Single objective optimization problems are defined as determining j number of $\mathbf{b}$ design variables

$$\mathbf{b} = [b_1, ..., b_j]^T \tag{3.1}$$

such that a design objective

$$\mathbf{f} = [f_1]^T \tag{3.2}$$

is minimized subjected to equality and inequality constraints.

$$\mathbf{h}(\mathbf{b}) = 0 \tag{3.3}$$

$$\mathbf{g}(\mathbf{b}) \leq 0 \tag{3.4}$$

Single objective optimization methods search for a single design solution; design variables $\mathbf{b}$ that give the minimum $\mathbf{f}$. This minimum can be characterized into a local minimum or a global minimum. A local minimum is a value $f(\mathbf{b}')$ for which $f(\mathbf{b}') < f(\mathbf{b})$ holds for all $\mathbf{b}$ in a selected neighborhood. If $f(\mathbf{b}') < f(\mathbf{b})$ holds for all $\mathbf{b}$ in feasible space, then the value $f(\mathbf{b})$ is called a global minimum. An example is shown in Figure 3.1. The local minimum in Figure 3.1 is between $0 < f_1 < 2$ and its global minimum is seen to be between $-2 < f_1 < -1$.

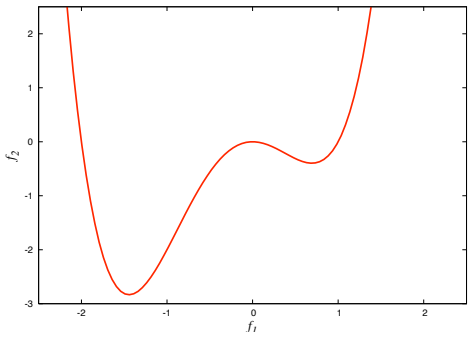


Figure 3.1: Global and local minima

The global minimum is associated with unconstrained problems. Local minima are mostly associated with constrained problems when constraints do not match the neighborhood of the global minimum.

A typical single optimization process follows through the flowchart shown in Figure 3.2. It begins with the definition of a design model. The creation of design variables **b** is followed by its single objective function evaluation. Design variables **b** are evaluated and updated until the solution has converged or reached a terminating criteria. Solution methods for single objective optimization problems are relatively easy due to the simplicity of its definition. Various literature such as [34] extensively goes through various solution methods for single objective optimization. Iterative methods such as gradient methods can be used effectively to solve problems with continuous objective functions. Problems with piecewise or discrete objective functions are usually more complicated and present a challenge, but nevertheless can be solved by applying some sort of treatment. A problem with single objective optimization is that premature convergence can occur in global minimization problems when the optimization process gets stuck at the local minimum.

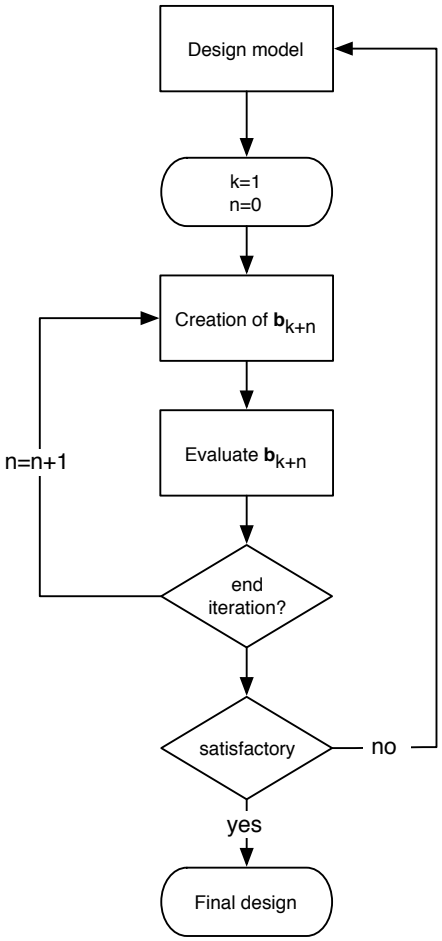


Figure 3.2: Typical optimization flowchart

3.2 Lagrange multipliers

Constraint handling is an important issue when dealing with optimization problems as practical problems possess at least one form of equality or inequality constraint, if not both. The definition of which are

$$\mathbf{h}(\mathbf{b}) = 0 \tag{3.5}$$

$$\mathbf{g}(\mathbf{b}) \leq 0 \tag{3.6}$$

respectively. Practical optimization methods have to be able to satisfy any equality or inequality constraint in the optimization process.

The Lagrange multiplier method[35] is a standard technique used in handling equality constraints. The Lagrangian function is defined as

$$L(\mathbf{x}, \boldsymbol{\lambda}) \equiv \mathbf{f}(\mathbf{b}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{b}) \tag{3.7}$$

where $\boldsymbol{\lambda}$ is a set of Lagrange multipliers for equality constraints. The conditions to achieve an optimal solution $[\mathbf{b}^*, \boldsymbol{\lambda}^*]$ are described by the Kuhn-Tucker conditions

$$\nabla_{\mathbf{b}} L(\mathbf{b}^*, \boldsymbol{\lambda}^*) = \nabla_{\mathbf{b}} \mathbf{f}(\mathbf{b}^*) + \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}^*)^T \boldsymbol{\lambda}^* = 0 \tag{3.8}$$

$$\nabla_{\boldsymbol{\lambda}} L(\mathbf{b}^*, \boldsymbol{\lambda}^*) = \mathbf{h}(\mathbf{b}^*) = 0. \tag{3.9}$$

Suppose the design variables and Lagrange Multipliers are updated iteratively by $\mathbf{b}_{k+1} = \mathbf{b}_k + \Delta\mathbf{b}_k$ and $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \Delta\boldsymbol{\lambda}_k$ respectively. It is now desirable to determine $[\Delta\mathbf{b}_k, \Delta\boldsymbol{\lambda}_k]$ such that the optimality conditions specified by the Kuhn-Tucker conditions are satisfied at $[\mathbf{b}_{k+1}, \boldsymbol{\lambda}_{k+1}]$. The resulting equation in matrix-vector form is:

$$\begin{bmatrix} \nabla_{\mathbf{bb}} L(\mathbf{b}_k, \boldsymbol{\lambda}_k) & \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k) \\ \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k)^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{b}_k \\ \boldsymbol{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{b}} \mathbf{f}(\mathbf{b}_k) \\ -\mathbf{h}(\mathbf{b}_k) \end{bmatrix} \tag{3.10}$$

An inversion of the matrix-vector Equation 3.10 results in equations for evaluating $[\Delta\mathbf{b}_k, \boldsymbol{\lambda}_{k+1}]$:

$$\begin{bmatrix} \Delta\mathbf{b}_k \\ \boldsymbol{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{bb}} L(\mathbf{x}_k, \boldsymbol{\lambda}_k) & \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k) \\ \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k)^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla_{\mathbf{b}} \mathbf{f}(\mathbf{b}_k) \\ -\mathbf{h}(\mathbf{b}_k) \end{bmatrix} \tag{3.11}$$

Note that $\nabla_{\mathbf{bb}} L(\mathbf{b}_k, \boldsymbol{\lambda}_k)$ is the Hessian of the Lagrange Equation 3.7.

To handle inequality constraints, a simple active sets strategy[34] has been proposed in addition to the Lagrange multiplier method for equality constraints[36]. With reference to Equation 3.7, the equality constraint $\mathbf{h}(\mathbf{b}_k)$ is replaced with the inequality constraint $\mathbf{g}(\mathbf{b}_k)$.

$$L(\mathbf{b}, \boldsymbol{\lambda}) \equiv \mathbf{f}(\mathbf{b}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{b}) \tag{3.12}$$

The active set strategy is

$$\begin{array}{ll} if\ (\mathbf{g}(\mathbf{b}_k) < 0) & \mathbf{g}(\mathbf{b}_k) = 0 \\ elseif\ (\mathbf{g}(\mathbf{b}_k) < 1) & \mathbf{g}(\mathbf{b}_k) = -\sqrt{\mathbf{g}(\mathbf{b}_k)} \\ else & \mathbf{g}(\mathbf{b}_k) = -\mathbf{g}(\mathbf{b}_k)^2 \end{array}$$

As long as the constraint is satisfied at $\mathbf{g}(\mathbf{b}_k) < 0$, there is no search in the direction to satisfy it. Constraints are added or deleted from the Lagrange multiplier equation when they are violated and when they are not. The proposed active set strategy does not treat inequality constraints as equality constraints and force movement along the constraint boundary, but rather search in areas where the inequality constraints are satisfied.

With both constraints involved, the Lagrangian function becomes

$$L(\mathbf{b}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \equiv \mathbf{f}(\mathbf{b}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{b}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{b}) \tag{3.13}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are both Lagrange multipliers for equality and inequality constraints respectively. Every iteration gives $[\Delta\mathbf{b}_k, \Delta\boldsymbol{\lambda}_{k+1}, \Delta\boldsymbol{\mu}_{k+1}]$ by

$$\begin{bmatrix} \Delta\mathbf{b}_k \\ \boldsymbol{\lambda}_{k+1} \\ \boldsymbol{\mu}_{k+1} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{bb}} L(\mathbf{b}_k, \lambda_k) & \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k) & \nabla_{\mathbf{b}} \mathbf{g}(\mathbf{b}_k) \\ \nabla_{\mathbf{b}} \mathbf{h}(\mathbf{b}_k)^T & 0 & 0 \\ \nabla_{\mathbf{b}} \mathbf{g}(\mathbf{b}_k)^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla_{\mathbf{b}} \mathbf{f}(\mathbf{b}_k) \\ -\mathbf{h}(\mathbf{b}_k) \\ -\mathbf{g}'(\mathbf{b}_k) \end{bmatrix} \tag{3.14}$$

where $\mathbf{g}'(\mathbf{b}_k)$ represents the value of the $\mathbf{g}(\mathbf{b}_k)$ returned by the active set strategy described earlier.

3.3 Level set methods

Level set methods are numerical schemes used to solve initial value problems in order to add dynamics to implicit functions [18]. The main concept behind level set methods is the Hamilton-Jacobi approach used to solve time-dependent equations. As a result, the numerical solutions to these time-dependent equations present a moving implicit surface.

3.3.1 Implicit description of topology

Assume that within a Cartesian domain Ω, to which all possible shapes are restricted on a Cartesian grid with points $\mathbf{x}$, a higher dimensional implicit function whose contours can be used to describe topology in Ω exists. Although any contour can be used, the common choice is the zeroth contour. The implicit function $\phi(\mathbf{x})$ is described as such,

$$\begin{aligned} \phi(\mathbf{x}) &= 0 \qquad & \mathbf{x} &\in \partial\Omega \\ \phi(\mathbf{x}) &> 0 \qquad & \mathbf{x} &\in \Omega^{+} \\ \phi(\mathbf{x}) &< 0 \qquad & \mathbf{x} &\in \Omega^{-} \end{aligned} \tag{3.15}$$

where the boundary, lack of and presence of voids is represented by $\partial\Omega$, Ω^{+} and Ω^{-} respectively. For the sake of simplicity, $\phi(\mathbf{x})$ is denoted by ϕ. Because of the way it uses its contours to describe topology, ϕ is usually referred to as the level set function. An example is shown in Figure 3.3 where Figures 3.3(a) and 3.3(b) show a level set function describing two voids in Ω.

3.3.2 Creating implicit functions

Although the level set function ϕ can be initialized with general functions[37], such as the radial basis function[25], a common practice is to initialize ϕ as a signed distance function [33, 38, 22]. This ensures that the function ϕ is always smooth enough to approximate its spatial derivatives[18].

(a) Higher dimensional implicit function (b) Boundaries, lack of and presence of voids in Ω

Figure 3.3: Embedding topology in a level set function

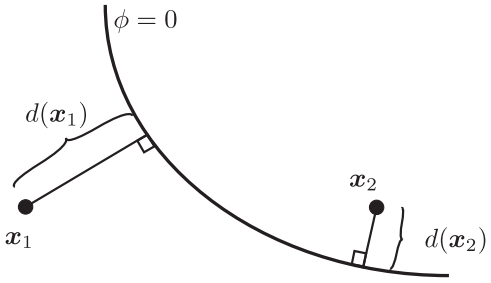


Figure 3.4: Distance function

Signed Distance Function

A distance function $d(\boldsymbol{x})$, where $\boldsymbol{x} \in \delta\Omega$, is defined as

$$d(\phi) = \min(|\boldsymbol{x} - \boldsymbol{x}_I|) \quad \text{for all} \quad \boldsymbol{x}_I \in \delta\Omega, \tag{3.16}$$

implies that $d(\boldsymbol{x}) = 0$ lies on an interface I. Geometrically, the distance function is defined as the shortest distance between a point in Ω and all points on the interface. An example is described visually in Figure 3.4 where two points and their corresponding distance function values are shown.

Also apparent from Figure 3.4 is that evaluating $-\nabla d$ at any point in Ω gives a vector that points in the direction of the interface or steepest descent. Furthermore, since d is Euclidean distance,

$$|\nabla d| = 1, \tag{3.17}$$

moving twice as close to the interface gives a value of d that is half as big [18].

Evaluating Equation (3.17) is possible if and only if there is a unique closest point on the interface with which $d(\boldsymbol{x})$ can be evaluated. Should a point $\boldsymbol{x}$ be equidistant from (at least) two distinct points on the interface, either distance value can be used. In that case, Equation 3.17 is only generally true since it does not hold at these points. However, this does not pose a problem numerically for the level set method since Equation (3.17) is generally only approximately satisfied.

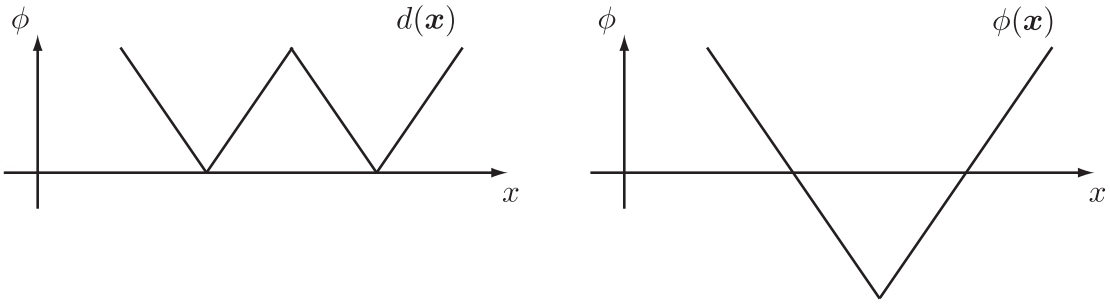
A signed distance function is an implicit function ϕ with $|\phi(\boldsymbol{x})| = d(\boldsymbol{x})$ for all $\boldsymbol{x}$. Thus, $\phi(\boldsymbol{x}) = d(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \delta\Omega$, $\phi(\boldsymbol{x}) = -d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^-$, and $\phi(\boldsymbol{x}) = d(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^+$. The description discussed earlier in Section 3.3.1 for describing topology implicitly can be easily implemented by using signed distance functions. In addition, there are a number of new properties that only signed distance functions possess. For example, as in Equation (3.17),

$$|\nabla\phi| = 1. \tag{3.18}$$

Unlike distance functions that have a kink at the interface where $d = 0$ is a minimum, signed distance functions are monotonic across the interface. Being monotonic enables signed distance functions to be differentiated on or near the interface with relative ease and significantly higher confidence. This is depicted in Figure 3.5. Figure 3.5(a) shows two kink at the interface when $\phi = 0$ whereas Figure 3.5(b) is smooth across the interface. Figure 3.6 shows an example of when a signed distance function is used to represent a 2D topology. The topology and its boundaries are clearly shown in the $x - y$ plane when we take the zeroth contour of the level set function.

3.3.3 Narrow banding

The narrow band approach was first introduced by Chopp[39] and analyzed extensively by Adalsteinsson and Sethian[40]. Instead of working on the whole Cartesian grid, referred to as the full matrix approach, the narrow band approach was proposed to work only near the



(a) Distance function (b) Signed distance function

Figure 3.5: Difference between a distance function and a signed distance function

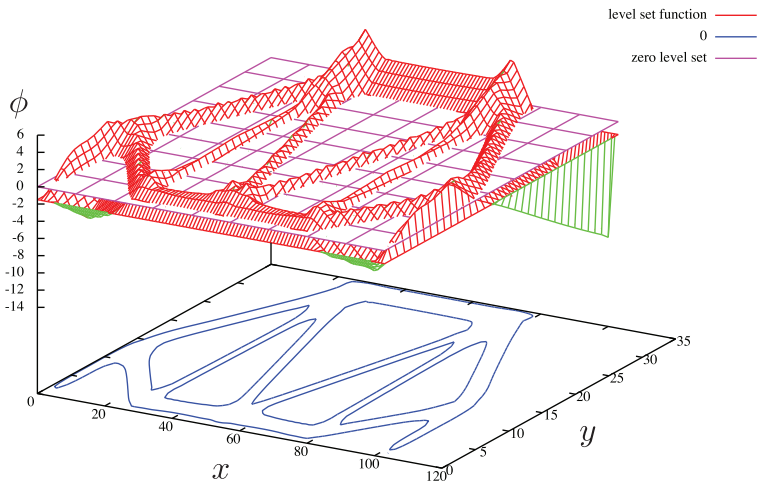


Figure 3.6: An example of a signed distance level set function

boundary of interest. The region near the boundary of interest is usually specified by a narrow band width of about six grid points on either side of the zero level set. The main reasons propelling the narrow band approach are:

- Speed: Calculations over the entire computational domain require $O(N^2)$ operations per time step in two dimensions and $O(N^3)$ operations pre time step in three dimensions, where N is the number of nodal points along each an axis. If we were to work within a neighborhood of the zero level set and assume that the boundary has approximately $O(N^2)$ nodes in three dimensions, the resulting operation count in 3D drops significantly to $O(kN^2)$, where k is the number of grid cells in the narrow band. Sethian[19] further states that a level set method utilizing the narrow band approach is typically ten times faster on a 160×160 grid than the full matrix approach.

- Time Steps: The maximum velocity is required in order to evaluate a time step that satisfies the CFL condition for each iteration. However, since movement of the boundary only occurs in the vicinity of the boundary, the maximum velocity can be adaptively chosen from within the narrow band to satisfy the CFL condition. This is advantageous when the velocity of the boundary changes substantially as in curvature flow.

3.3.4 Calculating additional quantities

Given an implicit function describing topology in Ω, calculating geometric properties, such as the length of the interface and the enclosed area, is straightforward. Calculating these geometric properties will make use of the Heaviside function $H(\phi)$,

$$H(\phi) = \begin{cases} 1 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases} \tag{3.19}$$

and its derivative, known as the dirac delta function $\delta(\phi)$.

An expression for the area A enclosed by Ω^+ makes use of $H(\phi)$ and is given by

$$A = \int_{\Omega} H(\phi) d\Omega. \tag{3.20}$$

The length L of the interface is similarly specified with

$$L = \int_{\Omega} \delta(\phi) |\nabla \phi| d\Omega. \tag{3.21}$$

Since both the Heaviside function and the delta function must be numerically approximated, regularized or smooth versions are usually used to smear their influence over a few grid cells. The regularized Heaviside function and delta delta function are shown in Equations (3.22) and (3.23) respectively.

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon, \\ \frac{1}{2} + \frac{3}{4}\left(\frac{\phi}{\epsilon} - \frac{\phi^3}{3\epsilon^3}\right) & \text{else if } -\epsilon \leq \phi \leq \epsilon, \\ 1 & \text{else,} \end{cases} \tag{3.22}$$

$$\delta(\phi) = \begin{cases} \frac{3}{4\epsilon}(1 - \frac{\phi^2}{\epsilon^2}) & -\epsilon < \phi < \epsilon, \\ 0 & \text{else,} \end{cases} \tag{3.23}$$

In the equations, the amount of smoothing can be tuned with ϵ. Figure 3.7 shows a portion of the interface when $\delta(\phi)$ is used to calculate length.

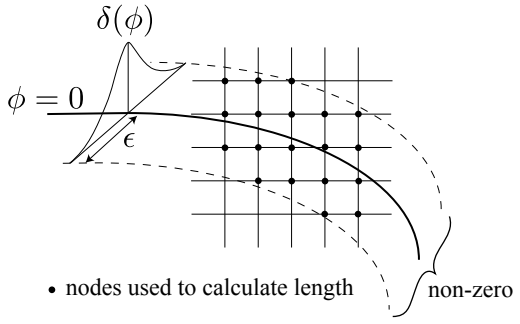


Figure 3.7: Calculating length of interface

The equations provided above only provide an approximate evaluation. When accuracy is a priority, the best approach is to simply find the interface, using any of standard contour plotters, and perform an accurate numerical quadrature. Finding the front explicitly usually only requires sweeping through the narrow band [19].

3.3.5 Hamilton-Jacobi equation

General Hamilton-Jacobi equations are of the form

$$\phi_t + H(\nabla\phi) = 0 \tag{3.24}$$

where H, not to be confused with the Heaviside function, can be a function of both space and time. Hamilton-Jacobi equations are dependent on the first derivative of ϕ which are hyberbolic in nature. Equation involving second order derivatives are parabolic and not considered to be of Hamilton-Jacobi type.

The level set equation shown in Equation (3.25) is an example of a Hamilton-Jacobi equation where $H(\nabla\phi) = N_n|\nabla\phi|$.

$$\frac{\partial\phi}{\partial t} + V_n|\nabla\phi| = 0 \tag{3.25}$$

Given an initial level set function $\phi(t = 0)$, the equation to move ϕ in time is given by

$$\phi_{t+\Delta t} = \phi_t - V_n|\nabla\phi|\Delta t \tag{3.26}$$

where Δt represents a small time increment.

3.3.6 Reinitialization

Reinitialization is still an active area of research and many other methods are still being proposed. However, the reinitialzation methods used in this thesis are restricted to those described here.

An important consideration when using level set functions is the reinitialization of the level set function [19]. In most cases, it is impossible to prevent ϕ from deviating away from a signed distance function because as the interface evolves, ϕ will generally drift away from its initialized signed distance property. Thus, a special technique termed reinitialization is applied periodically to ϕ in order to restore its signed distance property. Generally, the techniques presented here only restore ϕ to approximately signed distance, which is usually sufficient for most cases.

How often ϕ has to be reinitialized depends on the problem at hand. If the problem strictly requires ϕ to be a signed distance function at all times, reinitialization has to be performed with high-order accurate schemes and regularly.

The most straightforward way of implementing this is periodically stop the calculation and use a contour plotting algorithm to locate and discretize the $\phi = 0$ isocontour. Distances from the $\phi = 0$ can then be explicitly measured for every point in the Ω. Unfortunately, this kind of reinitialization routine can be very slow, especially if it needs to be done regularly. However, if the problem is not as sensitive to ϕ, reinitialization can be performed with faster lower-order schemes and only occasionally.

Even if the problem is not sensitive to the signed distance property of ϕ, it is still advisable to reinitialize ϕ from time to time as ϕ can develop noisy features and steep gradients that are not easily handled by finite difference approximations. For example, flat or steep regions may develop as the interface evolves, rendering computation of the normal vector, normal velocity and curvature at those places inaccurate [19]. Reinitializing occasionally ensures that ϕ stays smooth enough to approximate its spatial derivatives with some degree of accuracy.

A common approach to reinitialization that does not require the location of the interface explicitly involves solving a variant of the Hamilton-Jacobi equation to steady state.

$$\phi_t + S(\phi_0)(|\nabla| - 1) = 0 \tag{3.27}$$

This is termed the reinitialization equation [18], where $S(\phi_0)$ is a sign function taken as

$$S(\phi_0) = \begin{cases} 1 & \text{in } \Omega^+ \\ -1 & \text{in } \Omega^- \\ 0 & \text{in } \partial\Omega \end{cases} \tag{3.28}$$

Osher and Fedkiw [18] states that by using this equation, there is no need to initialize any points near the interface for use as boundary conditions. The points near the interface in Ω^+ use the points in Ω^- as boundary conditions, while the points in Ω^- conversely look at those in Ω^+.

As long as ϕ is relatively smooth and the initial data are somewhat balanced across the interface, this method works rather well. Conversely, this cannot be said when ϕ is not smooth or ϕ is much steeper on one side of the interface than the other as circular dependencies on initial data can cause the interface to move incorrectly from its initial starting position. This is the reason why $S(\phi_0)$ is defined using the initial values of ϕ (denoted by ϕ_0) so that the domain of dependence does not change if the interface incorrectly crosses over a grid point.

In discretizing Equation (3.27), the $S(\phi_0)|\nabla\phi|$ term is treated as motion in the normal direction. Here $S(\phi_0)$ is constant for all time and can be thought of as a velocity term. Numerical tests indicate that better results are obtained when $S(\phi_0)$ is numerically smeared out with:

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + (\Delta x)^2}} \tag{3.29}$$

Osher and Fedkiw [18] also pointed out that other researches suggested

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + |\nabla\phi|^2(\Delta x)^2}} \tag{3.30}$$

was a better choice, especially when the initial ϕ_0 was a poor estimate of the signed distance when $|\nabla\phi_0|$ was far from 1. In Equation (3.30), it is important to update $S(\phi_0)$ continually as the calculation progresses so that the $|\nabla\phi|$ term has the intended effect. In contrast,

Equation (3.29) is evaluated only once using the initial data. Numerical smearing of the sign function decreases its magnitude, slowing the propagation speed of information near the interface. Osher and Fedkiw [18] recommends using Godunov's method for discretizing the hyperbolic $S(\phi_0)|\nabla\phi|$ term. After finding a numerical approximation to $S(\phi_0)|\nabla\phi|$, combine it with the remaining $S(\phi_0)$ source term at each grid point and update the resulting quantity in time with a Runge-Kutta method.

Ideally, the interface remains stationary during the reinitialization procedure, but numerical errors will tend to move it to some degree. Sussman and Fatemi [41] suggested imposing a local constraint individually in each grid cell. They approached the reinitialization problem from a area-preserving perspective. In their argument, if the interface does not move during reinitialization, the area is preserved. Therefore it is reasonable that one can preserve the area while allowing the interface to move as well. Instead of using the exact area, the authors used Equation (3.20) to compute the volume in each grid cell. A constraint is applied locally in each grid cell by the addition of a correction term to the right-hand side of Equation 3.27

$$\phi_t + S(\phi_0)(|\nabla\phi - 1|) = \lambda\delta(\phi)|\nabla\phi|. \tag{3.31}$$

The constraint that the area in each cell does not change, i.e., $(A_\Omega)_t = 0$, is equivalent to

$$\int_\Omega H'(\phi)\phi_t d\Omega = 0 \tag{3.32}$$

or

$$\int_\Omega \delta(\phi)(-S(\phi_0)(|\nabla\phi| - 1) + \lambda\delta(\phi)|\nabla\phi|)d\Omega = 0 \tag{3.33}$$

using Equation (3.31) and the fact that the derivative of the Heaviside function is the dirac delta function. A separate λ is defined in each grid cell using Equation (3.33) to obtain:

$$\lambda = -\frac{\int_\Omega \delta(\phi)(-S(\phi)(|\nabla\phi| - 1))d\Omega}{\int_\Omega \delta^2(\phi)|\nabla\phi|d\Omega} \tag{3.34}$$

or

$$\lambda = -\frac{\int_\Omega \delta(\phi)\left(\frac{\phi^{n+1}-\phi^n}{\Delta t}\right)d\Omega}{\int_\Omega \delta^2(\phi)|\nabla\phi|d\Omega} \tag{3.35}$$

where Equation (3.27) is used to compute ϕ^{n+1} from ϕ^n. In summary, Equation (3.27) is used to update ϕ^n in time. Then Equation (3.35) is used to compute a λ for each grid cell. Finally, the initial guess for ϕ^{n+1} obtained from Equation (3.27) is replaced with a corrected $\phi^{n+1} + \Delta t \lambda \delta(\phi)|\nabla\phi|$. It is shown in [41] that this specific discretization exactly cancels out a first order error term in the pervious formulation This is similar to the gradient projection method which is also used to evaluate volume constraints for topology optimization described in later sections.

3.4 Spatial Discretization

The finite difference techniques described in this section can be used for approximating spatial derivatives on a Cartesian grid.

First-order accurate forward difference

$$\frac{\partial\phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{3.36}$$

abbreviated as $D^+\phi$, a first-order accurate backward difference

$$\frac{\partial\phi}{\partial x} \approx \frac{\phi_i - \phi_{i-1}}{\Delta x} \tag{3.37}$$

abbreviated as $D^-\phi$, or a second-order accurate central difference

$$\frac{\partial\phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \tag{3.38}$$

abbreviated as $D^o\phi$.

3.4.1 First order discretization

Although the equations for first-order differences are given above, they cannot be used in a practical sense. They are usually used in an upwind scheme where the selection of D^+, D^- or D^o for approximating spatial derivatives is determined based on the sign of the wave

velocity V_n; D^+ or D^- is selected in the direction where the characteristic information is coming from. A basic upwind scheme is listed below.

$$\phi_{ijk}^{n+1} = \phi_{ijk}^{n} - \Delta t[\max(F_{ijk}, 0)\nabla^+ + \min(F_{ijk}, 0)\nabla^-] \tag{3.39}$$

where

$$\begin{aligned} \nabla^+ &= [\max(D_{ijk}^{-x}, 0)^2 + \min(D_{ijk}^{+x}, 0)^2 + & (3.40)\\ &\quad \max(D_{ijk}^{-y}, 0)^2 + \min(D_{ijk}^{+y}, 0)^2 + & (3.41)\\ &\quad \max(D_{ijk}^{-z}, 0)^2 + \min(D_{ijk}^{+z}, 0)^2]^{\frac{1}{2}} & (3.42) \end{aligned}$$

$$\begin{aligned} \nabla^- &= [\max(D_{ijk}^{+x}, 0)^2 + \min(D_{ijk}^{-x}, 0)^2 + & (3.43)\\ &\quad \max(D_{ijk}^{+y}, 0)^2 + \min(D_{ijk}^{-y}, 0)^2 + & (3.44)\\ &\quad \max(D_{ijk}^{+z}, 0)^2 + \min(D_{ijk}^{-z}, 0)^2]^{\frac{1}{2}} & (3.45) \end{aligned}$$

Shorthand notation in which $D^{+x}\phi_i^n$ is written as D_i^{+x}.

3.4.2 Second order discretization

If higher accuracy is required, the above schemes can be extended. The basic trick is to build a switch that turns itself off whenever a shock is detected; otherwise, it will use a higher order approximation to the left and right values by means of a higher order polynomial using an Essentially Non-Oscillatory ENO construction [19, 18]. These details will not be presented here. The scheme is given by

$$\begin{aligned} \nabla^+ &= [\max(A, 0)^2 + \min(B, 0)^2 + & (3.46)\\ &\quad \max(C, 0)^2 + \min(D, 0)^2 + & (3.47)\\ &\quad \max(E, 0)^2 + \min(F, 0)^2]^{\frac{1}{2}} & (3.48) \end{aligned}$$

$$\nabla^- = [\max(B,0)^2 + \min(A,0)^2 + \tag{3.49}$$

$$\max(D,0)^2 + \min(C,0)^2 + \tag{3.50}$$

$$\max(F,0)^2 + \min(E,0)^2]^{\frac{1}{2}} \tag{3.51}$$

where

$$A = D^{-x}_{ijk} + \frac{\Delta x}{2} m(D^{-x-x}_{ijk}, D^{+x-x}_{ijk}) \tag{3.52}$$

$$B = D^{+x}_{ijk} - \frac{\Delta x}{2} m(D^{+x+x}_{ijk}, D^{+x-x}_{ijk}) \tag{3.53}$$

$$C = D^{-y}_{ijk} + \frac{\Delta y}{2} m(D^{-y-y}_{ijk}, D^{+y-y}_{ijk}) \tag{3.54}$$

$$D = D^{+y}_{ijk} - \frac{\Delta y}{2} m(D^{+y+y}_{ijk}, D^{+y-y}_{ijk}) \tag{3.55}$$

$$E = D^{-z}_{ijk} + \frac{\Delta z}{2} m(D^{-z-z}_{ijk}, D^{+z-z}_{ijk}) \tag{3.56}$$

$$F = D^{+z}_{ijk} - \frac{\Delta z}{2} m(D^{+z+z}_{ijk}, D^{+z-z}_{ijk}) \tag{3.57}$$

$$\tag{3.58}$$

and the switch function is given by

$$m(x,y) = \begin{cases} \begin{cases} x & \text{if } |x| \le |y| \\ y & \text{if } |x| > |y| \end{cases} & xy \ge 0 \\ 0 & xy < 0 \end{cases}. \tag{3.59}$$

Note that these schemes are explicit in time and hence can be programmed in a straightforward manner.

3.5 Temporal discretization

While the methods above enable us to approximate spatial derivatives, we would still need some form of temporal discretization if we are to evolve the level set equation shown in Equation (3.25) forward in time. This section will detail two popular standard schemes, the Forward Euler and the TVD Runge-Kutta methods.

3.5.1 Forward Euler

Once ϕ and V_n are defined at every grid point on the Cartesian grid, numerical methods to evolve ϕ forward in time can be applied, effectively moving the interface across the grid. Updating ϕ in time consists of finding new values of ϕ at every grid point after some time increment Δt. New values of ϕ are denoted by $\phi^{n+1} = \phi(t^{n+1})$, where $t^{n+1} = t^n + \Delta t$.

A rather simple first-order accurate method for the time discretization of Equation 3.25 is the forward Euler method given by,

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + V_n|\nabla\phi| = 0 \tag{3.60}$$

where V_n is the given external velocity field at time t^n, and $\nabla\phi^n$ represents the gradient using values of ϕ at time t^n.

3.5.2 TVD Runge Kutta

The Total Variation Diminishing (TVD) Runge-Kutta (RK) method can be used when a higher-order temporal discretization is necessary in order to obtain accurate numerical solutions. While there are numerous RK schemes, this TVD RK schemes guarantee that no spurious oscillations are produced as a consequence of the higher-order accurate temporal discretization as long as no spurious oscillations are produced with the forward Euler building block described earlier [18].

The first-order accurate TVD RK scheme is simply the forward Euler method. Higher-oder accurate methods are obtained by sequentially taking Euler steps and combining the results with the initial data using a convex combination.

The second-order accurate TVD RK scheme is identical to the standard second-order accurate RK scheme. It is also known as the midpoint rule, as the modified Euler method, and as Heun's predicor-corrector method. The steps are listed below. First, an Euler step is taken to advance the solution to time $t^n + \Delta t$,

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + V_n|\nabla\phi|^n = 0, \tag{3.61}$$

followed by a second Euler step to advance the solution to time $t^n + 2\Delta t$

$$\frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + V_{n+1}|\nabla\phi|^{n+1} = 0, \tag{3.62}$$

followed by an averaging step

$$\phi^{n+1} = \frac{1}{2}\phi^n + \frac{1}{2}\phi^{n+2} \tag{3.63}$$

that takes a convex combination of the initial data and the result of two Euler steps. The final averaging step produces the second-order accurate TVD approximation to ϕ at time $t^n + \Delta t$.

The third-order TVD RK scheme is as follows. First, an Euler step is taken to advance the solution to time $t^n + \Delta t$,

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + v^n \cdot \nabla\phi^n = 0 \tag{3.64}$$

followed by a second Euler step to advance the solution to time $t^2 + 2\Delta t$,

$$\frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + v^{n+1} \cdot \nabla\phi^{n+1} = 0 \tag{3.65}$$

followed by an averaging step

$$\phi_{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{n+2} \tag{3.66}$$

that produces an approximation to ϕ at time $t^n + \frac{1}{2}\Delta t$. Then another Euler step is taken to advance the solution to time $t^n + \frac{3}{2}\Delta t$,

$$\frac{\phi^{n+\frac{3}{2}} - \phi^{n+\frac{1}{2}}}{\Delta t} + v^{n+\frac{1}{2}} \cdot \nabla\phi^{n+\frac{1}{2}} = 0 \tag{3.67}$$

followed by a second averaging step

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{n+\frac{3}{2}} \tag{3.68}$$

that produces a third-order accurate approximation to ϕ at time $t^n + \Delta t$. This third-order accurate TVD RK method has a stability region that includes part of the imaginary axis.

3.5.3 Stability and CFL condition

Stability becomes an issue when a finite difference approximation is used. It guarantees that small errors in the approximation are not amplified as the solution is marched forward in time. The Courant-Friedreichs-Lewy condition (CFL condition) can be used to enforce stability. The CFL condition states that the numerical waves should propagate at least as fast as the physical waves. This means that the numerical wave speed of $\frac{\Delta x}{\Delta t}$ must be at least as fast as the physical wave speed $|u|$, i.e., $\frac{\Delta x}{\Delta t} > |u|$. This leads us to the CFL time step restriction of

$$\Delta t < \frac{\Delta x}{\max|u|} \tag{3.69}$$

where $\max|u|$ is chosen to be the largest value of $|u|$ over the entire cartesian grid.

Equation (3.69) is usually enforced by choosing a CFL number α with

$$\Delta t \left(\frac{\max|u|}{\Delta x} \right) = \alpha \tag{3.70}$$

and $0 < \alpha < 1$. A common near-optimal choice is $\alpha = 0.9$, and a common conservative choice is $\alpha = 0.5$. A multidimensional CFL condition can be written as

$$\Delta t \max \left\{ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right\} = \alpha, \tag{3.71}$$

although

$$\Delta t \left(\frac{\max|\boldsymbol{v}|}{\min\{\Delta x, \Delta y, \Delta z\}} \right) = \alpha \tag{3.72}$$

is also quite popular.

There are inherent time step requirements for first order time explicit schemes like the forward Euler scheme described earlier.. Analogous with the underlying wave equation with a speed function V, the CFL condition requires the front to cross no more than one grid cell. The requirement is thus,

$$\max_{\Omega} V \Delta t \leq \Delta x \tag{3.73}$$

where the maximum of V is taken over all possible grid points, not simply those corresponding to the zero level set. Once the the maximum value of V is determined, evaluating an appropriate time step is straightforward.

Chapter 4

Multidisciplinary Optimization

4.1 Multi-objective optimization: Problem Statement and Pareto Optimality

There are many approaches to solving multidisciplinary design problems. One of them is multi-objective optimization[1, 36]. In the multi-objective optimization approach, objective functions represent design objectives from more than one design discipline. As industrial designs become more complex, the design process has to adapt to remain competitive. Concurrent consideration of multiple design objectives from multiple disciplines enable the creation of a robust design at a competitive pace and cost. In addition to the benefits of multi-objective optimization, multidisciplinary optimization exploits the implicit interactions between design objectives from various design disciplines.

Multi-objective optimization problems are defined as determining j number of $\mathbf{b}$ design variables

$$\mathbf{b} = [b_1, ..., b_j]^T \tag{4.1}$$

such that n number of $\mathbf{f}$ design objectives

$$\mathbf{f} = [f_1, ..., f_n]^T \tag{4.2}$$

are minimized subjected to equality and inequality constraints.

$$\mathbf{h}(\mathbf{b}) = 0 \tag{4.3}$$

$$\mathbf{g}(\mathbf{b}) \leq 0 \tag{4.4}$$

In order to discuss any multi-objective optimization methods, it is a prerequisite to first start with the definition of the solution to search for[36]. Multi-objective optimization methods, in contrast to single objective optimization methods, take into account the tradeoff of multiple design objectives and search for a design solution space instead of a single design solution. For any arbitary set of design objectives, it is not analytically possible to evaluate the design solution space. Numerical methods have to be used to search for multiple design solutions that well describe the desired solution space. Such design solutions can be determined by checking their Pareto optimality[36]. If m design objectives are considered, a design solution with design variables $\mathbf{b}_u$ is Pareto optimal if and only if there is no other design solution with design variables $\mathbf{b}_v \in R^n$ for which $\mathbf{v} = \mathbf{f}(\mathbf{b}_v) = [v_1, ..., v_m]^T$ dominates $\mathbf{u} = \mathbf{f}(\mathbf{b}_u) = [u_1, ..., u_m]^T$, i.e., there is no vector $\mathbf{b}_v$ such that the following equation is satisfied.

$$v_i \leq u_i, \forall i \in \{1, ..., m\} \wedge v_i < u_i, \exists i \in \{1, ..., m\} \tag{4.5}$$

Pareto optimal solutions are the solutions of any multi-objective optimization problem as they cannot be dominated by any other solutions and no Pareto optimal solutions can be improved upon without affecting the Pareto optimality of its surrounding solutions. The n-dimensional objective space made up of the Pareto optimal solutions is termed the tradeoff curve or Pareto front[11]. There are two different characteristics of a Pareto front. The first is its curvature, convex or concave, and the second is its continuity. Figure 4.1 shows a discontinuous Pareto front consisting of both a convex and concave curve.

The multi-optimization process follows through a similar process as with the single optimization process as shown in Figure 3.2. The distinctive difference is at the evaluation stage. Instead of a evaluating a single objective function, multi-objective optimization requires

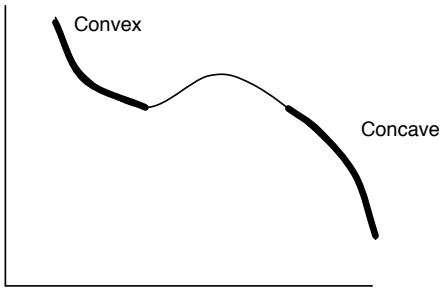


Figure 4.1: Discontinuous convex and concave Pareto fronts

the simultaneously evaluation of multiple objective functions. Multi-objective optimization methods can be broadly categorized into Scalarization methods and Pareto methods[1]. The aim of both methods are the same. They provide a set of optimal alternative solutions for decision makers to choose from.

4.2 Multi-objective sizing optimization

4.2.1 Real valued function

The fundamental concept of scalarization methods is to solve multi-objective optimization problems by translating multiple objective functions into a single scalar objective. This results in a scalar problem and solution methods for single objective optimization can easily be used. A popular multi-objective optimization method is the weighted sum approach[1, 36]. This approach combines multiple objective functions into a scalar objective function by constructing a weighted sum of all the objectives. It is represented by the following weighted sum equation

$$f(\mathbf{b}) = \sum_{i=1}^{n} w_i f_i(\mathbf{b}) \tag{4.6}$$

where for n objectives, w_i represents the weight attached to the i^{th} objective. The selection of the weights is debatable as they do not explicitly express the relative importance of each

objective. A major disadvantage of the weighted sum approach is that it cannot be used to find Pareto optimal solutions in multi-objective problems with concave Pareto fronts.

In order to evaluate multiple objectives of varying magnitudes equally, a scalar function has to be introduced to represent the contributions of each of the multiple objectives. The scalar function introduced and used here is termed the real valued function[36]. This real valued function has been formulated to work with multi-objective problems and it is consistent with the definition of its solutions. It requires a population of design alternatives and it returns multiple solutions. An advantage of this function is that it does not require the explicit selection of weights to represent the contributions of each of the multiple objectives. The individual real valued function for the i^{th} individual objective is given by

$$\phi_i' = \frac{f_i - f_{i,min}}{f_{i,max} - f_{i,min}} \tag{4.7}$$

where $f_{i,min}$ and $f_{i,max}$ represent the minimum and maximum values in a population sample. The individual real valued function is used to normalize the respective objective functions before aggregating them with weight values into the total real valued function. For the total number of n number of design objectives, the total real valued function is given by

$$\Phi = \sum_{i=1}^{n} w_i \phi_i' \tag{4.8}$$

and its gradient

$$\begin{aligned} \mathbf{g} &= \frac{d\Phi}{d\mathbf{b}} \\ &= \sum_{i=1}^{n} w_i \frac{d\phi_i'}{d\mathbf{b}}. \end{aligned} \tag{4.9}$$

The real valued function has been successfully used in multi-objective sizing optimization[36]. This dissertation presents the first instance of its applications in solving multi-objective topology optimization.

4.2.2 Gradient update and constraint handling

Given the real valued function described above, in order to evaluate the next state $k+1$ of the design variable, the general update equation

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \alpha_k \mathbf{d}_k \tag{4.10}$$

when applying the steepest descent method becomes

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \alpha_k \mathbf{g}_k \tag{4.11}$$

and when applying Newton-type methods becomes

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \alpha_k (\nabla \mathbf{g}_k)^{-1} \mathbf{g}_k \tag{4.12}$$

where

$$\mathbf{H} = \nabla \mathbf{g}_k. \tag{4.13}$$

A Quasi-Newton method is used to approximate $\mathbf{H}$ and increase the efficiency of the search. The BFGS update method will be used as it only requires an initial $\mathbf{B}$, the change in design variables $\partial\mathbf{b}$ and the change in first order gradients $\partial\mathbf{g}$.

$$\partial \mathbf{b}_k = \mathbf{b}_{k+1} - \mathbf{b}_k \tag{4.14}$$

$$\partial \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \tag{4.15}$$

The update equation for approximating $\mathbf{H}$ is

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \left[\frac{\partial \mathbf{g} \partial \mathbf{g}^T}{\partial \mathbf{g}^T \partial \mathbf{b}}\right]_k - \left[\frac{(\mathbf{H}\partial \mathbf{b})(\mathbf{H}\partial \mathbf{b})^T}{\partial \mathbf{b}^T \mathbf{H} \partial \mathbf{b}}\right]_k. \tag{4.16}$$

However, as only the inverse of $\mathbf{H}^{-1}$ is required, it is more effective to directly approximate it. The update equation for approximating $\mathbf{B} = \mathbf{H}^{-1}$ is

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \left[1 + \frac{\partial \mathbf{g}^T \mathbf{B} \partial \mathbf{g}}{\partial \mathbf{x}^T \partial \mathbf{g}}\right]_k \left[\frac{\partial \mathbf{b} \partial \mathbf{b}^T}{\partial \mathbf{b}^T \partial \mathbf{g}}\right]_k - \left[\frac{\partial \mathbf{b} \partial \mathbf{g}^T \mathbf{B} + \mathbf{B} \partial \mathbf{g} \partial \mathbf{b}^T}{\partial \mathbf{b}^T \partial \mathbf{g}}\right]_k. \tag{4.17}$$

The design variables are now updated by

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \alpha_k \mathbf{B}_k \mathbf{g}_k. \tag{4.18}$$

The Armijo line search[34] is used to evaluate the search step α_k. It is an inexact line search method which guarantees a sufficient degree of accuracy to ensure convergence. This method starts with $\alpha_k = 1$ and continues to halve it until the following condition is met.

$$\Phi(\mathbf{b}_k - \alpha_k \mathbf{d}_k) < \Phi(\mathbf{b}_k) + \epsilon \alpha_k \mathbf{g}_k \mathbf{d}_k \tag{4.19}$$

where $0 < \epsilon < 1$.

Constraint handling is carried out by using an active set strategy with Lagrange multipliers described in Section 3.2.

4.2.3 Pareto pooling

Pareto methods are named after Vilfredo Pareto (1848-1923) with whom the concept of Pareto optimality is credited to. Pareto methods attempt to find a set of efficient solutions, $\mathbf{b}^*$, such that the corresponding n number of objectives are non-dominated in n-dimensional objective space. Unlike scalarization methods, Pareto methods do not attempt to translate the multi-objective problem into a single objective optimization problem. Instead, objectives functions are kept separate throughout the optimization process[1]. Pareto methods use the Pareto optimality concept of non-dominance to distinguish between the solutions. The simplest form of a Pareto method is a combination of design space exploration and dominance filtering.

Pareto methods are appealing as they are able to generate a well distributed approximation of a local Pareto front. However, there are two important issues. Firstly, a complete evaluation of the solution space is impossible due to the j-dimensionality of the variable vector, $\mathbf{b}$, and the required computational effort for evaluating the respective objectives, $\mathbf{f}$. The second issue is that the solutions obtained from Pareto methods are just approximations

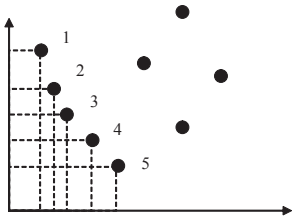


Figure 4.2: Pareto optimal solutions in two-dimensional objective space

of elements on the actual Pareto front. They only satisfy the Pareto optimality concept and are not necessary the exact Pareto optimal solutions.

Within Pareto methods, there are two prominent ranking methods[11]. Ranking is a process used to determine the degree of Pareto optimality a solution has. Rank 1 being the Pareto optimal solutions. Rank 2 being the second set of Pareto optimal solutions if Rank 1 solutions are removed and so on. The first proposed by Goldberg (1989), is a method of ranking and fitness assignment based on the Pareto optimality condition of each individual solution. With reference to Figure 4.2 individual solutions 1-5 are assigned rank 1 as they are undominated. Subsequent solutions—after the removal of rank 1—are assigned rank 2 and so on and so forth. Figure 4.3 depicts the process.

The second ranking method proposed by Fonseca and Fleming (1995) varies slightly. In addition to implementing the Pareto optimality condition for assigning ranks, the rank of each individual solution is further defined by an addition of one to the current number of solutions it is currently dominated by. For example, an individual in rank 2 and is dominated by 3 other individuals is given the rank value 5. Figure 4.4 depicts the process with the assigned rank in square brackets. Although all dominant solutions are similarly assigned rank 1, this process also shows the number of solutions dominating a particular solution.

Pareto methods do not suffer the same disadvantage at a concave Pareto front as scalarization methods. On the other hand, because they use the notion of non-dominance, they tend to be computationally heavy. Pareto methods such as the adaptive weighted sum approach[9] is able to find Pareto optimal solutions with a concave Pareto front. However, it

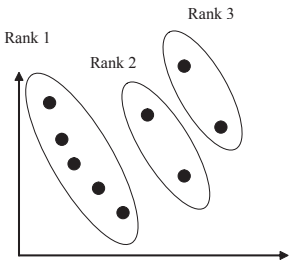


Figure 4.3: Goldberg's ranking method

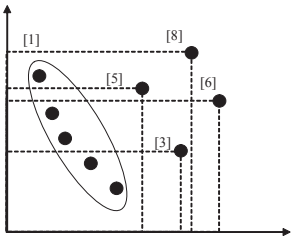


Figure 4.4: Fonseca and Fleming's ranking method

is still acknowledged that the scalar weighted sum approach offers the greatest transparency to non-expert users. Other prominent Pareto methods include Multi-objective Genetic Algorithms (MOGAs)[42, 11, 14, 15] which implement ideas from evolution theories to solve mutli-objective optimization problems

4.2.4 Center-of-Gravity method

The center-of-gravity method[36] is a novel decision maker that does not require weights to be explicitly defined in its formulation. Another novelty is that the center-of-gravity method selects a solution from the design variables' solution space, where different units of measure exists, because the controllable aspects in any design problem are its design variables. This method selects the closest solution to the center-of-gravity solution in $\mathbf{b}$ solution space. Suppose the Pareto optimal solutions obtained from a sequence of multi-objective optimization are $\hat{\mathbf{b}}^i, i = 1, ..., n$, where n represents the number of solution sets.

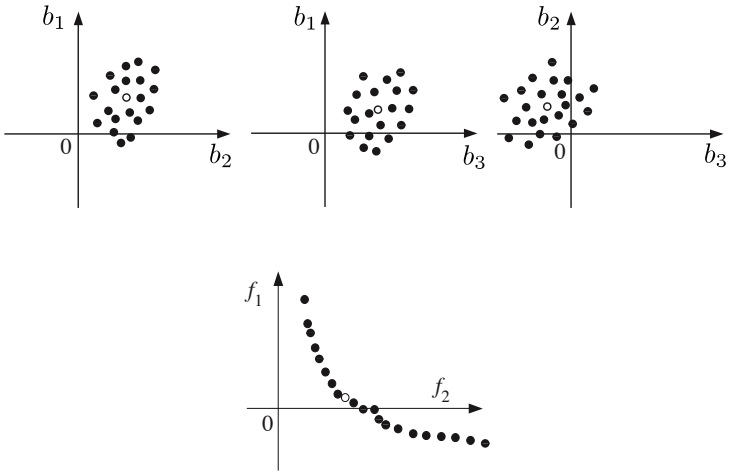


Figure 4.5: Center-of-Gravity method

The center-of-gravity is defined by the following equation.

$$\hat{\mathbf{b}} = \frac{\sum_{i=1}^{r} \hat{\mathbf{b}}^i}{r} \tag{4.20}$$

However, the center-of-gravity technique selects the final solution such that the distance, β, between the solution and the center of gravity solution is minimized. where n represents the number of design variables.

$$\beta = \left| \sqrt{\sum_{i=1}^{n} (\hat{\mathbf{b}}_n^i - \hat{\mathbf{b}}_n)^2} \right|_{min} \tag{4.21}$$

Figure 4.5 shows Pareto optimal solutions where two objectives $[\mathbf{f}] = [f_1, f_2]$ are mapped by three design variables $[\mathbf{b}] = [b_1, b_2, b_3]$. From Figure 4.5, the unfilled circle shows the final selected solution in a three-dimensional parameter space and its corresponding function values.

4.3 Multi-objective Shape Optimization

4.3.1 Calculating sensitivities for continuum systems

The objective function for minimizing the average compliance, $\Phi(\mathbf{u}) = \mathbf{F}^T\mathbf{u}$, and the equilibrium and volumetric constraints can be written into the functional:

$$\Phi^*(\mathbf{u}, \mathbf{b}, \mathbf{\Lambda}, \lambda) = \Phi(\mathbf{u}) + \mathbf{\Lambda}^T(\mathbf{F} - \mathbf{Q}(\mathbf{u}, \mathbf{b})) + \lambda(h(\mathbf{b}) - V_0) \tag{4.22}$$

where $\mathbf{u}$, $\mathbf{\Lambda}$, h and V_0 represents the displacement, the adjoint variable, volume constraint and the initial volume respectively. The variation of the functional $\Phi(\mathbf{u}, \mathbf{b}, \mathbf{\Lambda}, \lambda)$ would be given by

$$\begin{aligned}\delta\Phi^*(\mathbf{u}, \mathbf{b}, \mathbf{\Lambda}, \lambda) &= \left(\lambda\left(\frac{\partial h(\mathbf{b})}{\partial \mathbf{b}}\right)^T - \mathbf{\Lambda}^T\frac{\partial \mathbf{Q}}{\partial \mathbf{b}}\right)\delta\mathbf{b} + \delta\mathbf{u}^T\left(\mathbf{F} - \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{u}}\right)^T\mathbf{\Lambda}\right) \\ &\quad + \delta\mathbf{\Lambda}^T(\mathbf{F} - \mathbf{Q}) + \delta\lambda(h(\mathbf{b}) - V_0)\end{aligned} \tag{4.23}$$

By applying the stationary principle, the following equations are obtained.

$$\lambda\left(\frac{\partial h(\mathbf{b})}{\partial \mathbf{b}}\right)^T - \mathbf{\Lambda}^T\frac{\partial \mathbf{Q}}{\partial \mathbf{b}} = 0 \tag{4.24}$$

$$\mathbf{F} - \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{u}}\right)^T\mathbf{\Lambda} \equiv \mathbf{F} - \mathbf{K}^T\mathbf{\Lambda} = 0 \tag{4.25}$$

$$\mathbf{F} - \mathbf{Q} \equiv \mathbf{F} - \mathbf{K}\mathbf{u} = 0 \tag{4.26}$$

$$h(\mathbf{b}) - V_0 = 0 \tag{4.27}$$

Equation (4.24) shows that the adjoint variable is a conjugate of the Lagrange Multiplier. Equations (4.26) and (4.27) show the equilibrium constraint condition and the volume constraint condition respectively. By assuming that the condition $\mathbf{Q} = \mathbf{K}\mathbf{u}$ is satisfied in Eqs. (4.25) and (4.26), the following equation for the stiffness, $\mathbf{K}$ can be obtained.

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{u}} = \mathbf{K}^T \tag{4.28}$$

The relationship between $\mathbf{\Lambda}$ and $\mathbf{u}$ is now simply

$$\mathbf{\Lambda} = \mathbf{u}. \tag{4.29}$$

However, it should be stressed that this is only valid for linear problems where the objective function is to minimize average compliance. With reference to Eqs. (4.25) and (4.27), the sensitivity of functional Φ^* would be given as

$$\frac{\delta\Phi^*}{\delta\mathbf{b}} = -\left(\frac{\partial\mathbf{Q}}{\partial\mathbf{b}}\right)^T\mathbf{u} + \lambda\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}. \tag{4.30}$$

Since the real valued function is simply a normalized version of average compliance, it can be used in Equation (4.22) without any loss in generalization.

Now that the sensitivity of the structural response of interest is obtained, the real valued function of this individual structural response can be represented by

$$\phi_i' = w_i\frac{\phi_i - \phi_{i,min}}{\phi_{i,max} - \phi_{i,min}} \tag{4.31}$$

and its gradient by

$$\frac{d\phi_i'}{db_j} = \frac{w_i}{\phi_{i,max} - \phi_{i,min}}\left(\frac{d\phi_i}{db_j}\right). \tag{4.32}$$

4.3.2 Constraint handling

A physical representation is used to handle constraints when performing shape optimization. This is similar to the optimality criteria approach. In order to satisfy the volume constraint condition, the Lagrange Multiplier, λ, has to be evaluated separately. Considering that $\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}$ represents the direction, along which, volume would change the most, the volume constraint has be to implemented in such a way that the update direction, $\frac{\delta\Phi^*}{\delta\mathbf{b}}$ is oriented orthogonal to $\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}$. In the physical representation shown in Fig. 4.6, λ should be evaluated such that the inner product between $\frac{\delta\Phi^*}{\delta\mathbf{b}}$ and $\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}$ is zero.

$$\lambda = \frac{\left(\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}\right)^T\left(\left(\frac{\partial\mathbf{Q}}{\partial\mathbf{b}}\right)^T\mathbf{u}\right)}{\left(\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}\right)^T\frac{\partial h(\mathbf{b})}{\partial\mathbf{b}}} \tag{4.33}$$

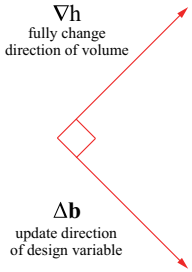


Figure 4.6: Volume constraint vectors

4.4 Multi-objective Topology Optimization

This section proposes a new class of problem for multi-objective topology optimization. Instead of using multiple objectives of the same nature, compliance for example, this novel problem uses an objective function which minimizes deviation from a desired curvature or topology specified beforehand by a designer or engineer. What this means is that structural topology is no longer confined to objective functions of only engineering metrics like structural compliance. It is now possible to add aesthetic and/or functional aspects to structural topologies through multi-objective topology optimization, resulting in a multidisciplinary design problem.

This section will first describe single objective level set methods before further describing the novel multi-objective topology optimization problem proposed in this dissertation.

4.4.1 Level set based topology optimization

Small deformation elastic behaviour is considered here. When representing topology implicitly with ϕ, the problem statement for a topology optimization based on strain energy

minimization, subjected to equilibrium and volumetric constraints, can be written as

$$\begin{aligned} \text{minimize} \quad & f(\phi) = \tfrac{1}{2}\int_\Omega H(\phi) E\boldsymbol{\varepsilon}^T \boldsymbol{D}\boldsymbol{\varepsilon} d\Omega \\ \text{s.t.} \quad & \nabla \cdot (H(\phi) E\boldsymbol{\sigma}) = \boldsymbol{p} \\ & \int_\Omega H(\phi) d\Omega = V_0 \end{aligned} \tag{4.34}$$

where E, $\boldsymbol{\varepsilon}$, $\boldsymbol{\sigma}$, $\boldsymbol{D}$, $\boldsymbol{p}$ and V_0 are the Young's modulus, strain, stress, characteristic tensor, external forces and constrained volume defined in Ω respectively. Using the Lagrange multiplier method, the objective function and the volume constraint can be expressed with the following functional $J(\phi, \lambda)$:

$$J(\phi, \lambda) = \frac{1}{2}\int_\Omega H(\phi) E\boldsymbol{\varepsilon}^T \boldsymbol{D}\boldsymbol{\varepsilon} d\Omega + \lambda \left(\int_\Omega H(\phi) d\Omega - V_0 \right) \tag{4.35}$$

where λ is a scalar Lagrange multiplier for the volume constraint. The velocity for evolving the function ϕ is obtained from the stationary condition of Equation (4.35):

$$\delta J(\phi, \lambda) = \delta_\phi J(\phi, \lambda) + \delta_\lambda J(\phi, \lambda) = 0. \tag{4.36}$$

In order to obtain a ϕ at the stationary point described by Equation (4.36), assuming the volume constraint is always satisfied, we need to evaluate $\delta_\phi J(\phi, \lambda) = 0$:

$$\begin{aligned} \delta_\phi J(\phi, \lambda) &= \frac{1}{2}\int_\Omega \frac{\partial H(\phi)}{\partial \phi} \delta\phi E\boldsymbol{\varepsilon}^T \boldsymbol{D}\boldsymbol{\varepsilon} d\Omega + \lambda \left(\int_\Omega \frac{\partial H(\phi)}{\partial \phi} \delta\phi d\Omega - V_0 \right) \\ &= \int_\Omega \left[\frac{1}{2} E\boldsymbol{\varepsilon}^T \boldsymbol{D}\boldsymbol{\varepsilon} + \lambda \right] \delta(\phi)\delta\phi d\Omega \\ &= 0 \end{aligned} \tag{4.37}$$

where the variation of the Heaviside function across the material boundary is given by

$$\delta_\phi H(\phi) = \frac{\partial H(\phi)}{\partial \phi}\delta\phi = \delta(\phi)\delta\phi = \delta\phi|_{\phi=0} \tag{4.38}$$

The corresponding Euler-Lagrange equation for any point in Ω can be expressed as

$$\left(\frac{1}{2} E\boldsymbol{\varepsilon}^T \boldsymbol{D}\boldsymbol{\varepsilon} + \lambda \right) \delta(\phi) = 0. \tag{4.39}$$

Since it is known that topological changes are only influenced by motion in the normal direction [43], the level set equation shown in Equation (3.25) can be used to solve Equation (4.39) numerically:

$$\frac{\partial \phi}{\partial t} + \left[\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} + \lambda\right] \delta(\phi) |\nabla \phi| = 0 \tag{4.40}$$

The discrete form of the update equation to evolve the level set is given by

$$\phi_{t+\Delta t} = \phi_t - \left[\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} + \lambda\right] \delta(\phi) |\nabla \phi| \Delta t. \tag{4.41}$$

Note that when ϕ is a signed distance function, the equation can be simplified to

$$\phi_{t+\Delta t} = \phi_t - \left[\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} + \lambda\right] \delta(\phi) \Delta t. \tag{4.42}$$

A gradient projection type method is used to calculate the value of λ [43]. Assuming that the volumetric constraint shown in Equation (4.34) is satisfied initially, this condition should be satisfied throughout the boundary evolution[31]:

$$\frac{d}{dt}\left(\int_\Omega H(\phi) d\Omega - V_0\right) = \int_\Omega \frac{\partial H(\phi)}{\partial \phi} \frac{\partial \phi}{\partial t} d\Omega = \int_\Omega \delta(\phi) \frac{\partial \phi}{\partial t} d\Omega = 0 \tag{4.43}$$

The Lagrange Multiplier, λ, can be obtained by substituting Equation (4.40) for $\frac{\partial \phi}{\partial t}$:

$$\begin{aligned} \int_\Omega \left[\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} + \lambda\right] \left(\delta(\phi)\right)^2 |\nabla \phi| d\Omega = 0 \\ \int_\Omega \frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \left(\delta(\phi)\right)^2 |\nabla \phi| + \lambda \int_\Omega \left(\delta(\phi)\right)^2 |\nabla \phi| d\Omega = 0 \\ \lambda = \frac{-\int_\Omega \frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \left(\delta(\phi)\right)^2 |\nabla \phi| d\Omega}{\int_\Omega \left(\delta(\phi)\right)^2 |\nabla \phi| d\Omega} \end{aligned} \tag{4.44}$$

With λ now specified, Equation (4.42) can easily be used to update ϕ.

There exists other method to calculating a suitable λ for satisfying the volume constraint. The bi-section algorithm[33] provides one such alternative.

4.4.2 Creative design: desired curves, surfaces and topology

In this section, curvature refers to curves and surfaces in Cartesian space, while shapes are simply a subset of curvature and refer to continuous curvature that are often connected.

Figure 4.7 shows a typical single objective optimization problem and outcome. When optimizing the topology of such a material domain subjected to the loading condition and volumetric constraints shown on the left in Figure 4.7, an outcome similar to the figure on the right of Figure 4.7 can be expected. A problem like this is usually based on engineering design principles, such as minimizing compliance. As explained earlier, multi-objective topology

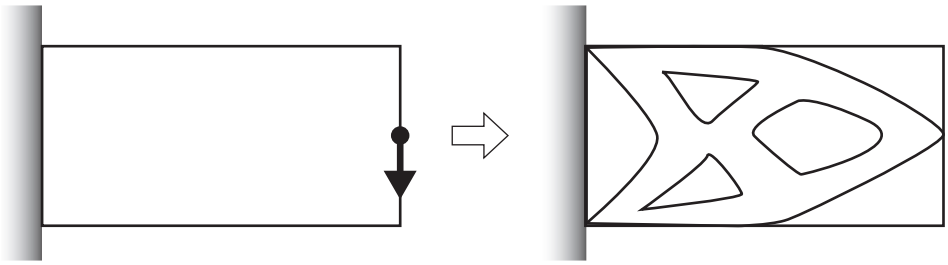


Figure 4.7: Standard topology optimization problem

optimization, so far, only uses a combination of compliance objective functions.

This dissertation would like to propose a new approach to multi-objective topology optimization. Instead of a problem similar to or consisting of a combination of Figure 4.7, we propose a multi-objective problem whereby aesthetics and/or functionality can be prescribed by a designer or engineer. Consider the picture in Figure 4.8. What if desired curvature (show in dashed lines) is prescribed and the designer or engineer hopes that this topological property would exist in the final optimized result. What would the outcome of this new multi-objective topology optimization problem be?

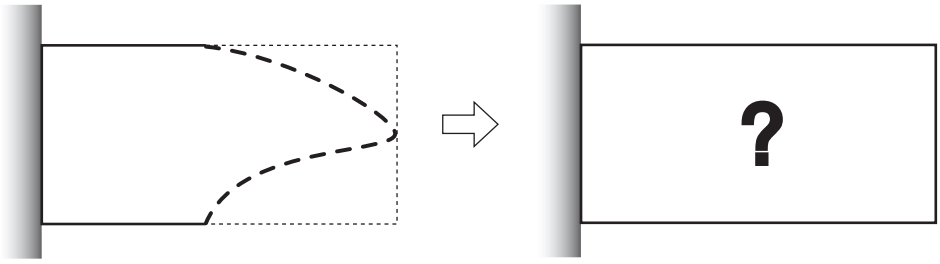
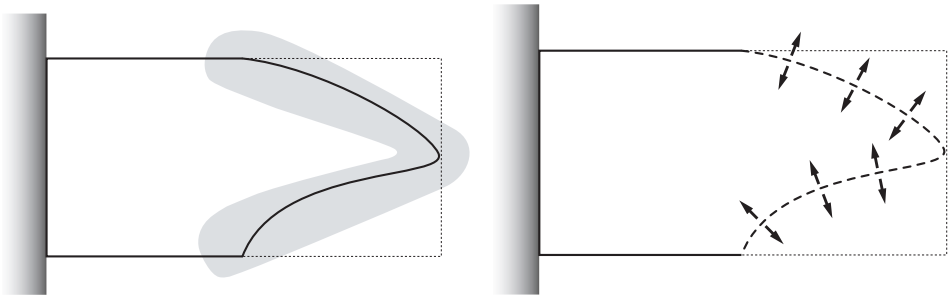


Figure 4.8: New problem for topology optimization

Usually when some curvature of shape is desired, the current technique used is simply to restrict movement in the concerned boundaries. In level set based topology optimization, the simplest method is to zero out the velocity field around the desired curvatures, for example the gray areas shown in Figure 4.9(a). By constraining the local velocity field to zero, the boundary does not move and the desired curvature is preserved.

However, preserving the exact boundary might not always be the best solution. Often, a slight perturbation in the boundary might improve compliance. Therefore, instead of constraining the local velocity field as shown in Figure 4.9(a), we propose letting the boundary move as shown in Figure 4.9(b). We will use multi-objective principles and work to find a balance between the compliance of the structure and, in case of Figure 4.9(b), keeping the desired boundary.



(a) Boundary is constrained (b) Boundary is not constrained

Figure 4.9: Restricting movement on the boundary

If the desired curvatures represent some form of architectural design and a structure has have minimum compliance, the general idea proposed here aims at finding a balance between those two objectives. The next section will detail the mathematical model which will be used to achieve this.

Measuring deviation from specific topology

Figure 4.10 shows two different topologies. In order to evolve one topology towards the other, we need to know how to describe the differences between the two topologies. Fortunately, this

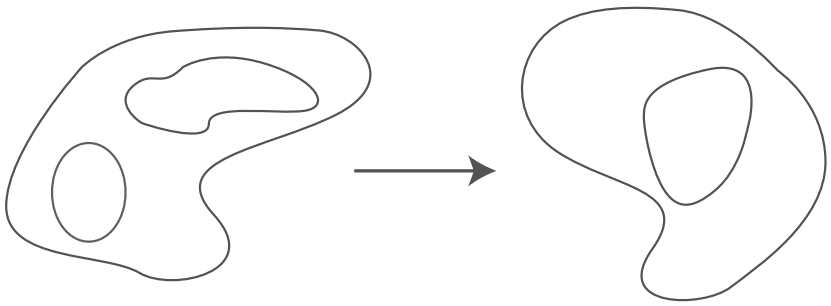


Figure 4.10: Evolving one topology towards the other

can be achieved very easily with signed distance level set functions: the difference between any two topology can be measured locally by taking the difference in the value of their level set function. This section will explore the model which will be used to solve the novel multi-objective topology optimization problem described above. In order to measure the deviation of a boundary from its desired curvature, the following objective function will be used:

$$f(\phi) = \int_{\Omega} \frac{1}{2}(\phi_{desired} - \phi)^2 d\Omega \tag{4.45}$$

Due to its similarity to the least square error objective function used for compliant mechanisms in [21], this objective function shall be termed the signed least square error objective function. The term "signed" is used because the velocity field derived from this objective function will also have direction. From this objective function, a velocity field that will evolve the level set function to $\phi_{desired}$ can be obtained. This velocity field is designed to move the front in the direction towards the desired curvature.

The derivative of this objective function with respect to ϕ presents such a velocity field.

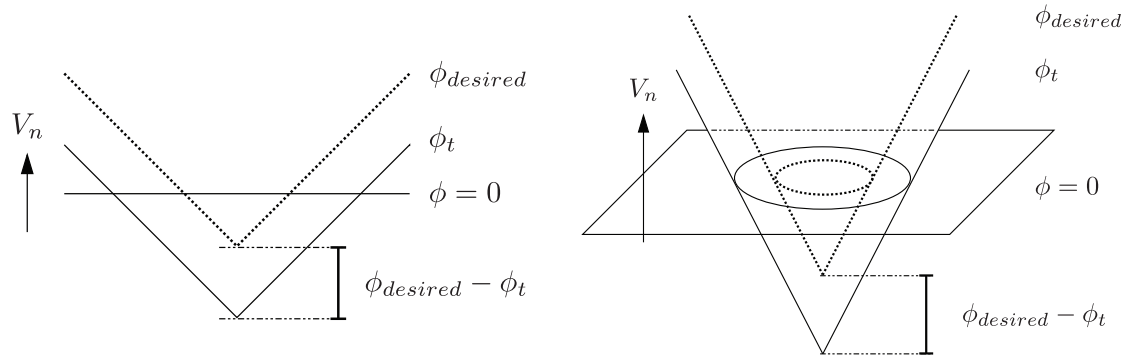
$$\frac{\partial f(\phi)}{\partial \phi} = -\int_{\Omega} (\phi_{desired} - \phi) d\Omega \tag{4.46}$$

This is intuitive because if we were to use the level set equation

$$\phi_t + V_n |\nabla \phi| = 0 \tag{4.47}$$

the velocity given by Equation (4.46), evolves ϕ in the normal direction $\nabla\phi$ to $\phi_{desired}$ in a single step. For the example shown in Figure 4.11, the velocity V_n has to be applied in order

to move ϕ_t to $\phi_{desired}$. Here, the reason for using the term "signed" to describe the objective function becomes obvious, if the situation in Figure 4.11 is reversed, the velocity function Equation (4.46) would have its sign reversed and evolve the level set function accordingly.



(a) In 2D (b) In 3D

Figure 4.11: Obtaining desired level set values ϕ

The selection of $\phi_{desired}$ is very straightforward. Since ϕ is initialized to signed distance, picking $\phi_{desired}$ near the desired boundary only involves picking nodes within a certain distance of the desired boundary. Usually, $\phi_{desired}$ are simply ϕ values selected from within a specified narrowband of the desired boundary. Although the choice of width for the narrowband is left entirely up to user selection, a wide narrowband for choosing $\phi_{desired}$ is recommended since interaction with the desired boundary via the signed least square error objective function is only limited to within this particular narrowband. We shall refer to the domain specified by this narrowband as Ω_{unb}, where "*unb*" is an abbreviation for "user-defined narrowband". The local variant of Equation 4.45 in this case is simply written as

$$f(\phi) = \int_{\Omega_{unb}} \frac{1}{2}(\phi_{desired} - \phi)^2 d\Omega_{unb} \tag{4.48}$$

and evaluated accordingly.

If the weighted sum approach is used to solve this multi-objective topology optimiza-

tion problem, the following procedure can be followed. The objective functions are first aggregated into a functional J:

$$J(\phi, \lambda) = w_1 f_1(\phi) + w_2 f_2(\phi) + \lambda h(\phi) \tag{4.49}$$

$$\begin{aligned} J(\phi, \lambda) = \frac{w_1}{2} \int_\Omega H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} d\Omega + \frac{w_2}{2} \int_{\Omega_{unb}} (\phi_{desired} - \phi) d\Omega_{unb} \\ + \lambda \left(\int_\Omega H(\phi) d\Omega - V_0 \right) \end{aligned} \tag{4.50}$$

By taking the derivative of functional J with respect to ϕ,

$$\frac{\partial J(\phi, \lambda)}{\partial \phi} = \frac{w_1}{2} \int_\Omega \left[E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} + \lambda \right] \delta(\phi) d\Omega - w_2 \int_{\Omega_{unb}} (\phi_{desired} - \phi) d\Omega_{unb} = 0$$

we can arrive at the corresponding Euler-Lagrange equation for any point in Ω,

$$w_1 \left(\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \delta(\phi) \right) - w_2 (\phi_{desired} - \phi) s_{unb}(\phi) + \lambda \delta(\phi) = 0 \tag{4.51}$$

where $s_{unb}(\phi)$ is simply a switch function used to determine if the respective node is within the user-specified narrowband:

$$s_{unb}(\phi) = \begin{cases} 1 & \text{if } \phi \in \Omega_{unb}, \\ 0 & \text{else.} \end{cases} \tag{4.52}$$

The level set equation used to solve this problem can then be written as:

$$\frac{\partial \phi}{\partial t} + \left[w_1 \left(\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \delta(\phi) \right) - w_2 (\phi_{desired} - \phi) s_{unb}(\phi) + \lambda \delta(\phi) \right] |\nabla \phi| = 0. \tag{4.53}$$

The discrete form of the update equation to evolve the level set is given by

$$\phi_{t+\Delta t} = \phi_t - \left[w_1 \left(\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \right) \delta(\phi) - w_2 (\phi_{desired} - \phi) s_{unb}(\phi) + \lambda \delta(\phi) \right] |\nabla \phi| \Delta t. \tag{4.54}$$

However, both objective functions are measured in different scales. Therefore there is a need to normalize the objective functions to avoid complete dominance by objective functions with large values. The normalization and aggregation of objective functions will be carried

out by the real valued function described in Chapter 4.2.1. The concept is to normalize the objective functions first and then aggregate them with weights into a single function for optimization. When using the real valued function to normalize the respective objective functions, the following discrete equation can be used to evolve the level set function in time:

$$\phi_{t+\Delta t} = \phi_t - \left[w_1 \frac{\left(\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon}\right) \delta(\phi)}{f'_{1,max} - f'_{1,min}} - w_2 \frac{(\phi_{desired} - \phi) s_{unb}(\phi)}{f'_{2,max} - f'_{2,min}} + \lambda \delta(\phi) \right] |\nabla \phi| \Delta t \tag{4.55}$$

where maximum and minimum values of the objective functions, $f'_{1,max}$, $f'_{1,min}$, $f'_{2,max}$ and $f'_{2,min}$, are discrete and selected from their respective domains, Ω and Ω_{unb}. This is significantly different from the way the real valued function is used in sizing or shape optimization. In sizing and shape optimization, the maximum and minimum function values were chosen from a set of design configurations. Here, it is chosen from the design variables of a specific design configuration. When the functions for $f'_1(\phi)$ and $f'_2(\phi)$ are defined discretely,

$$f'_1(\phi) = \frac{1}{2} H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \tag{4.56}$$

$$f'_2(\phi) = \frac{1}{2} (\phi_{desired} - \phi)^2, \tag{4.57}$$

the values for the first objective function in this case are given by

$$f'_{1,max} = \max(f'_1(\phi) \in \Omega) \tag{4.58}$$

$$f'_{1,min} = \min(f'_1(\phi) \in \Omega) \tag{4.59}$$

and the values for the second objective function, the signed least square error, are then given by

$$f'_{2,max} = \max(f'_2(\phi) \in \Omega_{unb}) \tag{4.60}$$

$$f'_{2,min} = \min(f'_2(\phi) \in \Omega_{unb}). \tag{4.61}$$

In the equation above, the maximum and minimum values, f'_{max} and f'_{min}, are selected automatically at each iteration; $f'_{1,max}$ and $f'_{1,min}$ are selected globally within Ω and $f'_{2,max}$ and $f'_{2,min}$ are selected locally from within Ω_{unb}. Since topological preferences are defined

locally by $\phi_{desired}$, a local selection of maximum and minimum values for $f_2'(\phi)$ is justifiable. The same procedure can easily be employed for all instances of the signed least square error objective function. The corresponding equation to evaluate the Lagrange Multiplier is given by:

$$\lambda = \frac{-\left[w_1 \int_\Omega \frac{\frac{1}{2} E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \delta(\phi)}{f'_{1,max} - f'_{1,min}} d\Omega - w_2 \int_{\Omega_{unb}} \frac{(\phi_{desired} - \phi)}{f'_{2,max} - f'_{2,min}} d\Omega_{unb}\right] \delta(\phi) |\nabla \phi|}{\int_\Omega \left(\delta(\phi)\right)^2 |\nabla \phi| d\Omega}. \tag{4.62}$$

Desired topology in case of voids

Unlike curvature and shapes, voids are slightly different in nature. A void can possess any shape, so long as its a void. When dealing with desired voids, an additional switch function is required.

If($\phi < \phi_{desired}$) $\phi_{desired} - \phi = 0$, the objective function will be set to zero. Nothing is done when $\phi > \phi_{desired}$. Both cases are shown in Figure 4.12. This is intuitive because for the case when $\phi < \phi_{desired}$, the void is obviously expanding. Therefore, the initial desired void is not compromised and, as was previously stated, there is no need to only restrict the void such that is bears the same shape as the initial void.

However, the same cannot be said for when the void is contracting, $\phi > \phi_{desired}$. Since there is an initial void that is desired, if the level set function is contracting around that void, the boundaries must be treated like curvature. Hence no special treatment is required.

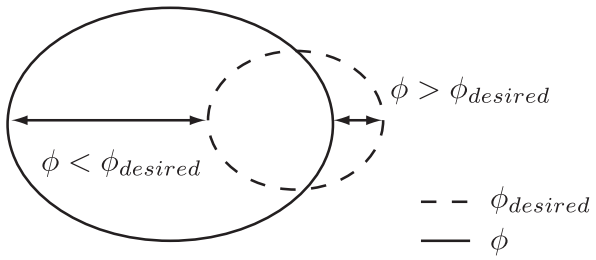


Figure 4.12: Movement of voids

4.4.3 Grid Data

When creating the Cartesian grid and the initial level set function, one usually creates

1. an array of nodes
2. an array of elements
3. an array storing the connectivity of the grid.

For example, with reference to Figure 4.13 where element numbers are circled, an array of node numbers will contain information, such as coordinates, of nodes 1-28; an array of element will contain information about elements 1-18, including connectivity information, for example, element 1 has connectivity {1,2,9,8}.

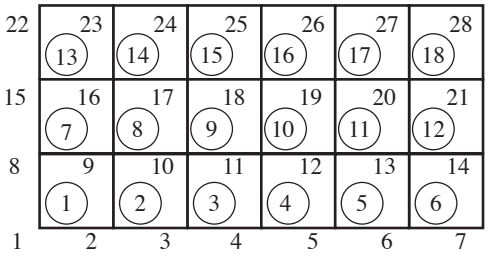


Figure 4.13: Sample grid with node numbers and element numbers

In addition to the above data structures, another data structure is also useful to keep in mind. It is the data structure specifying the surrounding elements of any node. For example, element 9 has 4 surrounding elements {1,2,8,7} and element 15 has only 2 surrounding elements {7,13}. This information can be useful in nodal renumbering operations.

Chapter 5

Efficiency and Adaptivity

5.1 Sizing optimization: Multi-objective adjoint variable method

Structural sizing optimization employing sensitivity analysis requires frequent calculation of the stiffness matrix in order to compute sensitivity. When problems become large and the number of variables greatly exceed the number of objective functions, calculating sensitivity becomes rather tedious. This section will detail a multi-objective adjoint variable formulation which aims to reduce computation time when performing such analysis.

The adjoint variable method has been shown to be identical to the Lagrange Multiplier approach [44]. This section will present an extension of the adjoint variable method that can be used in multi-objective analysis.

Suppose it is desired to calculate the sensitivities of m multiple structural responses and $\phi_i = \phi_i(\mathbf{U}(\mathbf{b}), \mathbf{b})$ is a structural response of interest that is given as a function of both displacement $\mathbf{U}(\mathbf{b})$ and design variable $\mathbf{b}$. The Lagrangian L is defined as

$$L = \phi - \boldsymbol{\lambda}^T \boldsymbol{\psi} \tag{5.1}$$

where $\boldsymbol{\psi}$ represents the internal force vector, $\mathbf{Q} = [Q_x^1, Q_y^1, ..., Q_x^n, Q_y^n]^T$, and the external

force vector, $\mathbf{F} = [F_x^1, F_y^1, ..., F_x^n, F_y^n]^T$ for n number of nodes.

$$\boldsymbol{\psi} = \mathbf{Q} - \mathbf{F} = 0 \tag{5.2}$$

and $\boldsymbol{\lambda}$ represents the Lagrange multiplier. The variational form of Eq. (5.1) due to δb_j

$$\delta L = \delta\phi - \delta\boldsymbol{\lambda}^T\boldsymbol{\psi} - \boldsymbol{\lambda}^T\delta\boldsymbol{\psi} \tag{5.3}$$

shows that $\delta L = \delta\phi$ holds because both $\boldsymbol{\psi}$ and $\delta\boldsymbol{\psi}$ must be zero. By using Eq. (5.1) and applying the stationary condition of L with respect to $\mathbf{U}$ i.e. $\frac{\partial L}{\partial \mathbf{U}} = 0$ the following equation

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}} &= \frac{\partial \phi}{\partial \mathbf{U}} - \left(\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{U}}\right)^T \mathbf{Q} - \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{U}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{U}}\right)^T \mathbf{F} + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^T \boldsymbol{\lambda} \\ &= \frac{\partial \phi}{\partial \mathbf{U}} - \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{U}}\right)^T \boldsymbol{\lambda} \end{aligned} \tag{5.4}$$

can be obtained because $\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{U}}$ and $\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$ are zero.

Finally, by using Eq. (5.1) and the stationary condition that $\frac{\partial L}{\partial \mathbf{U}} = 0$, the following derivative equation for $\frac{dL}{db_j}$ can be obtained.

$$\begin{aligned} L &= \phi - \boldsymbol{\lambda}^T\boldsymbol{\psi} \\ dL &= \left(\frac{\partial L}{\partial b_j}\right) db_j + \left(\frac{\partial L}{\partial \mathbf{U}}\right)^T d\mathbf{U} \\ \frac{dL}{db_j} &= \frac{\partial L}{\partial b_j} \end{aligned} \tag{5.5}$$

where

$$\begin{aligned} \frac{\partial L}{\partial b_j} &= \frac{\partial \phi}{\partial b_j} - \frac{\partial \boldsymbol{\lambda}^T}{\partial b_j}\mathbf{Q} - \boldsymbol{\lambda}^T\frac{\partial \mathbf{Q}}{\partial b_j} + \frac{\partial \boldsymbol{\lambda}^T}{\partial b_j}\mathbf{F} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{F}}{\partial b_j} \\ &= \frac{\partial \phi}{\partial b_j} - \boldsymbol{\lambda}^T\frac{\partial \mathbf{Q}}{\partial b_j} \end{aligned} \tag{5.6}$$

because $\frac{\partial \boldsymbol{\lambda}^T}{\partial b_j}$ and $\frac{\partial \mathbf{F}}{\partial b_j}$ must be zero. Taking Eq. (5.5) into consideration and that $\delta\phi = \delta L$, i.e. the derivatives are the same, the sensitivity of the desired structural response ϕ can be

obtained:

$$\begin{aligned} \frac{d\phi}{db_j} &= \frac{\partial L}{\partial b_j} \\ &= \frac{\partial \phi}{\partial b_j} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{Q}}{\partial b_j} \end{aligned} \tag{5.7}$$

In Eq. (5.7), $\frac{\partial \mathbf{Q}}{\partial b_j}$ can be calculated by using finite difference methods and $\boldsymbol{\lambda}$ is calculated by solving the adjoint structural system equation:

$$\begin{aligned} \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{U}}\right)^T \boldsymbol{\lambda} &= \frac{\partial \phi}{\partial \mathbf{U}} \\ \mathbf{K}^T \boldsymbol{\lambda} &= \frac{\partial \phi}{\partial \mathbf{U}} \end{aligned} \tag{5.8}$$

When the desired structural response ϕ is not an explicit function of the design variables b_j, the term $\frac{\partial \phi}{\partial b_j}$ can be dropped. The adjoint variable method has an advantage over the direct differentiation method when the number of $\mathbf{b}$, N_b, is greater than the number of ϕ, N_ϕ. This is because Eq. (5.8) is solved for N_ϕ times and the sensitivities $\frac{d\phi}{db_j}$ is easily calculated from Eq. (5.7). In constrast, for global finite difference methods, the equations must be solved N_b times to calculate $\frac{d\phi}{db_j}$. For example, a problem with two ϕ and ten b has to go through twenty calculations to evaluate the sensitivities when using global finite difference methods. However, the adjoint variable method only requires two calculations to evaluate the sensitivities.

5.2 Shape optimization: Discrete Force Method

Sometimes when performing shape optimization, elements can collapse or boundaries rendered non-smooth and zigzag from the movement of nodes. This section will detail a smoothing technique used to overcome the above problems.

In the case of steepest descent method, the variation of $\mathbf{b}$ is set to the vector parallel to the sensitivity of functional Φ^* with respect to the design variables, as

$$\Delta \mathbf{b} = -\alpha \cdot \frac{\partial \Phi^*}{\partial \mathbf{b}} = -\alpha \cdot \left\{ -\left(\frac{\partial \mathbf{Q}}{\partial \mathbf{b}}\right)^T \mathbf{u} + \lambda \frac{\partial h(\mathbf{b})}{\partial \mathbf{b}} \right\} \tag{5.9}$$

where α is a properly defined positive constant. Please note the internal force vector $\mathbf{Q}$ in this section does not reflect the same boundary conditions as shown earlier in Section 5.1. It is shown easily that the functional Φ^* always decreases with the variation of $\mathbf{b}$.

$$\Delta\Phi^* = \left(\frac{\partial\Phi^*}{\partial\mathbf{b}}\right)^T \cdot \Delta\mathbf{b} = \left(\frac{\partial\Phi^*}{\partial\mathbf{b}}\right)^T \cdot \left(-\alpha \cdot \frac{\partial\Phi^*}{\partial\mathbf{b}}\right) = -\alpha \cdot \left|\frac{\partial\Phi^*}{\partial\mathbf{b}}\right|^2 < 0 \tag{5.10}$$

However, this procedure often causes some defects especially in shape optimization problems, which result in non-smooth zigzag boundary and collapse of elements. Thus, a remeshing technique is required. In order to overcome this difficulty, a discrete force method which works as a smoothing technique for updating geometry and can be applied to any nonlinear problems was developed [45]. The discrete force method is termed as such because it is formulated in a complete discrete form and body force is applied instead of the surface traction force. Referring to the discrete force method [45], the variation vector of design variable is determined by solving the following equation.

$$\frac{\partial\mathbf{Q}}{\partial\mathbf{u}} \cdot \Delta\mathbf{b}_{DFM} = -\alpha \cdot \frac{\partial\Phi^*}{\partial\mathbf{b}} \tag{5.11}$$

$$\Delta\mathbf{b}_{DFM} = -\alpha \left\{\frac{\partial\mathbf{Q}}{\partial\mathbf{u}}\right\}^{-1} \cdot \frac{\partial\Phi^*}{\partial\mathbf{b}} \tag{5.12}$$

Here, the tangent stiffness is again utilized as a smoother to improve the irregularity of original sensitivity vector $\Delta\mathbf{b}$. If the tangent stiffness is positive definite, it is again shown that the functional Φ^* always decreases with $\mathbf{b}_{DFM}$.

$$\Delta\Phi^* = \left(\frac{\partial\Phi^*}{\partial\mathbf{b}}\right)^T \cdot \Delta\mathbf{b}_{DFM} = -\alpha \left(\frac{\partial\Phi^*}{\partial\mathbf{b}}\right)^T \cdot \left\{\frac{\partial\mathbf{Q}}{\partial\mathbf{u}}\right\}^{-1} \cdot \frac{\partial\Phi^*}{\partial\mathbf{b}} = -\frac{1}{\alpha}\Delta\mathbf{b}_{DFM}^T \cdot \frac{\partial\mathbf{Q}}{\partial\mathbf{u}} \cdot \Delta\mathbf{b}_{DFM} < 0 \tag{5.13}$$

In this formulation, only the internal force is re-computed with the initial coordinates perturbed in the direction of the adjoint vector $\mathbf{\Lambda}$ and the same routine for the evaluation of internal force can be utilized. The formulation can be applied to general nonlinear problems of which equilibrium equation can be expressed in discrete form as in $\mathbf{Q}(\mathbf{u}) = \mathbf{F}$. As for shape optimization, any kinds of objective functions can be defined as a function of $\mathbf{u}$ and $\mathbf{b}$.

5.3 Topology optimization

This section will present two methods which can be used to significantly speed up the optimization process for level set based topology optimization. The first method details a high-pass filter that aggressively removes material based on strain energy values in the material domain Implementation of the second method will significantly reduce the computational costs associated with FEM calculations.

5.3.1 High-pass filter

It is often stated in literature that the final topology as well as the speed of the level set based topology optimization process is dependent on the configuration of the initial level set, particularly the number of initial voids. This section aims to introduce a numerical scheme that removes the dependency of the optimization process on the initial configuration of the level set by only providing an initial level set with a full material domain.

Instead of gradually removing inefficient material elements until it reaches an optimal state described by the optimization objectives and constraints, this paper proposes removing material nodes based on nodal strain energy. Since topology is defined implicitly at the nodes, removing material nodes is more straightforward than removing elements. Due to the implicit description of topology, the removing of material is simply accomplished by simply setting the respective ϕ values at the nodes to -1.

In order to remove material efficiently when using level set functions, we propose the use of a high-pass filter. By using the proposed high-pass filter, it is possible to remove a significant amount of material in a quick and deterministic manner. The high-pass filter proposed here works by removing material nodes with low strain energy and only allowing nodes with high strain energy to pass.

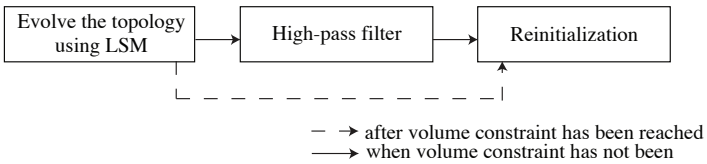


Figure 5.1: Flowchart when implementing high-pass filter

Numerical Implementation

The detailed optimization procedure with the high-pass filter is as follows:

- Initialization of full material domain level set function which has only non-zero signed distance.
- Turn on high-pass filter.
- Analyzing the structural response and evaluate nodal strain energy.
- Evolve the topology using LSM.
- Check if volume condition has been reached for the first time.
- If yes, turn off high-pass filter for the rest of the optimization process.
- If no, high-pass filter is left on.
- Reinitialize the level set function.
- Check for convergence and repeat steps (3)-(6) until the final topology is stable.

The schematic view of the filter and how it fits into the topology optimization process is shown in Figure 5.1. Step (5) checks if the volume constraint has been reached for the first time. Once it has been reached, the filter is turned off and the rest of the optimization process is left solely to LSM. This ensures that the filter does not keep removing materials

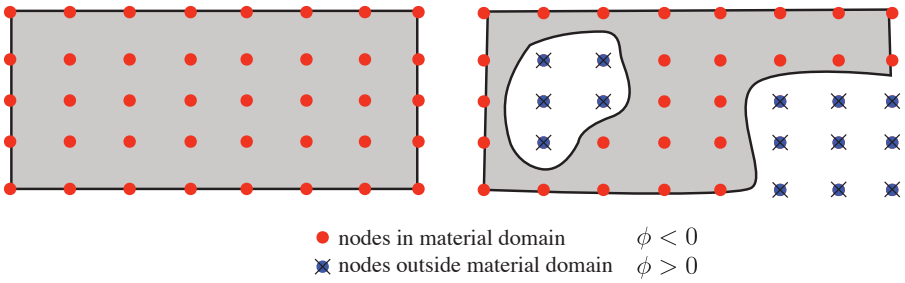


Figure 5.2: How the high-pass filter works

once the volume constraint has been achieved and allows LSM to smoothen out the material boundaries.

An example of how the high-pass filter works is shown in Figure 5.2. When the nodes with low strain energy are removed, new voids and boundaries are created in the design domain.

However, by using the high-pass filter, a new parameter called the filter cutoff ratio R is introduced. The determination of the filter cutoff ratio is the key to controlling the optimization process as it affects the amount of material removed per iteration as well as the speed of the optimization process. This will be evident in the numerical examples presented in Section 7.1.

The use of this filter is extremely straightforward. By inspecting the level set value at the node, it is easy to identify the nodes that are within the material domain, $\phi > 0$. By specifying the filter cutoff ratio R, the objective is to remove a percentage of the current material domain at time t by removing a certain number of the material nodes.

We use the ratio to create an array with size equal to the "number of nodes to remove" and populate that array with individual non-zero minimum strain values. The filter cutoff value will be the maximum value in that array; the last entry. Nodes with strain energy values less than the filter cutoff value will be removed. In essence, the filter cutoff ratio is used to obtain the filter cutoff value. However, the cutoff ratio does not necessarily imply

that we only remove the amount of material representative of that ratio. The ratio only guarantees that we remove at least that percentage of material at each iteration since some nodes might have the same strain energy values. Therefore a judicious choice for the ratio is prerequisite.

A drawback of using the filter is that reinitialization of the level set function is required at each iteration whenever the filter is in effect. This is because the filter removes material independent of the material boundaries. If reinitialization is not carried out, the level set function will fail to possess the signed distance property.

5.3.2 Physically removing elements

In level set problems, many schemes such as the narrow band scheme [19] have been proposed to speed up the calculation of spatial derivatives for the level set functions. However, when used in structural optimization, level set computations are dwarfed by FEM computations. Any reduction in the amount of computations required by FEM will significantly boost the speed of the optimization process. This section describes a numerical scheme aimed at reducing the amount of time required by FEM computations in level set based topology optimization.

Element classification

An algorithm for categorizing elements into three groups, interior, surface and narrowband, will be described here. A similar categorizing approach described in Belytschko et al.[24, 25] was used to create finite element models from implicit functions and to subdivide elements for quadrature.

Table 5.1 shows the procedure for classifying elements. Although at any one time there will be four different sets of elements; interior Ω_{int}, exterior Ω_{ext}, surface Ω_{sur} and narrow band Ω_{nb}; only three types of elements are explicitly marked:interior, surface and narrow band. As shown in Fig. 5.3, the design domain Ω is simply a union of the interior, exterior

Table 5.1: Procedure for classifying elements

- Loop over all elements

  $count = 0$

  Loop over all nodes in element

    if $(\phi_{node}) > 0$ $count = count + 1$

  if $(count > 0)$

    if $(count == 4) \Rightarrow$ Interior element

    if $(count <= 3) \Rightarrow$ Surface element

      Mark elements surrounding node as in narrow-band

- end loop

and surface sets.

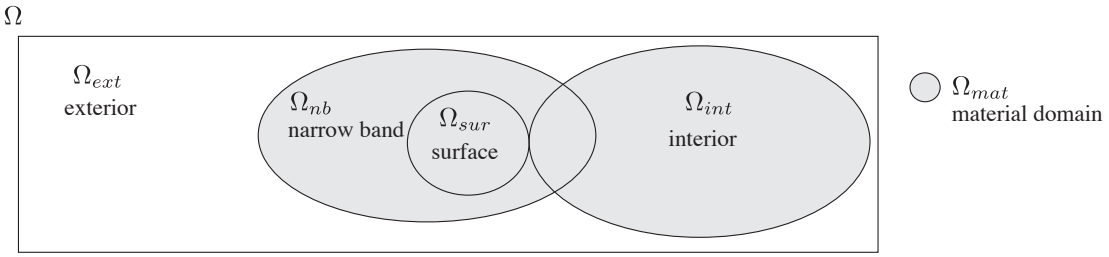


Figure 5.3: Element sets within a design domain

$$\Omega_{int} \cup \Omega_{ext} \cup \Omega_{sur} = \Omega \tag{5.14}$$

The set of surface elements Ω_{sur} is a subset of Ω_{nb}:

$$\Omega_{sur} \in \Omega_{nb} \tag{5.15}$$

From these categories, a material domain can be described two ways:

$$\Omega_{mat} = \Omega_{int} \cup \Omega_{sur} \tag{5.16}$$

$$\Omega_{mat} = \Omega_{int} \cup \Omega_{nb} \tag{5.17}$$

Since the smoothing length in Eq. (3.22) is usually set to one grid spacing, Eq. (5.17) is recommended for describing the material domain. Exterior elements are represented as a complement of the material domain.

$$\Omega_{ext} = \Omega_{mat}^{C} \tag{5.18}$$

An example of the different element sets is shown in Fig. 5.4. Although there are various methods to characterize a narrow band, a narrow band is usually defined by the number of nodes on either side of the zero level set[19]. For the sake of discussion in this paper, we shall call this the node model narrow band. This paper proposes defining a narrow band with elements instead of nodes. The narrow band in this case is defined by the surface elements

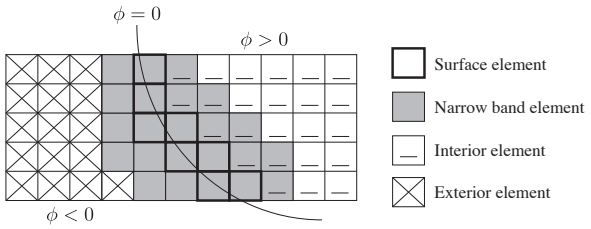


Figure 5.4: Element sets

and the surrounding elements of the individual nodes in each surface element. We shall call this the element model narrow band. While the node model narrow band is mainly used for reducing computational costs associated with level set calculations, such as update and re-initialization costs, the element model narrow band will be used to reduce computational costs associated with FEM calculations. This will be discussed in the next subsection.

Element and Node Renumbering

Once the elements have been categorized, we can begin the process of removing the exterior elements and their corresponding nodes from the optimization process. Two arrays of element numbers and node numbers are kept. A static list stores all the initial element and node numbers in the design domain Ω. The second list is termed the "current" list. It stores the renumbered element and node numbers of the material domain Ω_{mat}. The current list is created from the static list at the start of each iteration.

This differs from the element removal and reintroduction strategy proposed by Bruns and Tortorelli[46]. They remove elements by modifying the degree of freedom vector and number of degrees of freedom at each optimization iteration. Special treatment is required for the stiffness matrix and there is no classification and renumbering of elements involved. A significant advantage of our approach over that of Bruns and Tortorelli[46] is that we remove elements in the pre-FEM process stage: this allows any black box FEM routine to be used.

At each iteration, the algorithm detailed in Table 5.1 is run to categorize the elements.

The current list is then created for the material domain by using the set described in Eq. (5.17). An example of renumbering elements in the static list to the current list can be seen in Fig. 5.5. The nodes are renumbered accordingly. A simple renumbering structure similar

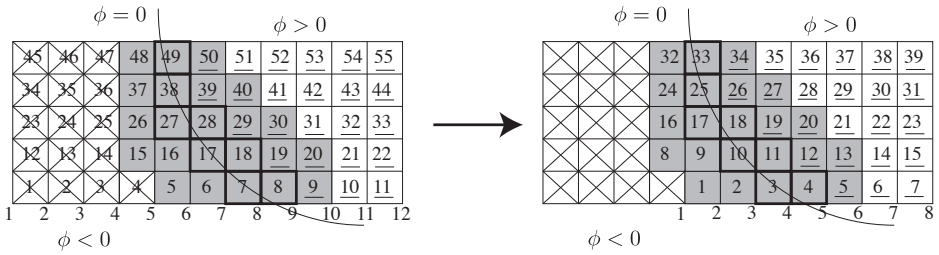


Figure 5.5: Before element renumbering (left) and after element renumbering (right)

to that shown in Fig. 5.5 can be used. On the other hand, a particular node renumbering scheme that is better suited to certain solvers can also be employed. For example, a Reverse Cuthill McKee (RCM) ordering scheme can be used to reduce the matrix bandwidth when using a sparse matrix solver. The renumbered elements and nodes in Ω_{matnb} are mapped to their respective coordinates in Ω. Connectivity for elements and nodes in Ω_{matnb} are also created. The set of renumbered elements, nodes and connectivity defining Ω_{matnb} can then be input to any black box FEM routine.

Since the velocity function defined in Eq. (4.42) is dependent on the dirac delta function shown in Eq. (3.23), the velocity function is only defined within one grid spacing (the smoothing length) of the zero level set. Even with the displacement field defined in Ω_{matnb}, we only need to compute the strain energy within one grid spacing of the zero level set in order to evolve it. The narrow band defined in Ω_{nb} is sufficient to ensure that the velocity function is properly represented. Since Ω_{nb} contains the zero level set, we can intentionally use more gauss points when performing gaussian integration within the narrow band for strain energy and volume. This ensures an accurate value for strain energy and volume. Note that only ϕ values at the nodes of the elements in Ω_{surf} need to be updated. For elements in Ω_{int}, calculating volume is trivial as the grid is structured. For strain energy

values in Ω_{int}, a low number of gauss points can be used in order to maximize computational efficiency.

Table 5.2: Procedure for evaluating strain energy

- Loop over all elements

  if element $\in \Omega_{matnb}$

  Integrate with more gauss points for strain energy and volume

  else if element $\in (\Omega_{int} \cap \Omega_{nb})^C$

  Integrate with fewer gauss points for strain energy

  Volume calculation is trivial

- end loop

Given a displacement field, the algorithm for evaluating strain energy is described in Table 5.2. When seen from a marco level, the process of categorizing elements for optimization and extracting the "current" list makes the entire level set topology optimization process appear adaptive.

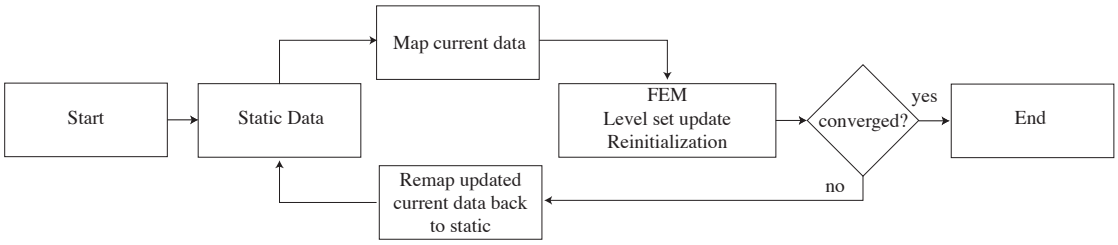


Figure 5.6: Proposed level set topology optimization flowchart

Chapter 6

Numerical Examples for Sizing & Shape Optimization

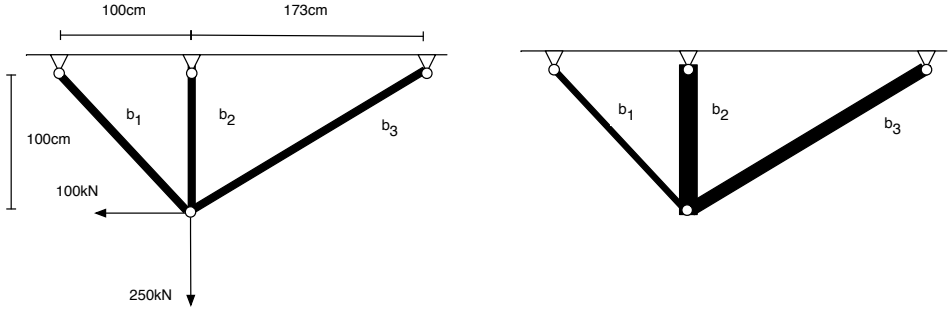
6.1 Sizing Optimization

6.1.1 Truss design

When performing sizing optimization on a truss design, the cross-sectional areas of each individual truss becomes a design variable. The design of a truss system can be described as a multi-objective design problem [47, 16] since it requires the consideration of structural metrics, such as stress, deformation etc., and other metrics, such as cost and aesthetics. This section will present the solution to two multi-objective truss optimization problems where the number of design variables are greater than the number of design objectives.

Three member truss

This section will discuss the solution to the three-bar truss shown in Fig. 6.1(a). Two objective variations of this problem have been solved by Koski[47] and Kim and de Weck[9]. The optimization problem to be solved here has three objectives instead. The cross sectional



(a) Three member truss structure (b) Final three member truss design

Figure 6.1: Problem configuration and final design

areas of each truss member are the design variables $[b_1, b_2, b_3]^T$ that have to be determined such that three design objectives, volume V, horizontal displacement of the load supporting node u_x and the vertical displacement of the load supporting node u_y are minimized. The three objectives are formulated as

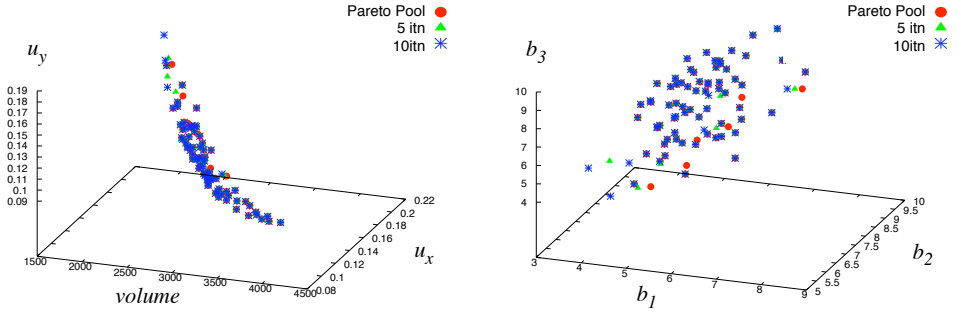
$$f_{volume} = 141.4b_1 + 100b_2 + 200b_3 \tag{6.1}$$

$$f_{horizontal\ displ} = u_x \tag{6.2}$$

$$f_{vertical\ displ} = u_y \tag{6.3}$$

The design variables were initialized within $5 \leq b_{1,2,3} \leq 10$. Ten update iterations were used for this problem and the pareto front is shown in Fig. 6.2. Figure 6.2 also shows the history of the pareto optimal solutions in the Pareto pool, and during the fifth and tenth update iterations. Figure 6.2(a) shows the Pareto optimal design solutions in three dimensional $volume - u_x - u_y$ objective space. Figure 6.2(b) shows their respective variables plotted in $b_1 - b_2 - b_3$ variable space. In Fig. 6.2(a), the smooth convex surface in objective space represents the Pareto front. Table 6.1 lists the configuration of the selected Pareto optimal design solution and its distance from the center-of-gravity solution. The final design was selected with $\beta_{min} = 0.53$, according to Eq. (4.21), has design variables $[b_1, b_2, b_3]^T = [5.63, 7.66, 7.28]^T$ with volume $3017cm^3$, $u_x = 0.11cm$ and $u_y = 0.12cm$. Figure 6.1(b) shows

the final design not drawn to scale, but emphasizing the difference in thickness between truss members.



(a) Objective space $volume - u_x - u_y$ (b) Variable space $b_1 - b_2 - b_3$

Figure 6.2: Objective and variable space for Three member truss design

Table 6.1: Final design configuration for three member truss

Designs	b_1	b_2	b_3	$volume$	$u_{2,x}$	$u_{2,y}$	β
1	5.63	7.66	7.28	3017	0.11	0.12	0.53
COG	5.66	7.62	7.81	3122	0.11	0.12	0

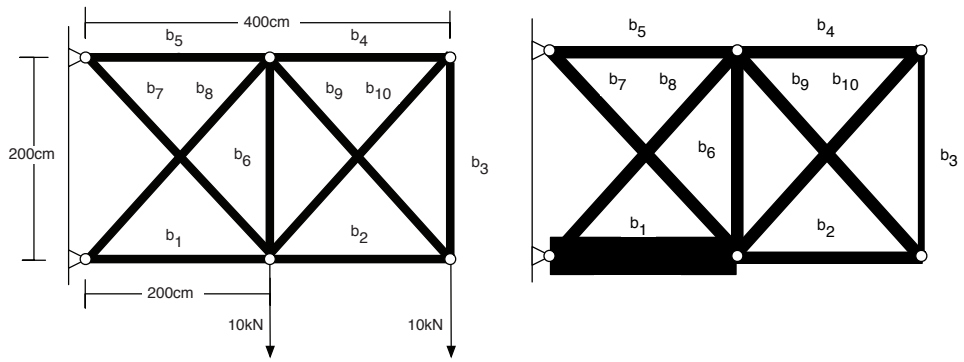
Ten member truss

This section will discuss the solution to the ten-bar truss shown in Fig. 6.3(a). The cross sectional areas of each truss member are the design variables $[b_1, ..., b_{10}]^T$ that have to be determined such that three design objectives, volume V, vertical displacement of both load supporting nodes u_2 and u_3 are minimized. The three objectives are formulated as

$$f_{volume} = 200 \sum_{i=1}^{6} b_i + 282.8 \sum_{i=7}^{10} b_i \tag{6.4}$$

$$f_{vertical\ displ} = u_{2,y} \tag{6.5}$$

$$f_{vertical\ displ} = u_{3,y} \tag{6.6}$$



(a) Ten member truss structure (b) Final ten member truss design

Figure 6.3: Problem configuration and final design

200 design variables were initialized within $5 \leq b_{1,...,10} \leq 10$. Twenty gradient update iterations were used to solve this problem. Figure 6.4 shows the Pareto front and the initial Pareto pool solutions in three dimensional $volume - u_2 - u_3$ objective space. The shift of the initial Pareto pool solutions down the Pareto front shows that the optimization process is working to minimize all three objectives.

Tables 6.2 and 6.3 compares the design solution with $\beta_{min} = 5.47$ and the center-of-gravity solution. When compared to the center-of-gravity solution, the selected design solution had reduced $volume$ and $u_{2,y}$, but a slightly higher value for $u_{3,y}$. It has design variables $[b_1, b_2, ..., b_{10}]^T = [70.01, 6.69, ..., 7.41]^T$ with volume $29059cm^3$, $u_{2,y} = 0.007cm$ and $u_{3,y} = 0.015cm$. Figure 6.3(b) shows the final design not drawn to scale, but emphasizing the difference in thickness between truss members. The result is reasonable from a mechanical point of view. Interestingly, as shown in Fig. 6.3(b), the cross sectional area b_1 seems to be the decisive force in the design due to the generous amount of cross-sectional area allocated to it.

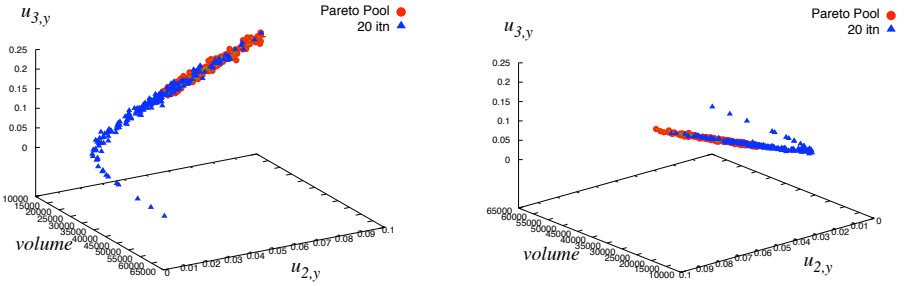


Figure 6.4: Objective space for ten member truss design

Table 6.2: Final design configuration for ten member truss: design variables

Designs	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Selected	70.01	6.69	4.36	8.26	9.45	8.24	5.92	5.06	8.68	7.41
COG	70.77	7.48	6.42	7.640	7.53	7.53	7.52	7.29	7.63	7.53

6.1.2 Parametric design

This section will present design solutions for a portable notebook computer. Due to advances in nano technology, notebook computers have been gaining computational strength and becoming evermore portable. The practicality of this design lies in the consideration of multidisciplinary, multi-objective design objectives. This design involves a total of four design parameters, four design objectives and five constraints. Design parameters are shown in Figure 6.5 and presented in Equation 6.8. Design objectives are shown in Equations 6.9, 6.10, 6.11 and 6.12, The screen size b_4 is a function of b_1 and b_2, and can also be represented by

$$b_4 = \frac{\sqrt{b_1^2 + b_2^2}}{2.54} \tag{6.7}$$

which is the display size in inches. Objective Equation 6.9 physically represents volume. However this is proportional to performance as large notebook computers normally are performance orientated. Thickness objective Equation 6.10 simply represents the desire to

Table 6.3: Final design configuration for ten member truss: design objectives

Designs	*volume*	$u_{2,y}$	$u_{3,y}$	β
Selected	29059	0.007	0.015	5.47
COG	29947	0.05	0.1	0

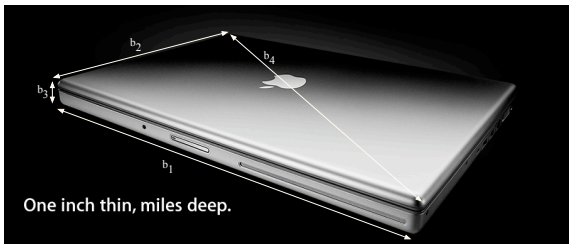


Figure 6.5: Computer model with four design parameters

minimize thickness, b_2, subjected to a physical constraint of $2cm$ minimum. The formulation of objective Equation 6.11 was the result of statistically recording consumer preferences when it came to a preferred screen size. Objective Equation 6.12 represents the cost function which is a function of the screen size b_4. The minimization of Equations 6.9, 6.10, 6.11 and 6.12 leads to a design which has maximized performance, minimized thickness, a popular screen size and minimized cost respectively. The constraints considered in this design include functional and dimensional constraints and are shown in Equations 6.13, 6.14, 6.15, 6.16 and 6.17.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} length \\ width \\ thickness \\ screen\ size \end{bmatrix} \tag{6.8}$$

$$f_{performance} = -b_1 b_2 b_3 \tag{6.9}$$

$$f_{thickness} = (b_3 - 2)^2 \tag{6.10}$$

$$f_{size\ preference} = 3.1667 b_4^2 - 88.833 b_4 + 585 \tag{6.11}$$

$$f_{cost} = 36.667 b_4^2 - 823.33 b_4 + 6099 \tag{6.12}$$

$$g_1 : 1000 + f_{performance} \leq 0 \tag{6.13}$$

$$g_2 : f_{cost} - 2700 \leq 0 \tag{6.14}$$

$$g_3 : 14 - b_2 \leq 0 \tag{6.15}$$

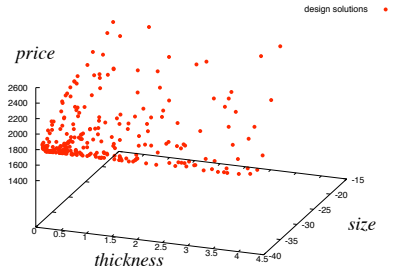
$$g_4 : b_3 - 4 \leq 0 \tag{6.16}$$

$$g_5 : 9 b_1 - 16 b_2 \leq 0 \tag{6.17}$$

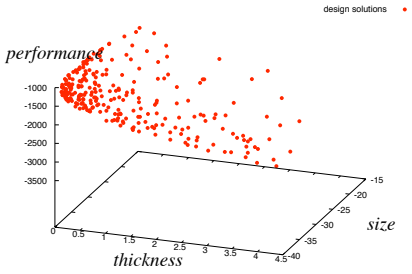
The expected solution for the objective space of this problem is in four dimensions. As it is impossible to plot a four dimensional graph of its results, four seperate three dimensional plots are shown instead in Figure 6.6. A clear Pareto front can be observed in each of the four three-dimensional plots shown in Figure 6.6. Table 6.4 lists five Pareto optimal design solutions consisting of β_{max} and β_{min} solutions. The final solution selected at $\beta_{min} = 0.60$ is described by $\mathbf{b} = [28.91, 20.98, 2.45]^T$ with $f_{cost} = 1772$ and $\Theta = 14$. Because standard units were used, this result corresponds to a design that has a display of 14in, thickness 2.45cm and costs $1772. In an interesting note, the 1 inch thickness of the final design corresponds well to the advertisement for the macbook in Figure 6.5.

6.2 Shape Optimization

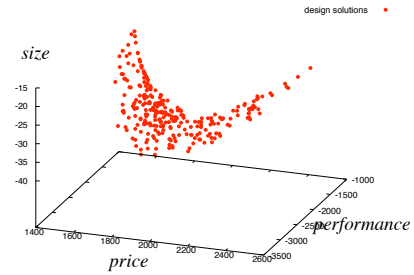
The shape optimization problem that will be discussed in this section has been solved by Shimoda et al[48] with a different approach. Instead of the traction and weight methods



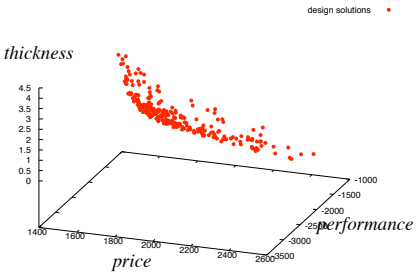
(a) $thickness - size - price$



(b) $thickness - size - performance$



(c) $price - performance - size$



(d) $price - performance - thickness$

Figure 6.6: Objective space solutions for notebook computer design

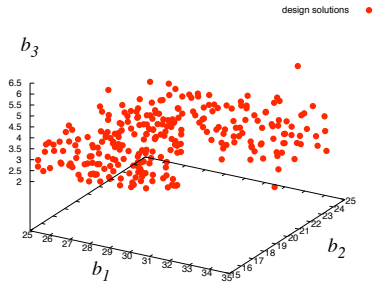
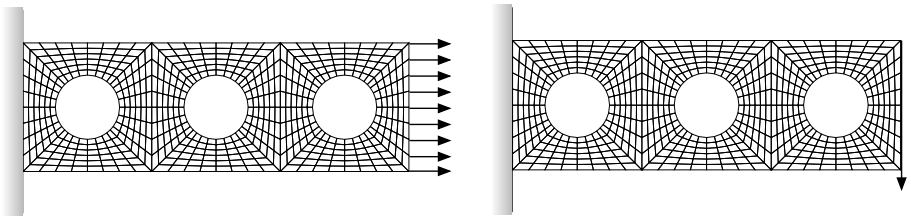


Figure 6.7: Variable space solutions for notebook computer design

Designs	b_1	b_2	b_3	$\|performance\|$	$thickness$	$size$	$cost$	β
1	25.08	16.15	2.99	1212	0.98	11.74	1487	5.69
2	27.09	22.09	2.40	1436	0.16	13.76	1712	2.28
3	25.57	18.03	2.17	1000	0.03	12.31	1520	4.06
4	28.91	20.99	2.45	1487	0.20	14.06	1772	0.60
5	34.45	24.35	2.09	1753	0.01	16.61	2540	6.94
COG	28.71	20.53	2.79	1645	0.625	13.90	1739	0

Table 6.4: Design solutions for portable notebook computer



(a) Tension Loading (b) Shear Loading

Figure 6.8: Continuum Beam Model and Loading Conditions

used by Shimoda et al.[48], this paper uses a combination of the adjoint variable method, the discrete force method and the real valued function to solve this multi-objective problem. The adjoint variable method is used to keep computational costs down when calculating shape sensitivities, the discrete force method provides smoothing and the advantage of using the real valued function is that no explicit selection of weights for the objective functions are necessary. The configuration of the continuum beam model to be optimized is shown in Figure 6.8. It has 654 nodes and 576 elements; the position of the nodes are the design variables. The optimization problem is to minimize the respective averaged compliance due to tension loading $\mathbf{F}_t$, Figure 6.8(a), and shear loading $\mathbf{F}_s$, Figure 6.8(b), subjected to an

equality volume constraint .

$$f_1(\mathbf{b}) = \int_{\tilde{A}} \mathbf{F}_t^T \mathbf{u}_t d\tilde{A} \tag{6.18}$$

$$f_2(\mathbf{b}) = \int_{\tilde{A}} \mathbf{F}_s^T \mathbf{u}_s d\tilde{A} \tag{6.19}$$

$$h(\mathbf{b}) - V_0 = 0 \tag{6.20}$$

Equations (6.18) and (6.19) represent the objective functions for the respective compliances due to tension loading and shear loading, where $\tilde{A}$ represents the loading surface. The length of the beam is $0.6m$ with a width of $0.2m$. The magnitude of the applied tension and shear forces is 10kN. Young's modulus and Poisson's ratio are 206GPa and 0.3 respectively.

By defining initial voids in the continuum to be oval, random initialization is carried out by randomizing the short-radius and having the long-radius follow suit such that the volume constraint is always satisfied. Figure 6.9 shows the Pareto front after optimization for 200 Pareto optimal solutions. Although the convergence rate for each point was different, some points converged faster than others, a total of 400 iterations were required to generate this Pareto front. The $x - y$ axes represent the objective values normalized against the "design sample" shape configuration shown earlier in Fig. 6.8. The design sample is represented in Fig. 6.9 at the point $(1, 1)$. Figure 6.10 shows three different Pareto optimal shape solutions taken from the Pareto front. Figures 6.10(a) and 6.10(b) show shapes at the extreme ends of the Pareto front where the respective loading conditions dominate. Suppose an engineer has a sample shape which has the same configuration as shown earlier in Fig. (6.8). With reference to this design sample, we can normalize the average compliances of the Pareto optimal solutions with the respective average compliances of the design sample. Cases when the normalized compliance is larger than unit value should be ignored as they do not represent reasonable improvements over the design sample. As the design variables were nodal positions, the COG method, detailed in Section 4.2.4, was not used to select a final solution. However, Fig. 6.10(c) shows a moderate design with normalized values of $(0.88, 0.96)$; against the design sample, it represents a 12% improvement when in tension

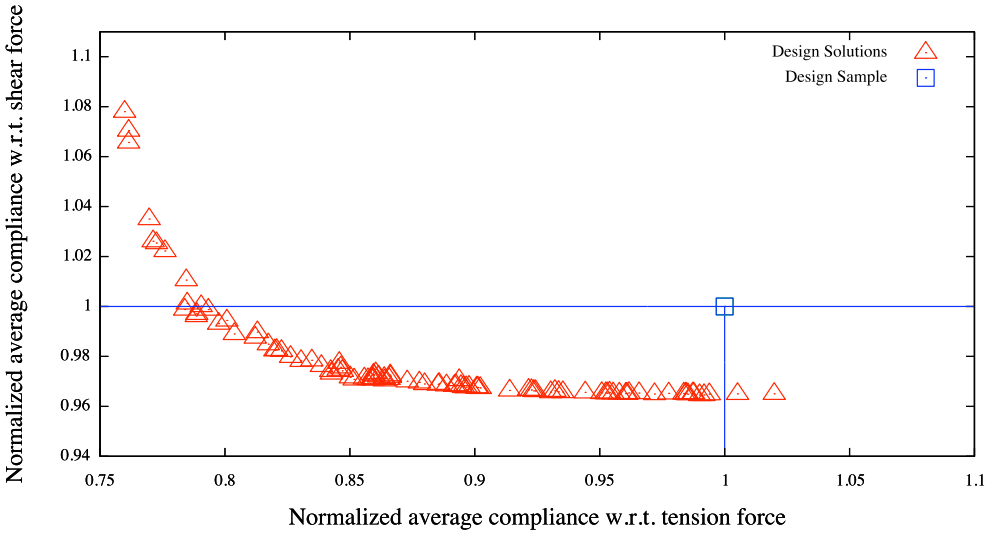
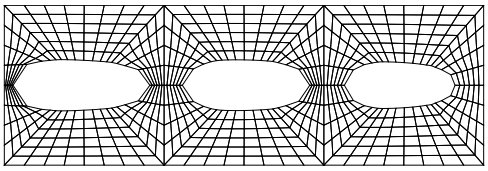
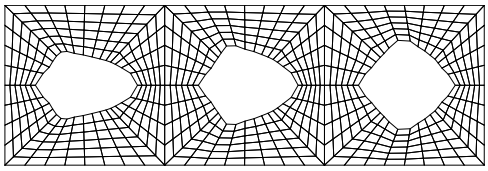


Figure 6.9: Pareto Front

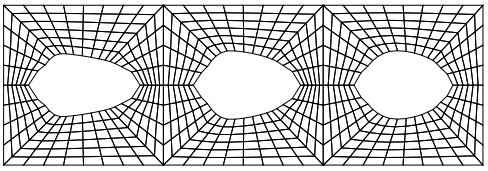
and a 4% improvement when in shear. By inspecting the Pareto front shown in Fig. (6.9), we can see that there is much less improvement for shear loading than there is for tension loading; this is theoretically reasonable. In terms of reliability, the results obtained here are comparable with Shimoda et al.'s results[48].



(a) Design 1: optimized for tension loads



(b) Design 2: optimized for shear loads



(c) Design 3 optimized for tension and shear loads

Figure 6.10: Different Pareto optimal shapes with respective average compliance values

Chapter 7

Numerical Examples for Topology Optimization

7.1 Utilizing a high-pass filter

In this section, numerical results from utilizing the high-pass filter described in Section 5.3.1 are presented. Two different design domains are considered in the examples: a square design domain and a rectangular design domain. A few factors are considered here. Basically we want to compare the effect that different filter cutoff ratios have on the final topology when given a full design domain, an initial configuration with no zero level sets. We also compare this effect against the case when not using the filter, but given initial design domains with varying material boundaries, initial configurations contain zero level sets. Convergence speed between using the high-pass filter and various initial level sets is also compared.

Two different cases are considered in the examples presented here.

Case 1: High-pass filter is utilized, a full material domain with no material boundaries (no initial zero level sets) is used.

Case 2: High-pass filter is not utilized, a material domain with material boundaries (initial

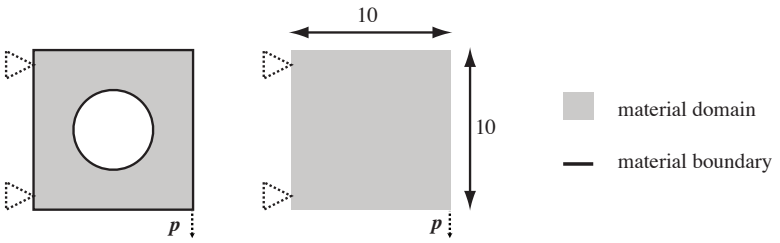


Figure 7.1: Two problems with different initial conditions

zero level sets) is used instead.

The problem statement for the examples shown in this section is to minimize strain energy density subject to equilibrium constraints and volumetric constraints:

$$\text{minimize} \quad f(\phi) = \frac{1}{2}\int_{\Omega} H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} d\Omega \tag{7.1}$$

$$\text{s.t.} \quad \nabla \cdot (H(\phi) E \boldsymbol{\sigma}) = \boldsymbol{p} \tag{7.2}$$

$$\int_{\Omega} H(\phi) d\Omega = V_0 \tag{7.3}$$

For the examples shown in this section, Young's Modulus, Poisson's ratio and loading were set to E =20GPa, $\nu = 0.3$ and $\boldsymbol{p}$ =50N respectively. Volume constraints were set to be 35% of the total design domain. A grid spacing of 0.2 was used for every example presented here.

7.1.1 Example 1: Square design domain

This example is based on a simple cantilever supported at one end and loaded at the opposite end. They are briefly depicted in Figure 7.1.

Since it was established earlier in Section 5.3.1 that different initial level sets often lead to different final solutions with varying convergence rates, we wish to select a favorable initial level set. This problem is quite simple and most initial level sets with a little complexity will more often than not, arrive at the same solution. For case two when the high-pass filter is

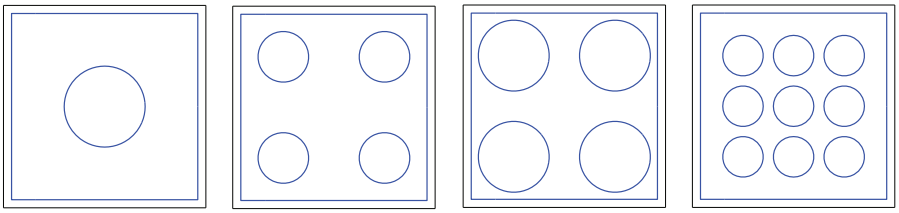


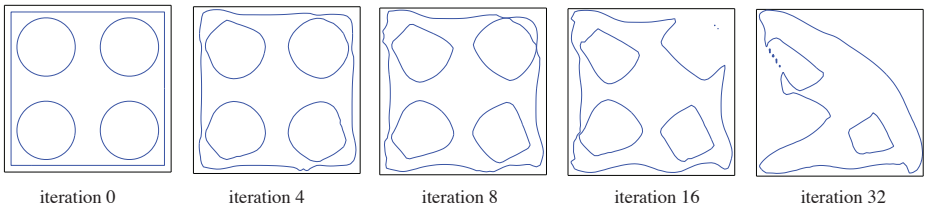
Figure 7.2: Four different initial level sets used

not utilized, a favorable initial level set would converge quickly. Four different initial level set functions shown in Figure 7.2 were considered in order to select one that converges quickly. In this case, the third configuration, with four relatively large holes, from the left in Figure 7.2 converged the quickest. This will also be evident later in Figures 7.5 and 7.6.

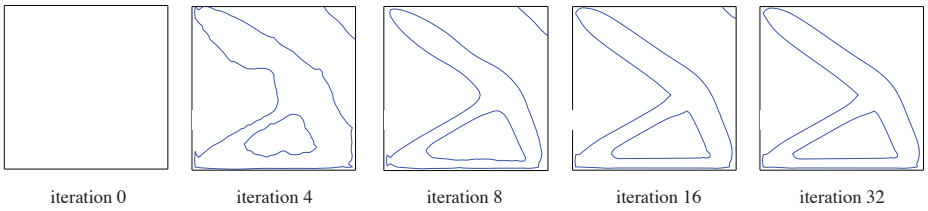
Figure 7.3 shows a "iteration-by-iteration" comparison between case one and case two. It is very clear that higher convergence rates can be obtained when the high-pass filter is utilized. When a high-pass filter with a cutoff ratio of 0.25 is utilized for this example, a glimpse of the final solution can already be obtained very early in the evolution process at the fourth iteration.

Figure 7.4(a) shows the final solutions obtained from the four different initial level sets shown in Figure 7.3. Except for the leftmost configuration, the remaining configurations gave similar topologies. Figure 7.4(b) shows final solutions obtained by eight different high-pass filter cutoff ratios: $0.05 \sim 0.40$ with 0.05 intervals. Except for the case when a low cutoff ratio of 0.05 was used, every other cutoff ratio returned the same topology.

Compliance and volume time histories for the four different conditions shown in Figure 7.4(a) and the eight different conditions shown in Figure 7.4(b) are plotted in Figures 7.5 and 7.6 respectively. Comparing Figures 7.5(a) and 7.5(b), it can be seen that although final compliance values are almost identical, the nature of the time histories are quite different. Figure 7.5(a), for example, shows a time history moving from high compliance to low compliance. In contrast, Figure 7.5(b) shows a time history moving from low compliance to



(a) Level set evolution with initial voids specified and high-pass filter not utilized

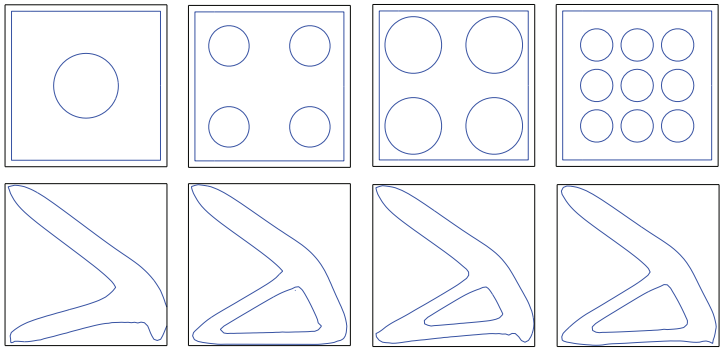


(b) Level set evolution with high-pass filter cutoff 0.25

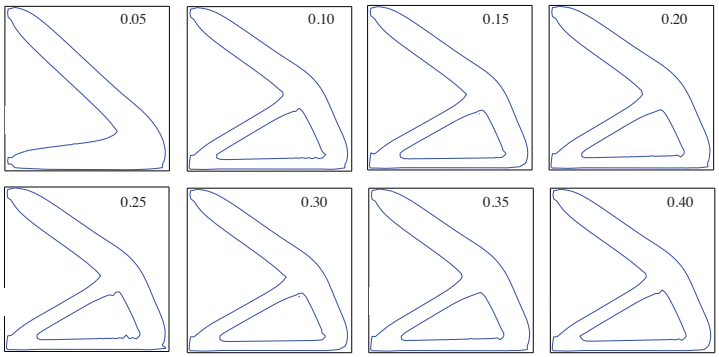
Figure 7.3: Comparing number of iterations

high compliance. This is due to the fact that a full material domain is used and minimal compliance is guaranteed at the start of the optimization process.

Figure 7.6(b) shows the volume time histories for the twelve different conditions described earlier. When comparing Figures 7.6(a) and 7.6(b), it is evident that when the high-pass filter is utilized, the volume constraint is reached very quickly; in most cases, within 10 iterations. In contrast, Figure 7.6(a) shows a longer period before the volume constraint is reached. The trend in Figure 7.6(b) supports the notion that the time required for achieving the volume constraint is reduced when a large cutoff ratio is used.

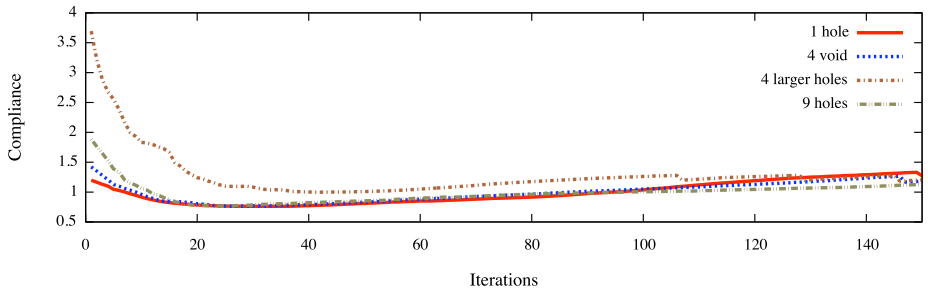


(a) Effect of initial level set configuration on final topology

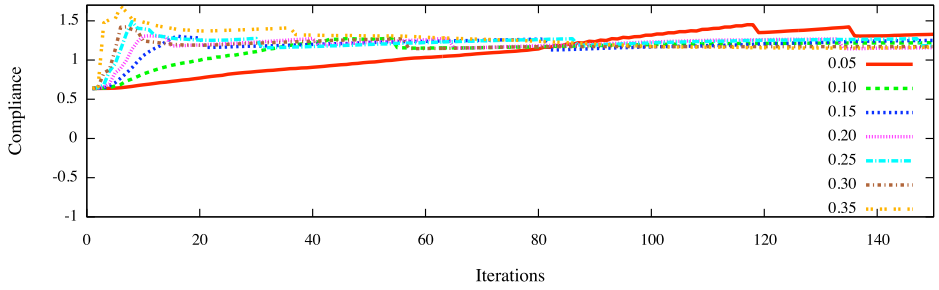


(b) Effect of filter cutoff ratio on final topology

Figure 7.4: Effect of different initial configurations and filter cutoff ratios on final topology

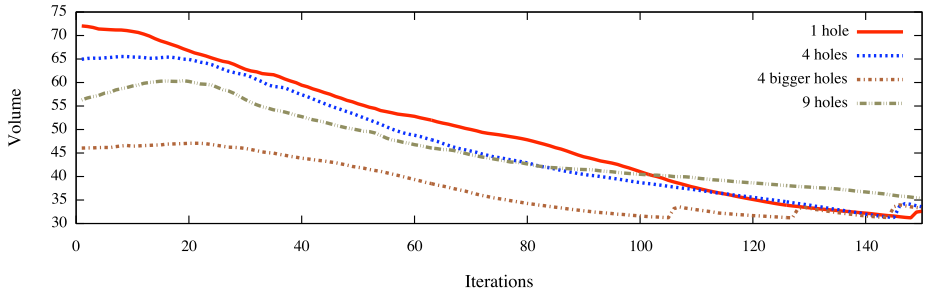


(a) Compliance history when using different initial level sets

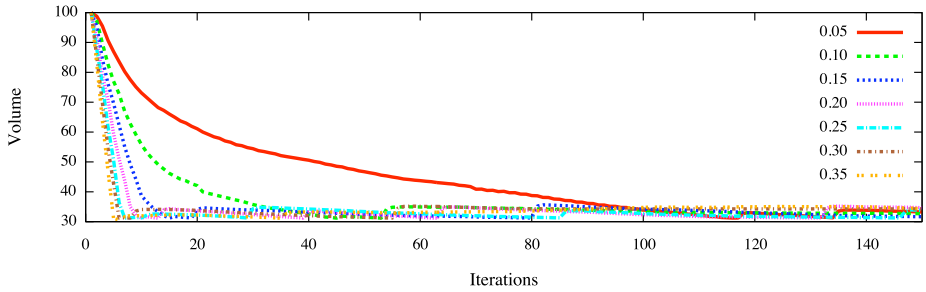


(b) Compliance history when using different filter cutoff ratios

Figure 7.5: Comparing compliance histories



(a) Volume history when using different initial level sets



(b) Volume history when using different filter cutoff ratios

Figure 7.6: Comparing volume histories

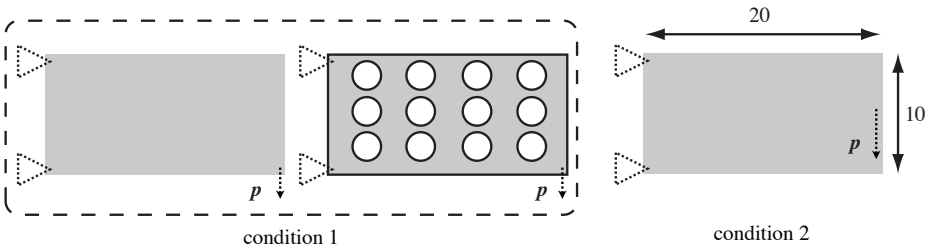


Figure 7.7: Two loading conditions considered

7.1.2 Example 2: Rectangular design domain

As with the previous example, the examples here are also based on a simple cantilever supported at one end and loaded at the opposite end. The only differences are the size of the design domain and two loading conditions are considered. The examples presented here have a design domain twice the length but of the same width as shown in Figure 7.7.

Loading condition 1

For case one, when an initial level set with twelve voids is subjected to loading condition 1, the level set evolution in 50 iterations can be seen in Figure 7.8. It is very clear that the optimization process has not converged to a solution. The final topology after 300 iterations is shown later in Figure 7.14(a). Figure 7.9 shows the compliance and volume histories over 300 iterations for case one. In this case, the volume constraint is achieved within the 150th iteration. The final topology had a minimum compliance value of approximately 4.5.

For case two, two different filter cutoff ratios were considered: 0.1 and 0.2. When a filter cutoff ratio of 0.01 was used, the level set evolution can be seen in Figure 7.10. In contrast to the level set evolution in Figure 7.8, a clear final topology can be observed after 50 iterations. The respective compliance and volume time histories over 300 iterations for case two with a filter cutoff ratio of 0.1 are shown in Figure 7.11. The final topology had four voids with a minimum compliance value of approximately 4.5.

The same 50 iterations for a filter cutoff ratio of 0.2 are plotted in Figure 7.12. Although

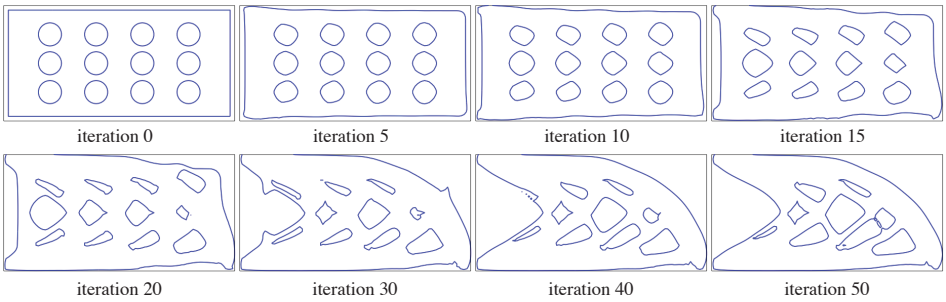


Figure 7.8: Level set evolution for loading condition 1

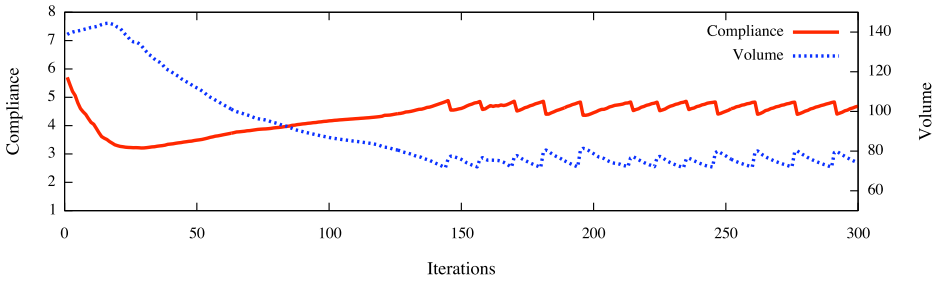


Figure 7.9: Compliance and volume vs. iterations for initial level set with 12 voids

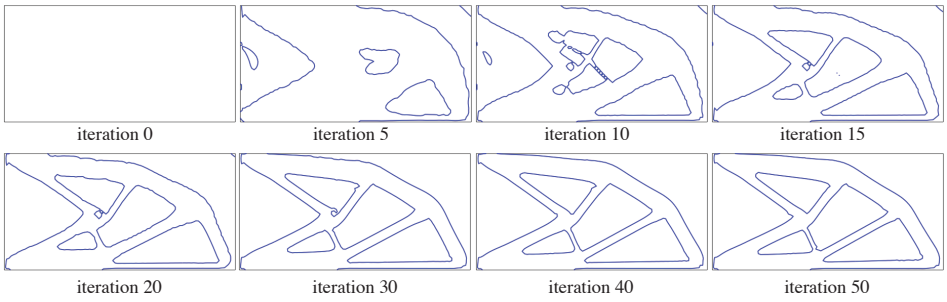


Figure 7.10: Level set evolution for loading condition 1 with filter cutoff 0.1

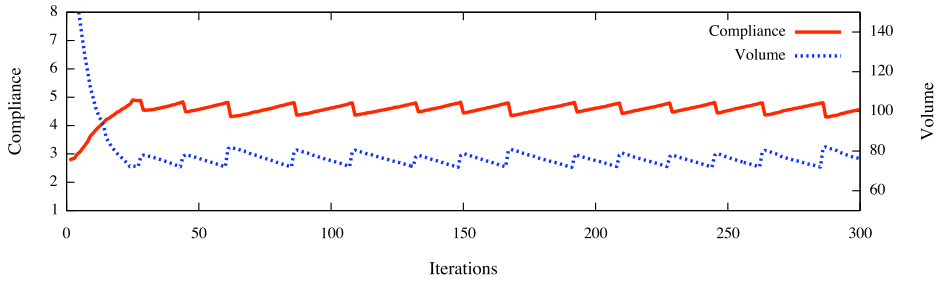


Figure 7.11: Compliance and volume vs. iterations with filter cutoff 0.1

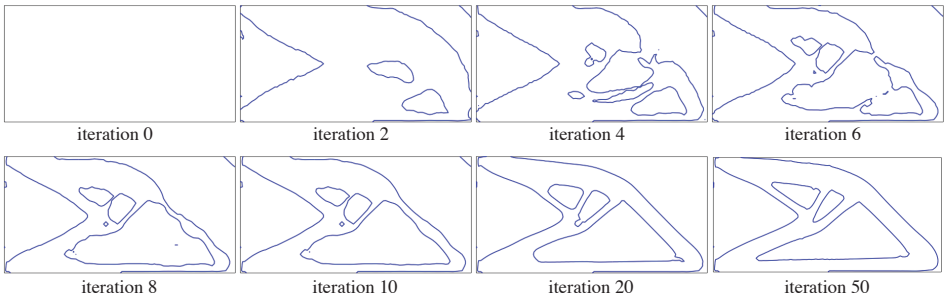


Figure 7.12: Level set evolution for loading condition 1 with filter cutoff 0.2

in this instance the optimization process resulted in a different topology, a clear topology can still be observed after 50 iterations. The final topology shown in Figure 7.12 has a total of three voids: one large void and two small voids. Figure 7.13 shows the compliance and volume histories over 300 iterations for case two with a filter cutoff ratio of 0.2. Due to the larger filter cutoff ratio, the volume constraint was achieved faster in Figure 7.13 when compared against Figure 7.11. In terms of the compliance time history, Figure 7.13 shows two high peaks with a compliance value of eight before eventually oscillating at about 4.5. In contrast, Figure 7.11 shows a smoother compliance time history with no sudden peaks.

Figure 7.14 shows the final topologies of all three examples considered here: Figure 7.14(a) shows the final topology for case 1, Figures 7.14(b) and 7.14(c) show the final topologies for case 2 when filter cutoff ratios of 0.1 and 0.2 were used respectively.

Compliance and volume time histories of all three examples were plotted together in

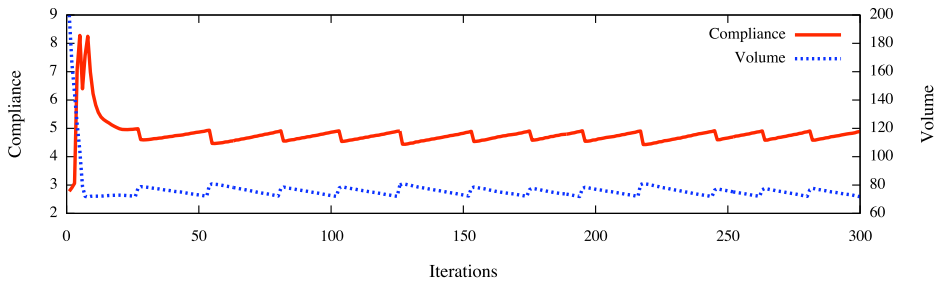
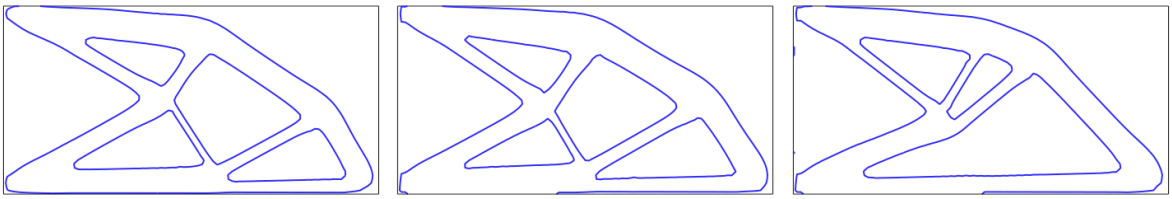


Figure 7.13: Compliance and volume vs. iterations with filter cutoff 0.2

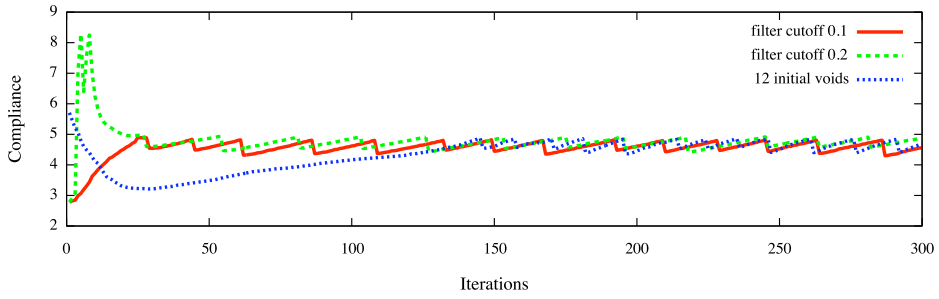


(a) Filter not used (b) Filter cutoff ratio 0.1 (c) Filter cutoff ratio 0.2

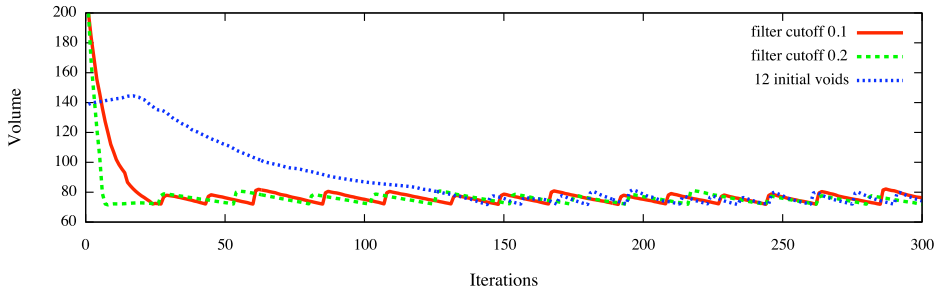
Figure 7.14: Final topologies (after 300 iterations) for 3 different problem parameters

Figure 7.15 for a visual comparison. Both Figures 7.15(a) and 7.15(b) show that instances when the filter was used resulted in faster convergence; a difference of approximately 125 iterations. In this particular example, Figures 7.10 and 7.12 provide a visual confirmation that the instances when the high-pass filter was used has indeed converged before the fiftieth iteration, whereas Figure 7.8 shows that the instance when the high-pass filter was not used has not converged at that point in time.

Despite some slight differences, the topologies in Figures 7.14(a) and 7.14(b) are quite similar with comparable compliance values. On the other hand, although possessing similar compliance values, the topology shown in Figure 7.14(c) is significantly different compared to Figures 7.14(a) and 7.14(b). This suggests that the problem might possess a flat solution space where there exist many different possible topologies.



(a) Compliance history



(b) Volume history

Figure 7.15: Compliance and volume time histories for 3 different problem parameters

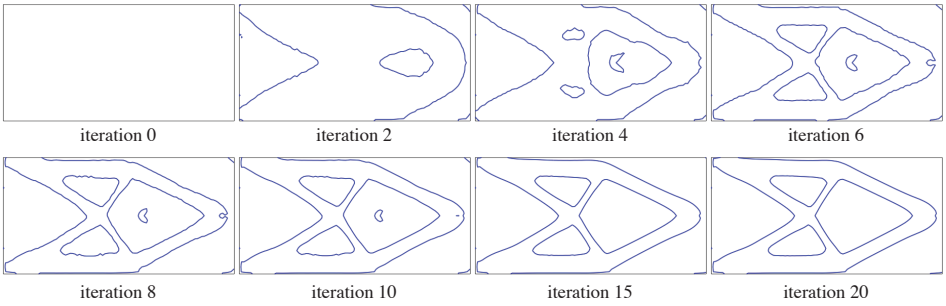


Figure 7.16: Level set evolution for loading condition 2 with filter cutoff 0.2

Loading condition 2

This section presents the solution to loading condition 2 shown in Figure 7.7. The difference between this example and the previous example is the position of the loading condition. In this instance, the loading condition is positioned midway opposite the boundary conditions and a filter cutoff ratio of 0.2 is used. Unlike previous results, the final topology here is expected to be symmetric.

Figure 7.16 shows the level set evolution of the optimization process and Figure 7.17 shows the respective compliance and volume time histories over 120 time iterations. From Figure 7.16, it can be seen that the topology has clearly converged around the fifteenth iteration; there are no significant differences between the topology in the fifteenth and twentieth iterations. For a problem used extensively in benchmarking [24, 32] convergence in fifteen iterations is extremely fast.

7.2 Removing elements void elements

In this section two examples are presented: a cantilever beam and a cantilever beam with a fixed void. The configurations for the the numerical examples presented here are shown in Figure 7.18 where $\boldsymbol{p} = 50$N. Young's Modulus and Poisson's ratio were set to 20GPa and 0.3 respectively. The examples were solved on a Core2 Duo MacBook 2.4Ghz with 4GB RAM.

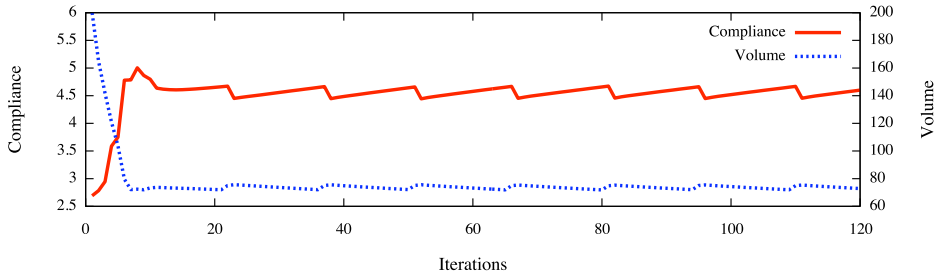
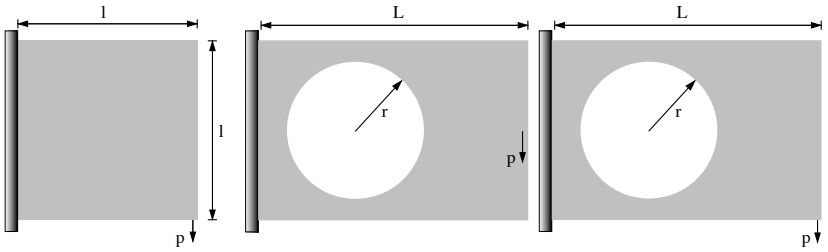


Figure 7.17: Compliance and volume time histories for loading condition 2 with filter cutoff 0.2

Although the computational time shown in the following figures are hardware dependent, the ratio determining the amount of time reduced is less biased.



(a) Example 1 (b) Example 2(A) (c) Example 2(B)

Figure 7.18: Example configurations: Cantilever beam and cantilever beam with fixed hole

Examples were solved with 4-node quadrilateral structured grids. In elements belonging to Ω_{nb}, 5×5 quadrature was used to evaluate for both strain energy and volume. Volume for elements in Ω_{int} is simply just the area of the element.

7.2.1 Example 1: Cantilever Beam

A cantilever beam with the load applied on the bottom right corner is considered in this example. The problem configuration is shown in Figure 7.18(a) where l=10.0. The example is solved twice on two different grid resolutions: 101×101 and 201×201 nodes. The volume constraint is set to 30% of the total volume in the design domain Ω.

101×101 nodes

Figure 7.19 shows the time history of the level set and the material domain defined by Ω_{matnb}. The compliance and volume time histories are shown in Figure 7.20. The total number of

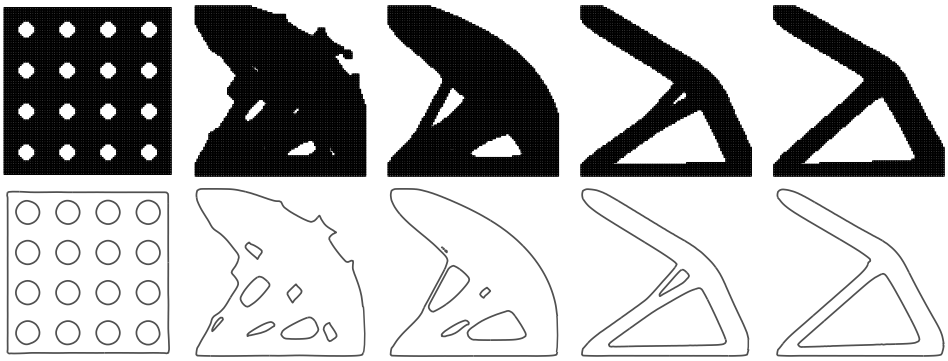


Figure 7.19: Level set (bottom) and Ω_{matnb} domain (top) time history for 101×101 grid

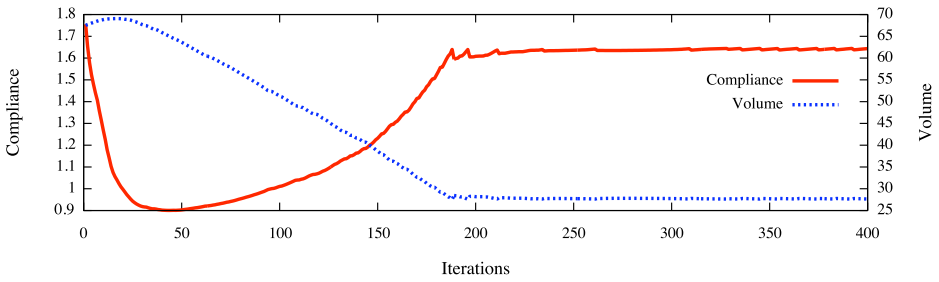


Figure 7.20: Compliance and volume time histories for 101×101 grid

degrees of freedom (DOF) in Ω_{matnb} and the number of elements in Ω_{nb} at each iteration are shown in Figure 7.21. The number of DOF at iteration zero shows the total number of DOF for the entire grid: 20402. Without element removal, the total number of DOF at each iteration would be 20402 and the average time taken for FEM calculation would be 6.75 seconds per iteration. With element removal, the total number of DOF decreases over the course of the optimization process as shown in Figure 7.21. Figure 7.22 shows the CPU time required at each iteration of the optimization process when elements are removed: this time

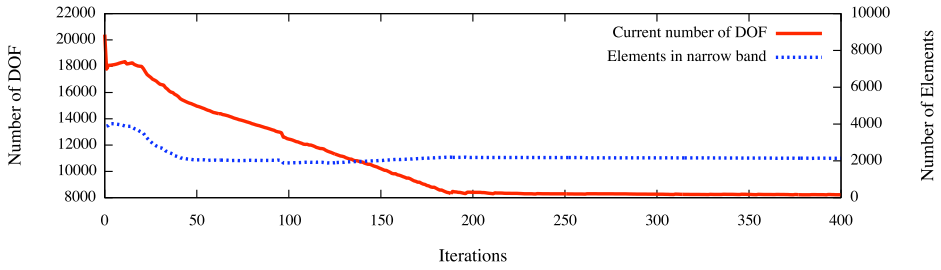


Figure 7.21: DOF and elements in narrow band time histories for 101×101 grid

includes the element removal procedure as well as time taken by FEM. The amount of time reduced in this example is summarized in Table 7.1. By removing the elements from the

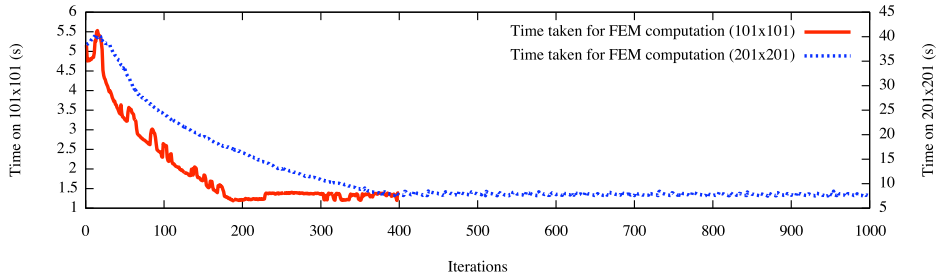


Figure 7.22: Time taken each iteration when elements are physically removed

FEM equations, we could reduce the computation time by about 70% over the course of 400 iterations taken to reach convergence. When the volume constraint was reached, the average time taken was 1.28 seconds per iteration: a 81% reduction compared to 6.75 seconds per iteration. These results are also summarized in Tables 7.1 and 7.2 respectively.

201×201 nodes

Although the final topology was similar to the previous example's, this example with a finer grid is used for speed comparison. The compliance and volume time histories, total number of DOF in Ω_{matnb} and the number of elements in Ω_{nb} are shown in Figures 7.23 and 7.24 respectively. Since the final topology is similar to the previous example's, the compliance

value shown in Figure 7.23 is also similar to the value shown in Figure 7.20. Without

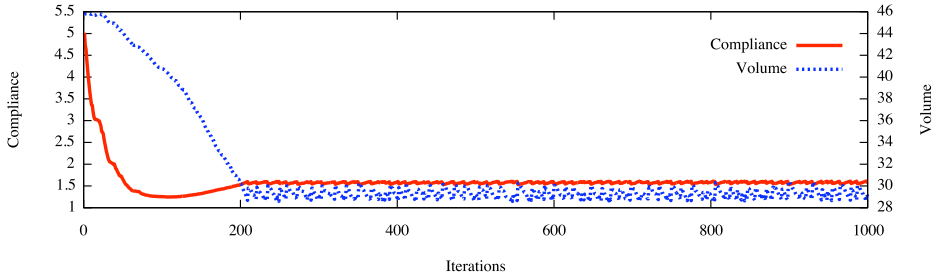


Figure 7.23: Compliance and volume time histories for 201×201 grid

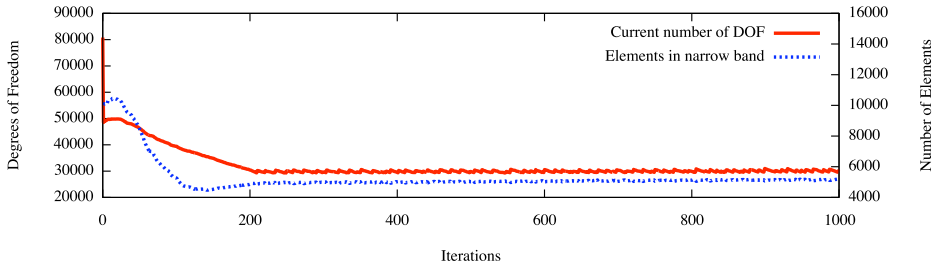


Figure 7.24: DOF and elements in narrow band time histories for 201×201 grid

element removal, the total number of DOF at each iteration would be 80802 (shown at iteration zero in Figure 7.24) and the average time taken for FEM calculation would be 60.26 seconds per iteration. With element removal, the total number of DOF decreases over the course of the optimization process as shown in Figure 7.24. The corresponding CPU time required at each iteration of the optimization process when elements were removed is shown in Fig. 7.22. The computation time was reduced by about 80% over the course of 1000 iterations taken to reach convergence. When the volume constraint was reached, the average time taken was 7.75 seconds per iteration: a 80% reduction compared to 60.26 seconds per iteration. These results are also summarized in Tables 7.1 and 7.2 respectively.

Table 7.1: Total time required

	DOF	No element removal†(s)	Element removal†(s)	% time reduced
Cantilever 101	20402	2701.93	798.32	70.45
Cantilever 201	80802	60260.00	12181.30	79.78

†CPU time on MacBook 2.4GHz Intel Core 2 Duo

Table 7.2: Time required per iteration after volume constraint is reached

	DOF	No element removal†(s)	Element removal†(s)	% time reduced
Cantilever 101	20402	6.75	1.28	81.03
Cantilever 201	80802	60.26	7.75	87.14

†CPU time on MacBook 2.4GHz Intel Core 2 Duo

7.2.2 Example 2: Cantilever with fixed hole

A cantilever beam with a fixed hole subjected to two different loading conditions are considered here. This example was previously studied by Bendsoe and Sigmund[17], and Belytschko et al.[25] where L=9.0, l=6.0, r=2.0 with its center at (3.0,3.0) and the volume constraint was set to 45%. Unlike the previous examples which show elements being removed adaptively, this example presents a case when certain elements are nonexistent during the optimization process: elements in the fixed void. In this section, loading case A refers to the loading condition described in Figure 7.18(b) and loading case B refers to the loading condition described in Figure 7.18(c). In both examples, a tighter volume constraint of 30% of the design domain Ω was set and a grid spacing of 0.05 was used.

Loading case A

For loading case A, the level set evolution is shown in Figure 7.25. Figure 7.26 shows the

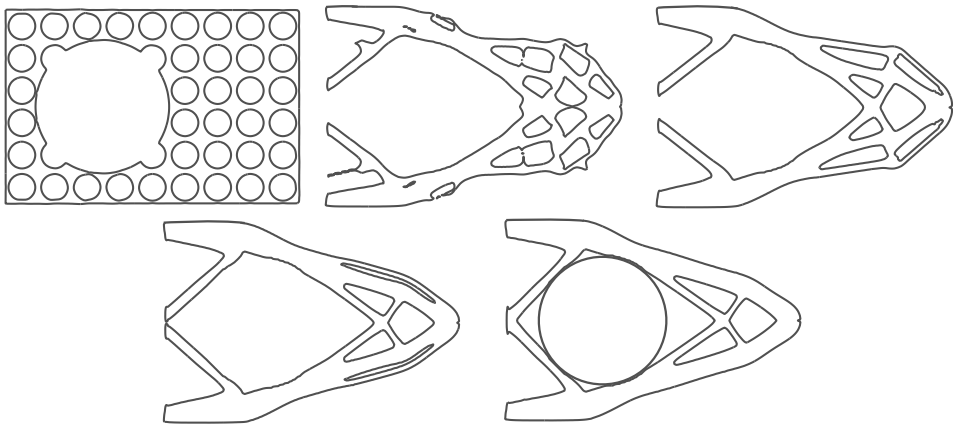


Figure 7.25: Level set history for cantilever with fixed hole subjected to loading case A

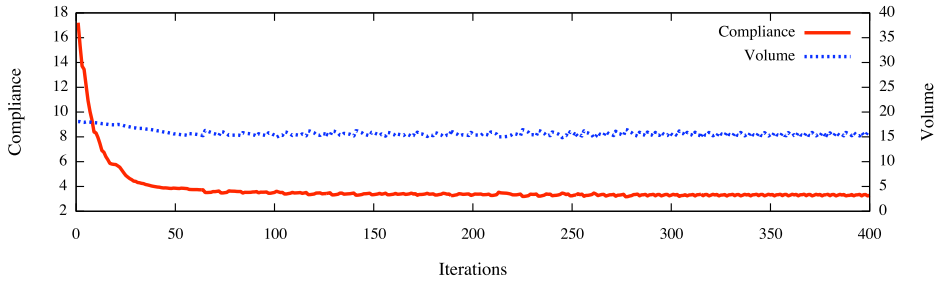


Figure 7.26: Compliance and volume time histories for loading case A

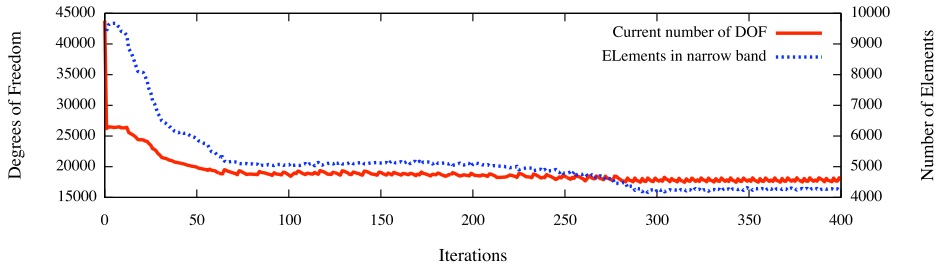


Figure 7.27: DOF and elements in narrow band time histories for loading case A

compliance and volume time histories for loading case A. No significant change can be seen in the volume history as the initial volume was close to the constraint volume condition. The time histories of number of DOF at each iteration and the number of elements within the narrow band are shown in Figure 7.27. The final topology shown in Figure 7.25 has four voids: one large and three small voids. The large void encompasses the fixed void condition, which is drawn alongside the final topology shown in Figure 7.25 for reference. The time required at each iteration of the optimization process is plotted in Figure 7.31. Without element removal, the total number of DOF would be constant at 43802 (shown at iteration zero in Figure 7.27) and the average time taken by the FEM computation would be approximately 24 seconds. However, when element removal was employed, the total number of DOF and CPU time required decreases over the course of the optimization process. In terms of total time saved, approximately 87% of CPU time can be saved by removing elements adaptively. Furthermore, when the volume constraint was reached, the average time time taken was a mere 1.91 seconds for a problem with 43802 DOF initially. The final topology obtained in this cases is almost identical to that of Bendsoe and Sigmund[17], and Belytschko et al.[25].

Loading case B

For loading case B, Figure 7.28 shows the level set evolution over time. Figure 7.29 shows the corresponding compliance and volume time histories. The number of DOF per iteration as well as the number of elements in the narrow band are shown in Figure 7.30. The time required at each iteration of the optimization process for this loading case is also plotted in Figure 7.31.

Without element removal, the total number of DOF at each iteration is 43802 (shown at iteration zero in Figure 7.30) and the average time taken for FEM calculation would be approximately 24 seconds per iteration. However, with element removal, the total number of DOF and the CPU time required decreases over the course of the optimization process. In terms of total time saved, approximately 89% of CPU time can be saved when element

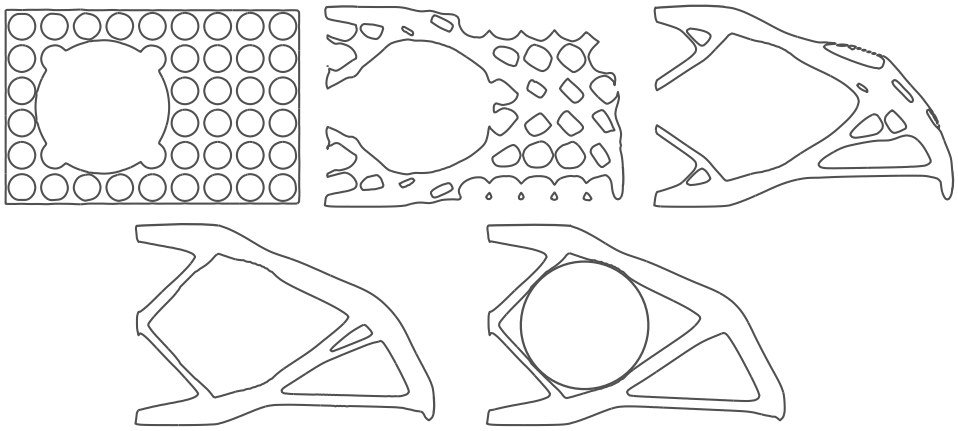


Figure 7.28: Level set history for cantilever with fixed hole subjected to loading case B

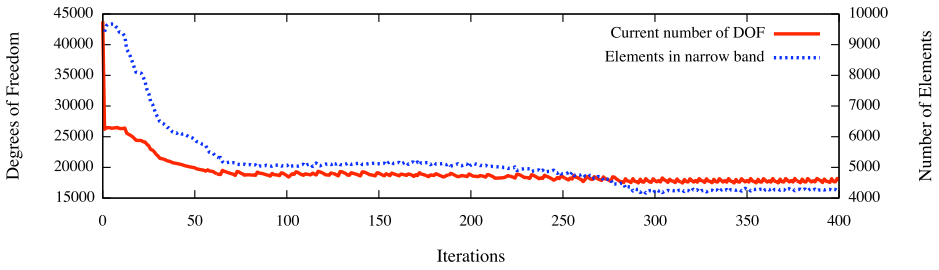


Figure 7.29: Compliance and volume time histories for loading case B

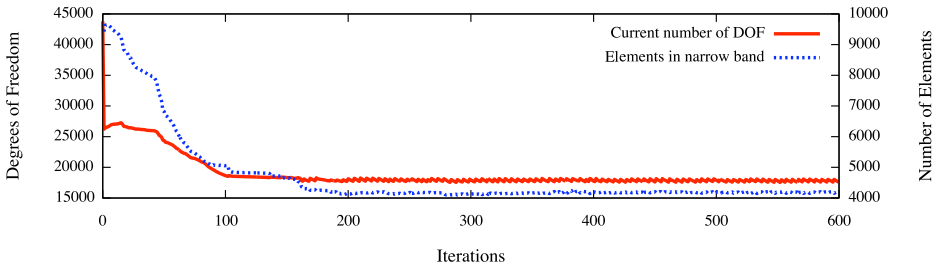


Figure 7.30: DOF and elements in narrow band time histories for loading case B

removal was employed. Furthermore, when the volume constraint was reached, the average time time taken was a mere 2.83 seconds for a problem with 43802 DOF initially. These results are summarized in Tables 7.1 and 7.2 respectively.

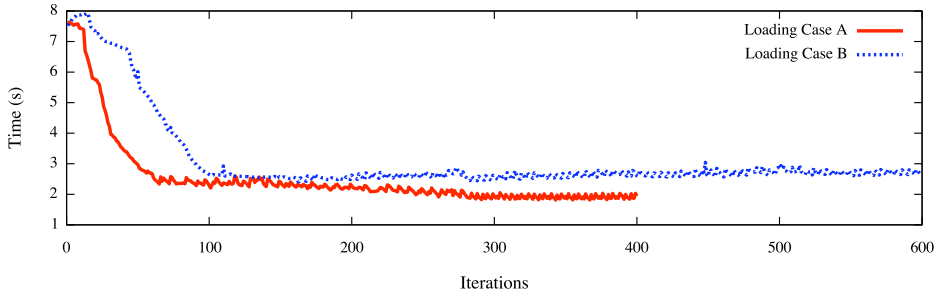


Figure 7.31: Time required for computing cantilevers with fixed holes

The final topology obtained in this case is a little different to that of Bendsoe and Sigmund[17], and Belytschko et al.[25]. The difference lies in the topology of the rightmost void. In their results, the rightmost void in Bendsoe and Sigmund[17], and Belytschko et al.[25] comprises of two voids, whereas the solution here presents one large void. The tighter volume constraint imposed in this example most likely influenced the layout of this one large void.

Table 7.3: Total time required for topology with fixed hole

	DOF	No element removal†(s)	Element removal†(s)	% time reduced
Fixed Hole Load A	43802	9685.86	1020.60	89.46
Fixed Hole Load B	43802	14527.34	1877.73	87.08

†CPU time on MacBook 2.4GHz Intel Core 2 Duo

Table 7.4: Time required per iteration after volume constraint is reached for topology with fixed hole

	DOF	No element removal†(s)	Element removal†(s)	% time reduced
Fixed Hole Load A	43802	24.21	1.91	92.11
Fixed Hole Load B	43802	23.56	2.83	87.99

†CPU time on MacBook 2.4GHz Intel Core 2 Duo

7.3 Creative design: Incorporating topological preferences

This section will present numerical examples to multi-objective topology optimization incorporating topological preferences. There will be two subsections. The first subsection will present examples when certain curvatures are desired in the optimized topology. The second subsection will present examples when certain voids are desired in the optimized topology.

The problem statement for the examples shown in this section is to minimize objective functions of strain energy density, $f_1(\phi)$, and signed least square error, $f_2(\phi)$, from a desired level set function value, $\phi_{desired}$, subject to equilibrium constraints and volumetric constraints:

$$\text{minimize} \quad f_1(\phi) = \frac{1}{2}\int_\Omega H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} d\Omega \tag{7.4}$$

$$f_2(\phi) = \frac{1}{2}\int_\Omega (\phi_{desired} - \phi)^2 d\Omega \tag{7.5}$$

$$\text{s.t.} \quad \nabla \cdot (H(\phi) E \boldsymbol{\sigma}) = \boldsymbol{p} \tag{7.6}$$

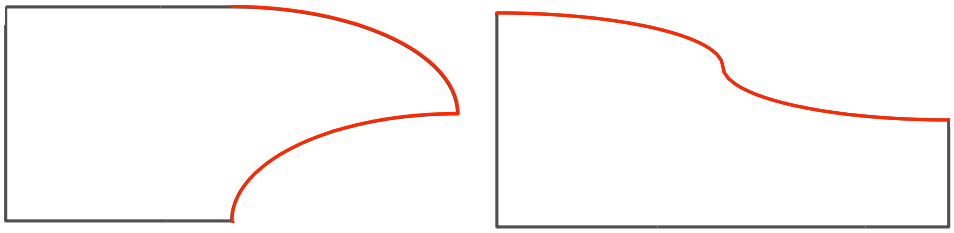
$$\int_\Omega H(\phi) d\Omega = V_0 \tag{7.7}$$

For the examples shown in this section, Young's Modulus, Poisson's ratio and loading were set to E =20GPa, $\nu = 0.3$ and $\boldsymbol{p}$ =50N respectively. The weight function for the signed

least square error is written as $w_{desired}$ and its value is varied in the following examples. Note that although the weight values are specified explicitly in the examples that follow, it is possible to automate the weight selection process.

7.3.1 Case 1: Unique curves

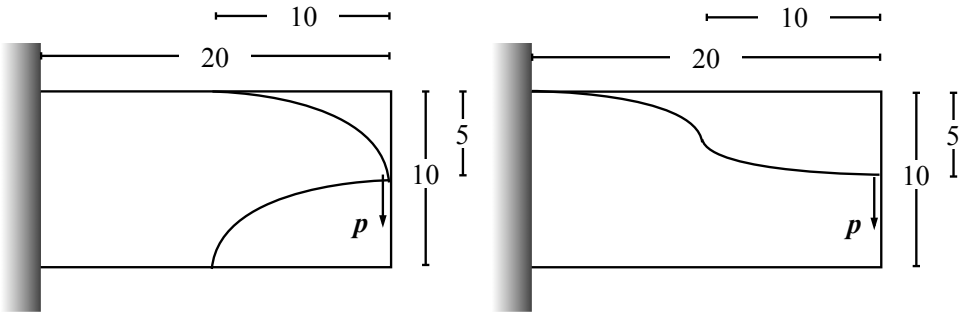
In this section, we explore two examples where specified unique curvatures are a desired topological property in the final topology. A single objective solution considering only the minimization of strain energy density would give a resulting topology similar to the one shown earlier in Figure 7.16. In addition to minimizing strain energy, the signed least square error described in Equation (4.45) for the respective desired curves shown in Figures 7.32(a) and 7.32(b) is also to be minimized. In this section, $w_{desired}$ is varied and unit weight is always used for $f_1(\phi)$. The weight function was set at unit because we do not wish for the boundary to evolve away from the respective loading and constraint conditions in favor of the desired boundary.



(a) Desired curves used for example 1 (b) Desired curves used for example 2

Figure 7.32: Desired curves shown in red

The configurations used for the two examples presented in this section are shown respectively in Figures 7.33 and 7.33(b). In the examples that follow, grid spacing, dx, was set to 0.25.



(a) Example 1 (b) Example 2

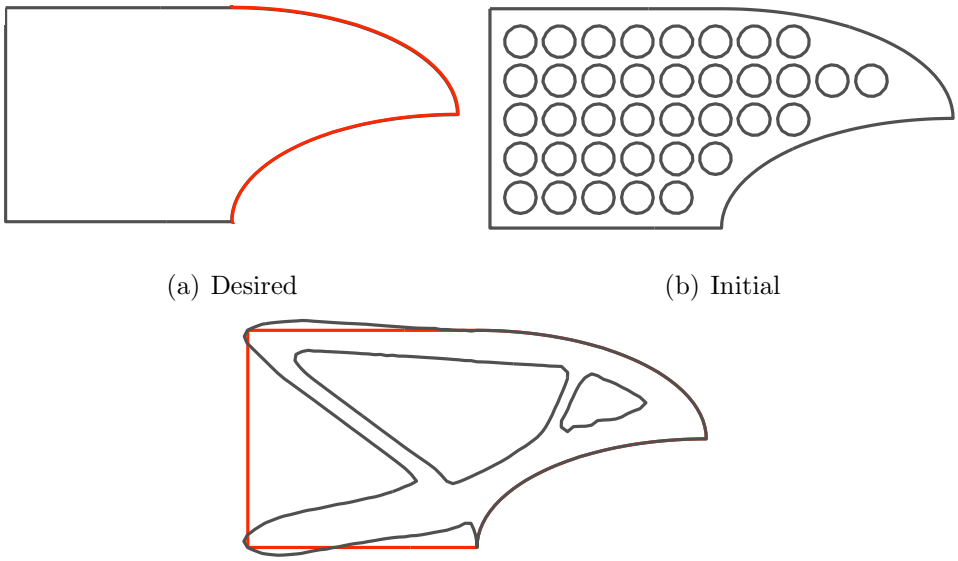
Figure 7.33: Example configurations

Example 1

Figures 7.34(a) and 7.34(b) show the desired curves and the initial level set function used in the optimization process. The desired curve used in this example consists of two elliptical sections in the upper and lower half of the right portion in the material domain. The dimensions of the desired curve in the material domain are shown in Figure 7.33(a). Figure 7.34(c) shows the topology if the boundaries where the desired curves are located are constrained and not allowed to move.

For this example, the volume constraint is set to 35% of the total volume in the design domain.

Figure 7.35 show the respective final topologies for different $w_{desired}$ weight values. The desired curves become more visible as the value of the weight is increased. In addition to how defined the desired curves are in the final topology, the weight values used also influence the size of the rightmost void in the final topology: as the weight values decrease, the hole increases in size. It appears from Figures 7.35(a) to 7.35(f) that the curvature in the bottom half of the material domain is not as easy to maintain as the curvature in the upper half. The curvature in the bottom half appears to be the main concern throughout the optimization process as it easily deviates from the desired curvature. As long as the curvature in the bottom half is somewhat distinguishable, the curvature in the upper half of the material



(a) Desired (b) Initial

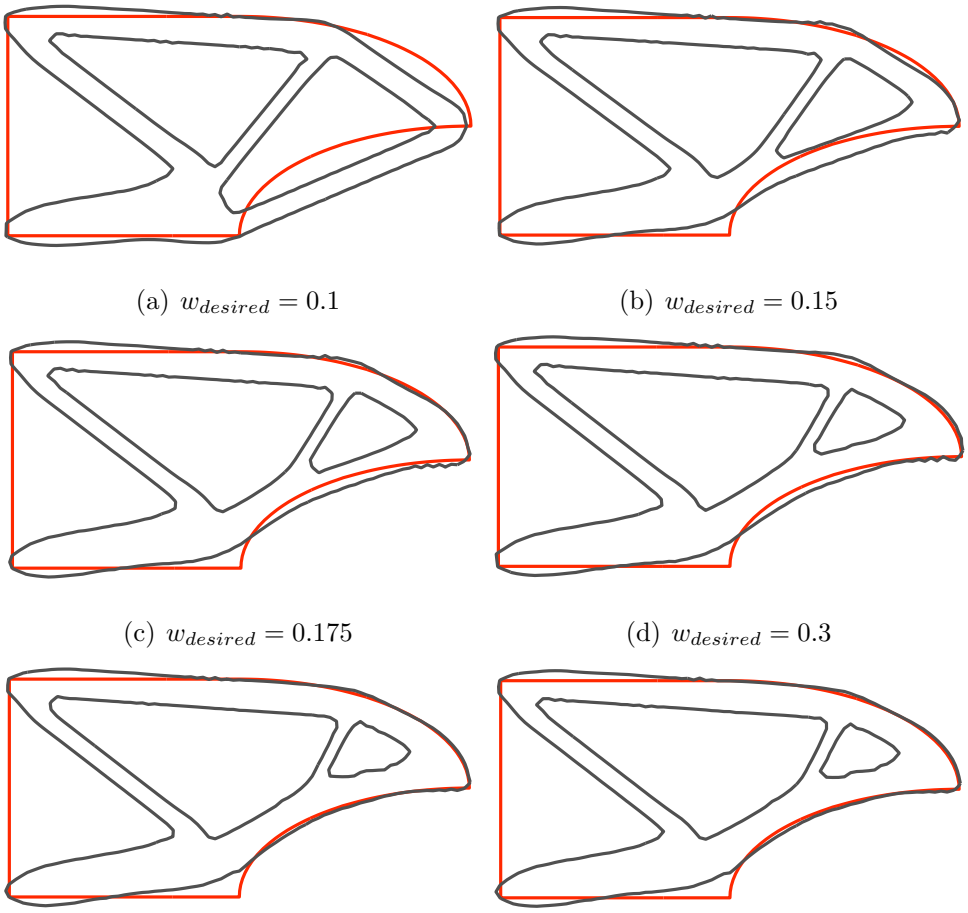
(c) Constrained solution

Figure 7.34: Example 1 configurations

domain is easily kept.

Figure 7.36 shows the Pareto front for this problem. The weight values used for $w_{desired}$ are (0.11, 0.125, 0.15, 0.175, 0.2, 0.225, 0.3, 0.6, 0.8, 1.0). The constrained solution shown in Figure 7.34(c) is also plotted with a signed least square error of zero for reference. The weight value of 0.1 used to obtain the final topology shown in Figure 7.35(a) was not plotted as it had an extremely high signed least square error. This indicates a disparity in signed least square errors between $w_{desired} = 0.1$ and $w_{desired} = 0.11$ and suggests that for weight values less than $w_{desired} = 0.11$, strain energy dominates the optimization process and the desired curvatures are no longer distinguishable.

Since the scheme of adaptively removing elements described in Section 5.3.2 is used here, the time required for each of the cases shown in Figure 7.35 are shown in Figure 7.37. With an initial configuration shown in Figure 7.34(b), the amount of time required was about 0.5 seconds an iteration. Once the volume constraint has been reached, the time required drops to an average of 0.225 seconds per iteration.



(a) $w_{desired} = 0.1$ (b) $w_{desired} = 0.15$

(c) $w_{desired} = 0.175$ (d) $w_{desired} = 0.3$

(e) $w_{desired} = 0.8$ (f) $w_{desired} = 1.0$

Figure 7.35: Example 1 final configurations

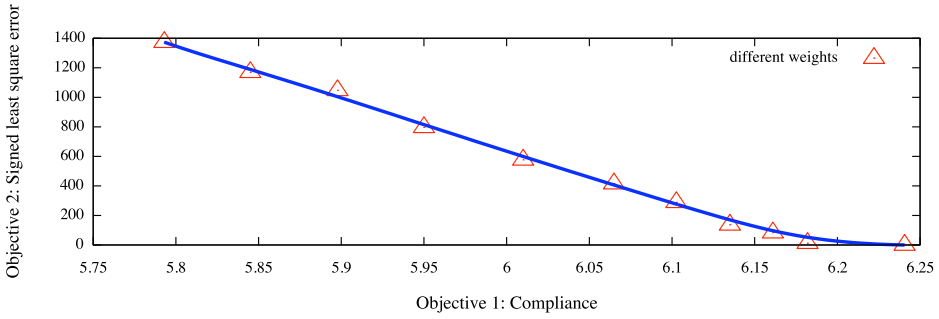


Figure 7.36: Example 1 Pareto Front

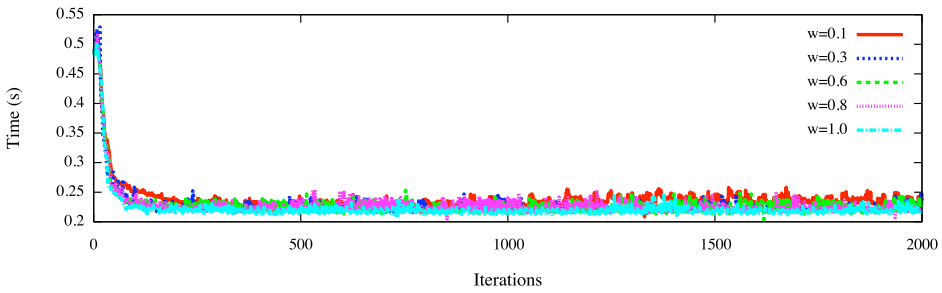


Figure 7.37: History of time required for the optimization process

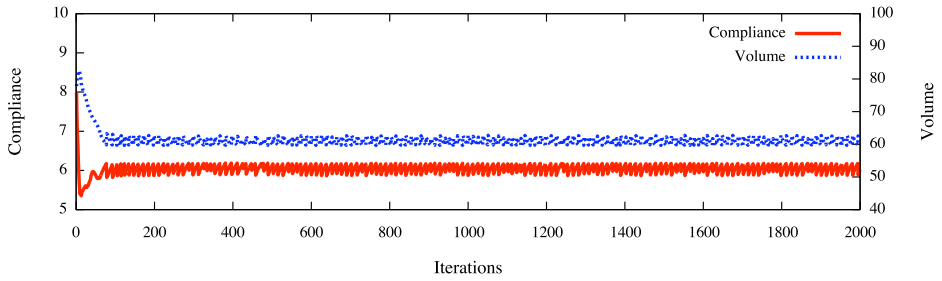


Figure 7.38: Compliance and volume vs. iterations for $w_{desired} = 0.8$

For the case when $w_{desired} = 0.8$ is used, time histories of compliance, volume, current number of DOF and elements in narrow band are plotted in Figures 7.38 and 7.39 respectively. (For explanations on current number of DOF, see Section 5.3.2).

Total degrees of freedom without element removal is 6642 shown at $t = 0$ in Figure 7.39. By removing void elements during the optimization process, the current number of degrees of freedom decreases until it plateaus at approximately 3900 when the volume constraint is reached. The same can be observed for the elements in the narrow band, which plateaus at approximately 1350.

Example 2

In this example, we explore the case when the variation from the desired curvature shown in red in Figure 7.40(a) is added as an objective to be minimized. As with the previous

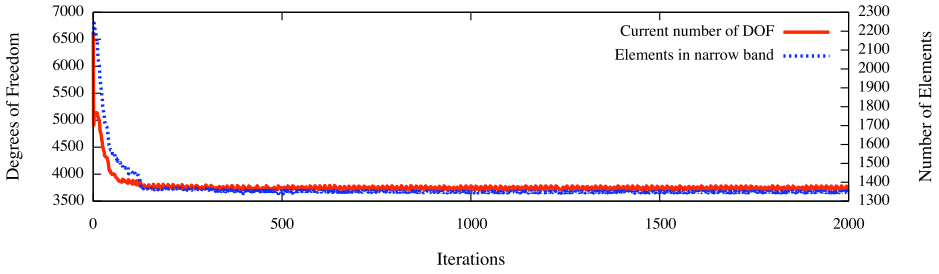
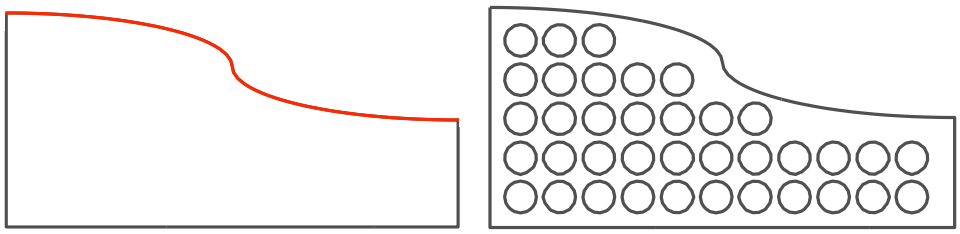
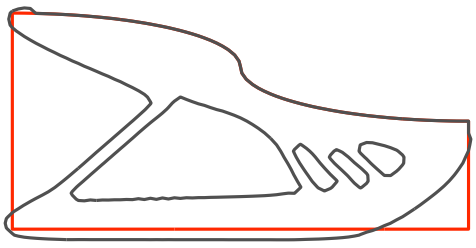


Figure 7.39: DOF and elements in narrow band for $w_{desired} = 0.8$

example, the desired curvature used in this example also consists of two elliptical sections. However, the positioning of these sections differ from the previous example. The elliptical sections are positioned in the upper half of the material domain on the left and right portions. The initial level set configuration used in this example is shown in Figure 7.40(b). Figure 7.40(c) shows a topology when the desired curve is treated as a constraint and not allowed to deviate. For this example, the volume constraint is set to 45% of the total volume in the design domain.



(a) Desired (b) Initial



(c) Constrained solution

Figure 7.40: Example 2 configurations

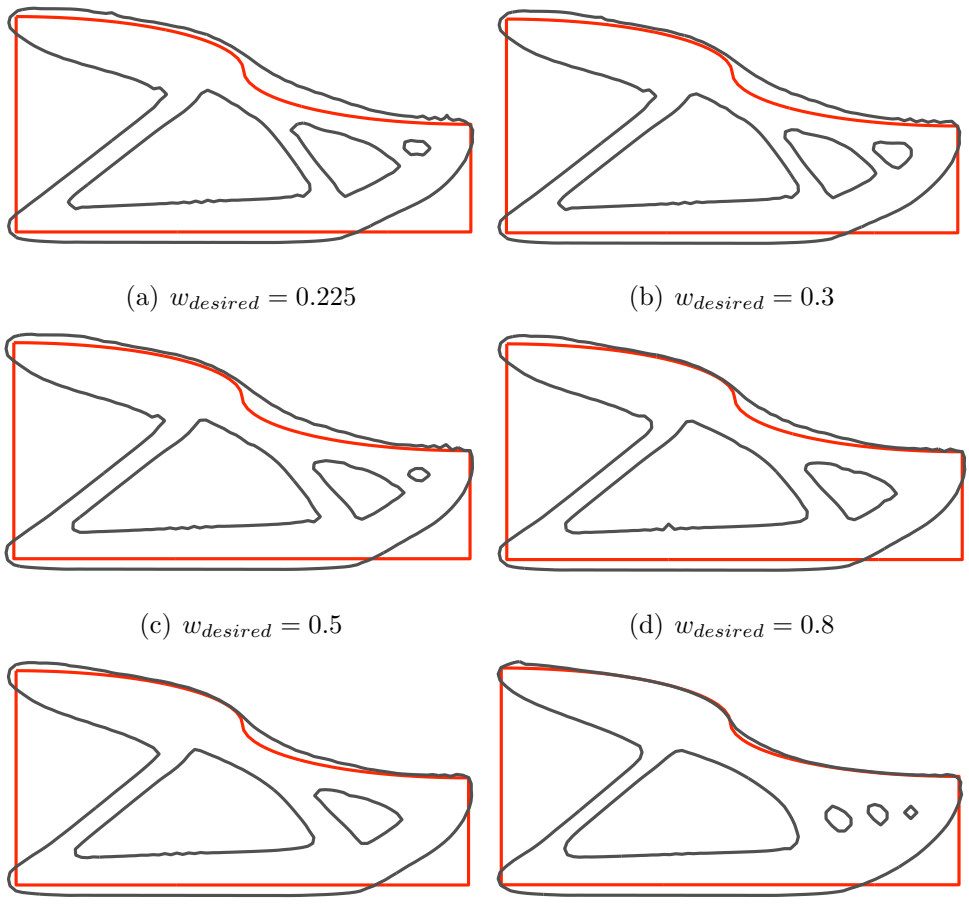
Figure 7.41 shows the respective final topologies for different $w_{desired}$ weight values. As with the previous example, the desired curves become more distinguishable as the value of $w_{desired}$ is increased. In addition to how defined the desired curves are in the final topology, the weight values used also influence the number of and size of voids in the rightmost portion of the final topology. Topologies in Figures 7.41(a) to 7.41(c) show two voids with varying sizes in the rightmost portion of the design domain. Topologies in Figure 7.41(d) and 7.41(e) both show only one void. Moreover, the topology in Figure 7.41(f) shows a similarity with Figure 7.40(c) and has three holes. The section where of the desired curvature where the concavity changes appears to be the main concern of the optimization process.

Figure 7.42 shows the compliance time histories of all the topologies shown in Figure 7.41. The compliance time history for the constrained solution depicted in Figure 7.40(c) is also plotted as $w_{desired} = 0$ for reference. The trend visible in Figure 7.42 shows the compliance value decreasing as $w_{desired}$ is relaxed. In other words, less emphasis is placed on maintaining the desired curves shown in Figure 7.40(a). The opposite can be seen in Figure 7.43 where the signed least square error, plotted with a log scale, decreases as the value of $w_{desired}$ increases.

Figure 7.44 shows the Pareto front for this problem. The weight values used for $w_{desired}$ are (0.225, 0.25, 0.3, 0.5, 0.8, 1.0, 2.0). The constrained solution shown in Figure 7.40(c) is also plotted with a signed least square error of zero for reference. For weight values less than $w_{desired} = 0.225$, strain energy dominates the optimization process and the desired curves are no longer distinguishable.

Since the scheme of adaptively removing elements described in Section 5.3.2 is used here, the time required for each of the cases shown in Figure 7.41 are shown in Figure 7.45. With an initial configuration shown in Figure 7.40(b), the amount of time required was approximately 0.65 seconds an iteration. Once the volume constraint has been reached, the time required drops to an average of 0.4 seconds per iteration.

For the case when $w_{desired} = 1.0$, time histories of compliance, volume, current number



(a) $w_{desired} = 0.225$ (b) $w_{desired} = 0.3$

(c) $w_{desired} = 0.5$ (d) $w_{desired} = 0.8$

(e) $w_{desired} = 1.0$ (f) $w_{desired} = 2.0$

Figure 7.41: Example 2 final configurations

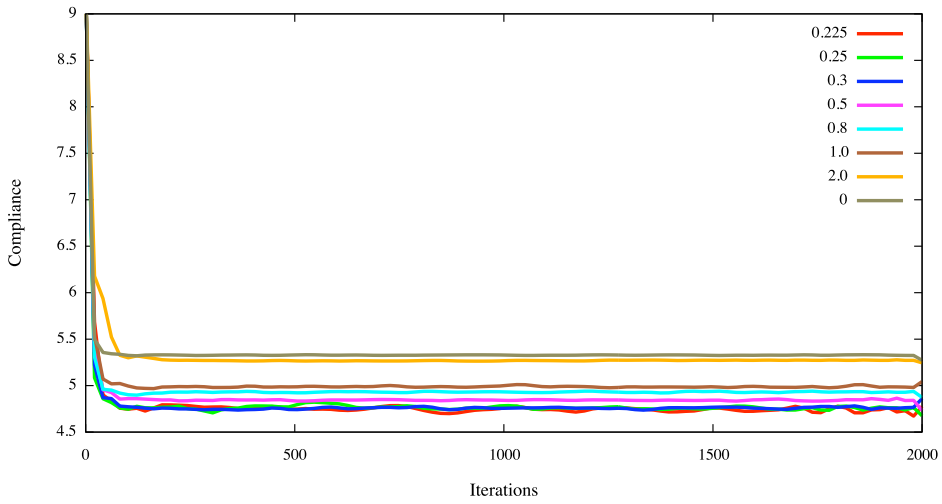


Figure 7.42: Compliance vs. iterations

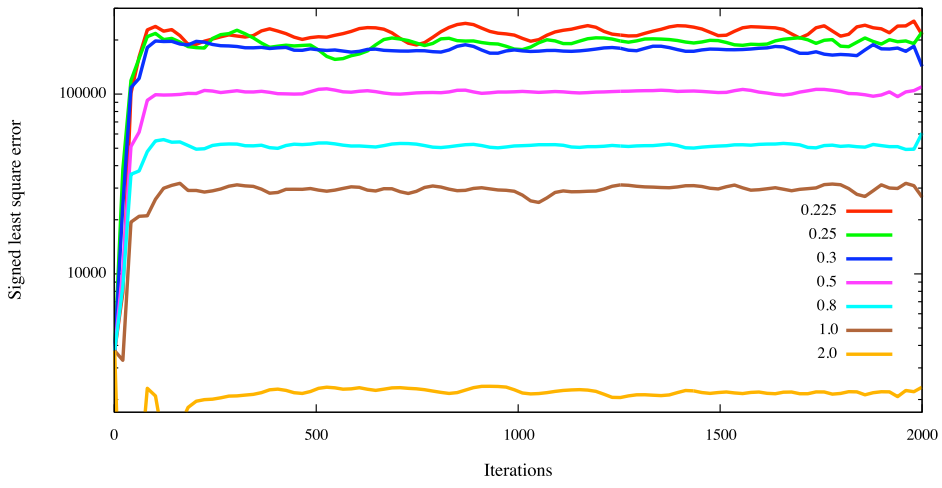


Figure 7.43: Signed least square error vs. iterations

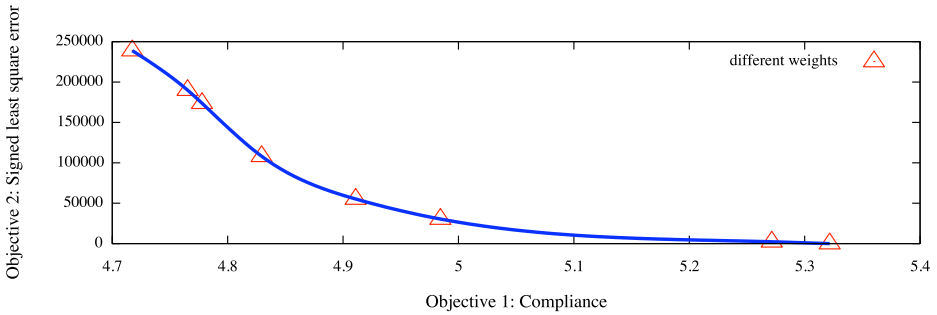


Figure 7.44: Example 2 Pareto Front

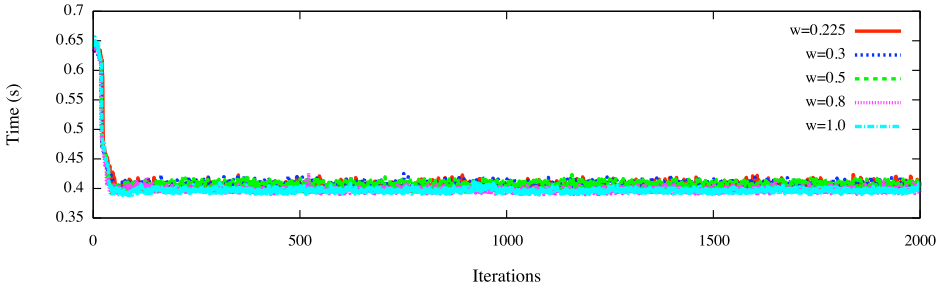


Figure 7.45: History of time required for the optimization process

of DOF and elements in narrow band are plotted in Figures 7.46 and 7.47 respectively. (For explanations on current number of DOF, see Section 5.3.2). Total degrees of freedom without element removal is 6642 shown at $t = 0$ in Figure 7.39. By removing void elements during the optimization process, the current number of degrees of freedom decreases until it plateaus at approximately 4800 when the volume constraint is reached. The same can be observed for the elements in the narrow band, which plateaus at approximately 2070. The final number of DOF and number of elements in the narrow band of this example is slightly larger when compared to the previous example as the volume constraint is set higher.

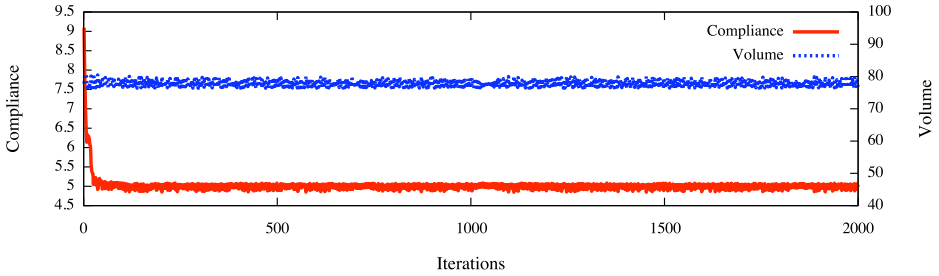


Figure 7.46: Compliance and volume vs. iterations for $w_{desired} = 1.0$

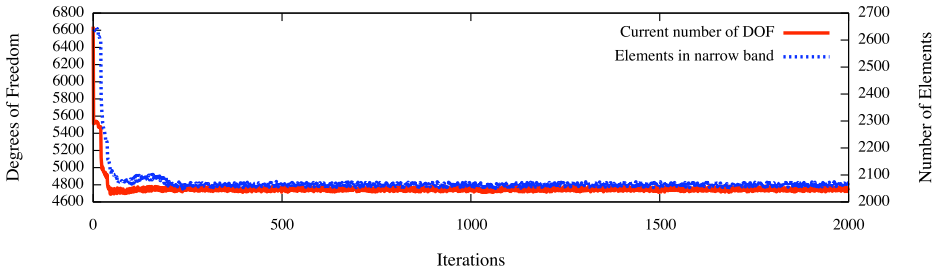
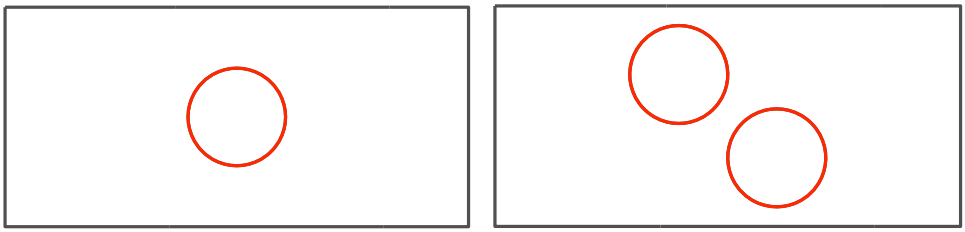


Figure 7.47: DOF and elements in narrow band for $w_{desired} = 1.0$

7.3.2 Case 2: Voids with movable boundaries

In this section, we explore two examples where specified voids are a desired topological property in the final topology. This presents an interesting problem where designers and

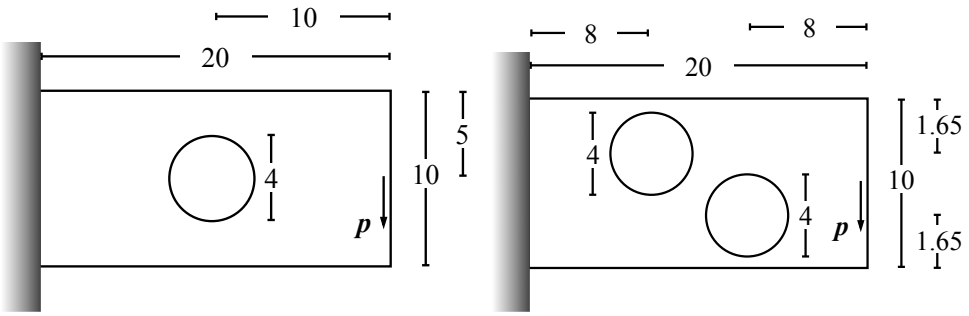
engineers can express their preference for voids in certain areas of the structure. The specified voids are shown in red in Figures 7.48(a) and 7.48(b). The first example consists of a single void whereas the second example consists of two voids. As with the previous section, the weight function for the signed least square error is written as $w_{desired}$ and unit weight is always used for $f_1(\phi)$. The full problem statement is written above in Section 7.3.



(a) Desired voids used for example 1 (b) Desired voids used for example 2

Figure 7.48: Desired voids shown in red

The configurations used for the two examples presented in this section are shown respectively in Figures 7.49(a) and 7.49(b). In the examples that follow, grid spacing, dx, was set to 0.25 and volume constraints were set to 45% of the material domain.



(a) Example 1 (b) Example 2

Figure 7.49: Example configurations

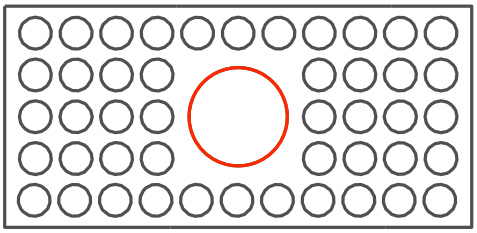


Figure 7.50: Initial level set with desired void shown in red

Example 1

The initial level set shown in Figure 7.50 is used for this example. It includes the desired void which is drawn in red. This example uses a unit weight value, $w_{desired} = 1$. The final outcome of the optimization process is shown in Figure 7.51. Except for the position of the three voids in Figure 7.51, the topology is quite similar to the one seen earlier in Figure 7.16. In this case, although the void has expanded to the right, it has prevented the material boundary from crossing on its left.

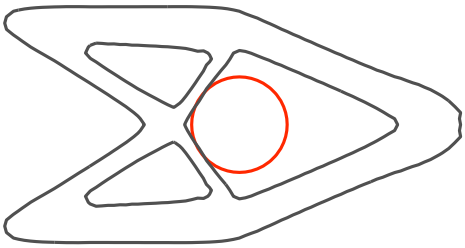


Figure 7.51: Final topology with desired void shown in red for $w_{desired} = 1.0$

The corresponding compliance and volume time histories are shown in Figure 7.52. Between iterations 100-200, some irregularities were observed, but these were eventually ironed out after the 200th iteration. The same can be seen in Figure 7.53 where the number of DOF and elements in narrow band time history is plotted. This is probably due to some merging or separating of voids during the optimization process.

Another example was tried with a weight value of $w_{desired} = 0.1$. In this instance, the

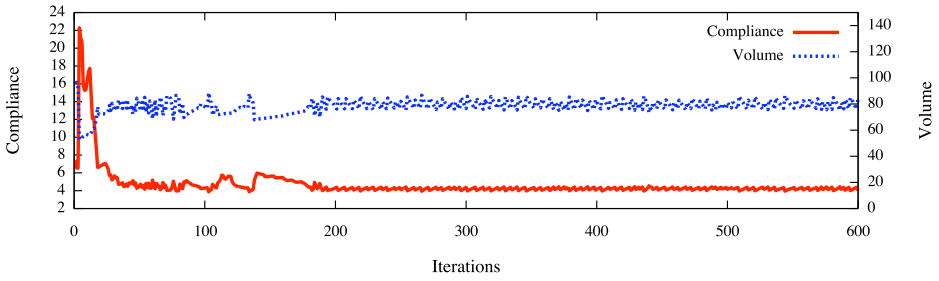


Figure 7.52: Compliance and volume time histories for $w_{desired} = 1.0$

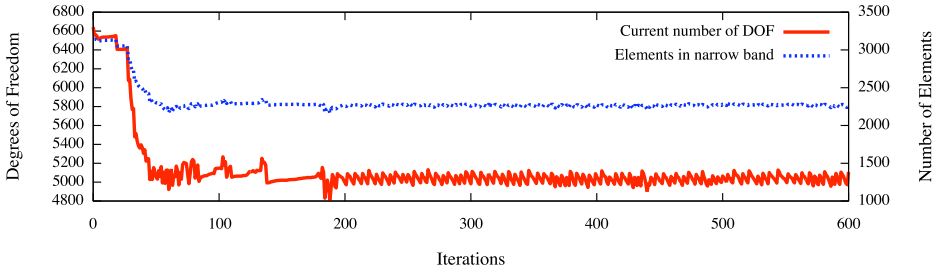


Figure 7.53: DOF and elements in narrow band for $w_{desired} = 1.0$

resulting topology turned out quite different as can be seen in Figure 7.54: it resembles a mitchell-type truss structure. Although the void in the center of the design domain is still visible, its desired boundaries are slightly compromised. The corresponding compliance and volume time histories for this instance are shown in Figure 7.54. The number of DOF and elements in narrow band time history are also plotted in Figure 7.56. By reducing $w_{desired}$ we could get a slight improvement in compliance by compromising on the shape of the void.

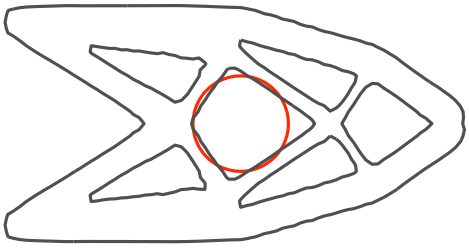


Figure 7.54: Final topology with desired void shown in red for $w_{desired} = 0.1$

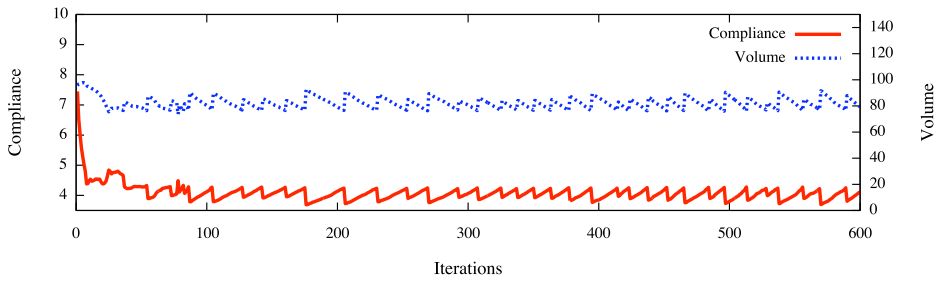


Figure 7.55: Compliance and volume time histories for $w_{desired} = 0.1$

Example 2

Instead of only specifying one desired void, this example specifies two for optimization. The initial level set shown in Figure 7.57 is used. It includes the desired void which is drawn in red. This example uses a unit weight value, $w_{desired} = 1$. The final outcome of the optimization process is shown in Figure 7.58. Without the specified voids, the final topology

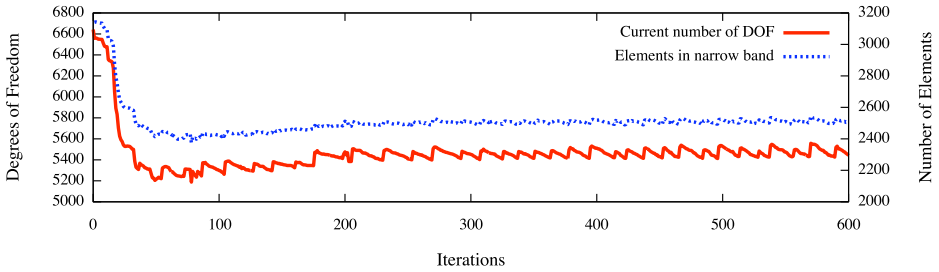


Figure 7.56: DOF and elements in narrow band for $w_{desired} = 0.1$

would have been symmetric. However, given the location of the specified voids, the final topology was not allowed to be symmetric. The final topology in Figure 7.57 shows that the specified voids have expanded in certain areas and are preventing the material boundary from compromising its space in other areas.

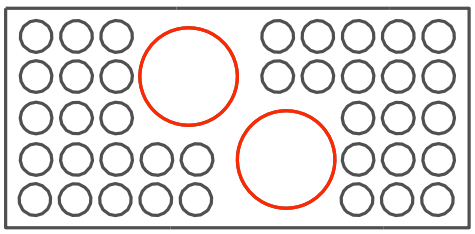


Figure 7.57: Initial level set with desired void shown in red

The compliance and time histories for this problem are shown in Figure 7.59. The corresponding number of DOF and elements in the narrow band are shown in Figure 7.60. A sudden drop is observed just before the 200th iteration in Figure 7.60. This is probably due to the merging of smaller voids into a large void as the number of elements in the narrow band decreased accordingly.

Another example was tried with a weight value of $w_{desired} = 0.2$. In this instance, the resulting topology turned out slightly different as can be seen when comparing Figures 7.58 and 7.61. In Figure 7.61, the rightmost large void in Figure 7.58 is separated into two smaller voids. Furthermore, much of the specified voids have been compromised. Although

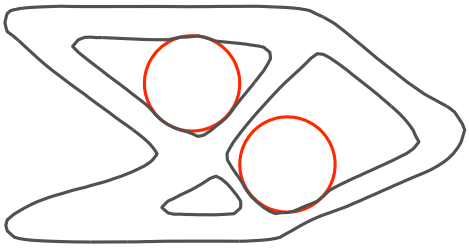


Figure 7.58: Final topology with desired void shown in red for $w_{desired} = 1.0$

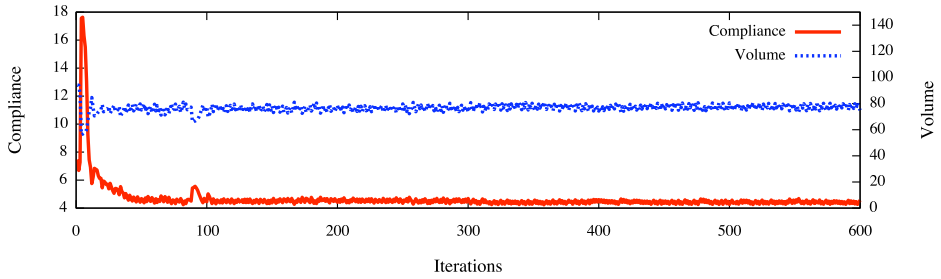


Figure 7.59: Compliance and volume time histories for $w_{desired} = 1.0$

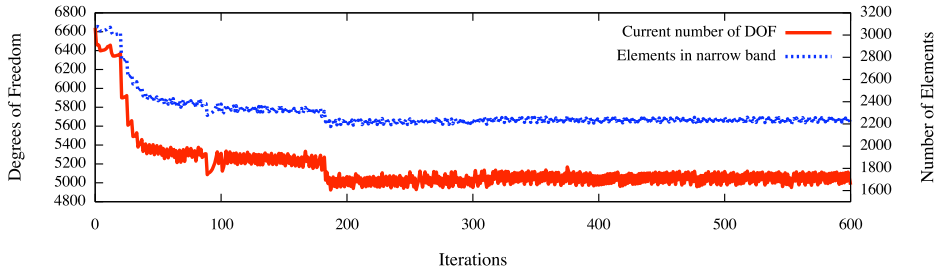


Figure 7.60: DOF and elements in narrow band for $w_{desired} = 1.0$

the specified voids are not strictly kept in this example, they provide an interesting case where voids can be loosely specified at certain locations in the design domain. The corresponding compliance and volume time histories for this instance are shown in Figure 7.62. The number of DOF and elements in narrow band time history are also plotted in Figure 7.63. By reducing $w_{desired}$ we could get a slight improvement in compliance by compromising on the shape of the void.

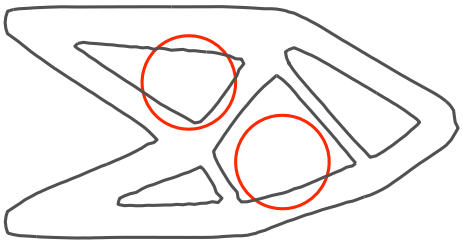


Figure 7.61: Final topology with desired void shown in red for $w_{desired} = 0.2$

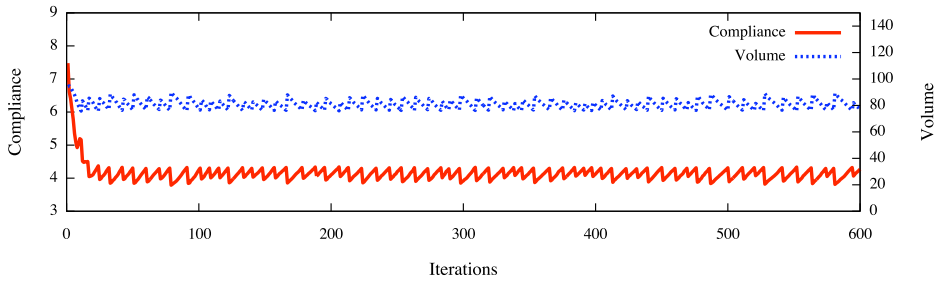


Figure 7.62: Compliance and volume time histories for $w_{desired} = 0.2$

Figure 7.64 shows both topologies resulting from the different weight values discussed above without the superimposed desired voids. In Figure 7.64(a), irregular curves can be seen in the outline of the two large voids where the desired voids are supposed to be. These irregular curves outline the boundaries of the desired voids that are not compromised. In contrast to Figure 7.64(a), the topology in Figure 7.64(b) does not show any irregular curves. Although the two topologies shown in Figure 7.64 have similar compliance values, the topol-

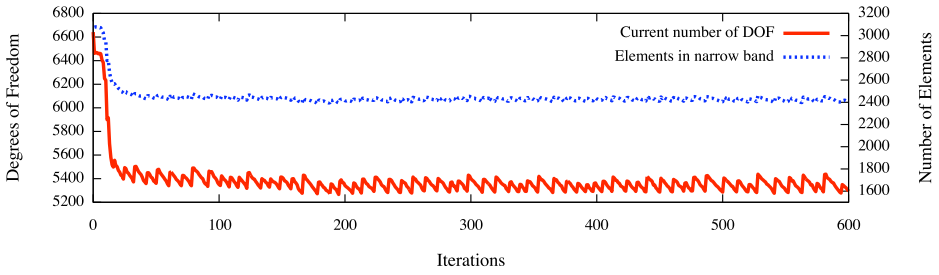
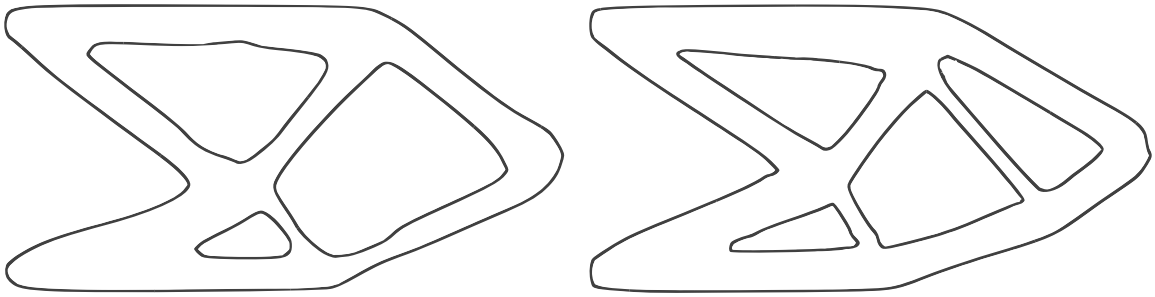


Figure 7.63: DOF and elements in narrow band for $w_{desired} = 0.2$

ogy in Figure 7.64(b) has a slightly lower value and by visual inspection, is less prone to fatigue failure.



(a) $w_{desired} = 1.0$ (b) $w_{desired} = 0.2$

Figure 7.64: Comparing final topologies for example 2

Chapter 8

Conclusions

8.1 Achievements and findings

This dissertation contributed a novel concept for allowing the structural optimization process to be influenced by specific designer inputs regarding topological properties and preferences, such as boundary curvature and location of voids. These specified inputs are not strictly enforced as the proposed concept is posed as a multidisciplinary problem that finds a balance between engineering and design objectives. Within the scope of this dissertation, engineering objectives were described with metrics such as compliance and design objectives were described with topological properties that add aesthetic and/or functional value to the structure. For example, design objectives can include aspects of architectural design.

A deterministic multi-objective method using a real valued function was developed to solve this unique multidisciplinary design problem that involves topological preferences. The real valued function is a key component of the multi-objective method that combines multiple objectives. This method was used to solve several sizing and shape optimization problems. The clear Pareto front shown in the respective solutions validate the multi-objective method. In sizing optimization, additional objective functions were introduced to reflect preference towards certain size values. Due to its advantages, level set methods were used to perform

topology optimization. Within the level set framework, it has been shown that designer inputs specifying topological properties can be created very naturally and with relative ease Desired topologies can be aesthetic effects drawn by designers or functional intents required by engineers. In the topology optimization process, an objective function that measures deviation from those desired topological properties was proposed. This objective function suggests preference for specified topological properties such as preferences for boundary curvature and/or placement of voids in the structure. How strictly these desired topological properties are adhered to depends on the weighting factors used when combining the multiple objective functions. This was clearly established in the numerical examples.

When considering the speed of the optimization processes involving structural analyses, FEM computation is the bottleneck. Speeding up processes has always been a challenging task. Certain techniques have been proposed in this dissertation to tackle this issue. For structural sensitivity analyses when the number of design variables greatly outnumber the number of objective functions, a combination of the real valued function and the adjoint variable method can be used to reduce the time required for the involved FEM computations. This was also applicable to shape optimization. However, in case of shape optimization, a discrete force method had to be used as a boundary smoother in order to ensure that the boundaries remain smooth and no finite elements collapse during the optimization process.

In order to significantly speed up the level set method used in topology optimization, a high pass strain energy filter and an adaptive scheme for removing elements determined to be of void material by the level set have been proposed to significantly reduce computational cost. The high pass strain energy filter reduces computational cost by aggressively removing material to meet the volume constraint. As a result, it increases time efficiency by reducing the number of iterations required to reach convergence. On the other hand, the introduced adaptive scheme takes a different approach and aimed at reducing the number of finite element equations in the system. By reducing the number of system equations, the amount of time required to solve the FEM equations can be reduced exponentially. An analysis

which required many hours can now be run in just under an hour. A major advantage of this adaptive scheme is also its generality; it can be applied to any level set based topology optimization method with relative ease and reduce the computational time required by as much as 90%.

The numerical examples shown in this dissertation substantiated the innovative use of the methods and techniques described above to solve multidisciplinary structural design involving topological preferences. By employing the proposed mathematical models, the problem becomes interactive as it becomes possible to include preferred topological properties. The specification for these preferences can be derived from various disciplines. For example, boundary curvature can be linked to aesthetics and manufacturing constraints might require specific locations for voids. Although the numerical examples only show one objective function representing desired topology, it is possible to have various instances of this function to represent different topological properties. By having different instances, it then becomes viable that different weights can be placed on different aspects of desired topological properties. When aesthetics are considered an objective in structural topology optimization, the interpretation of the multidisciplinary optimization problem becomes one that searches for beauty and performance in topology. It is hoped that this novel concept will spark further research interest into design processes that offer a two-way street between engineering and design teams.

8.2 Future work

The research presented in this dissertation is novel and aims to solve a very practical design problem; balancing design and engineering objectives when it comes to structural topology optimization. Although there are many possible directions for this research to pursue in the future, only the most promising shall be listed in this section.

A direction to pursue from here is to include many desired topological preferences and

treat each preference as an objective function. This increases the complexity of the problem by introducing more objective functions which in turn increases the number of weight values. However, the included benefit is more customizability for the user. The increased customizability is achieved by those additional weight values which the user can vary in order to set different preference/emphasis on local topological properties.

Since real world structures are three-dimensional, another obvious direction future work can take is to implement the proposed methods and techniques to solve three-dimensional multidisciplinary topology optimization involving topological preferences. Extending the level set method to describe three-dimensional bodies is relatively straightforward and has been done in a variety of research papers. Extending desired topological properties such as two-dimensional curves to three-dimensional surfaces is doable, but might require a little more effort. For three-dimensional structural problems, the proposed scheme to adaptively remove elements will be extremely useful in reducing the amount of computational time required in solving FEM system equations.

Something else that can be done in the future is to create a system that runs many instances of the topology optimization process in parallel. By doing this, users will not be required to specify explicit weight values. Instead, the system can be programmed to randomize the weight values in favor of developing the Pareto front. In other words, the user will input the conditions and constraints of the topology optimization problem as well as the desired topological properties. The system then runs completely on its own, developing topological solutions along the Pareto front. Once finished, the system will present the topological solutions on the Pareto front directly to the user. The user can then select a topological solution without having to worry about the selection of weight values. As a result, the whole multidisciplinary structural topology optimization process is automated.

Appendix A

Normalizing objective functions

This appendix provides supplementary information regarding the numerical implementation of an algorithm to select maximum and minimum values used in Chapter 4. The multi-objective topology optimization problem statement is given by

$$\text{minimize} \quad f_1(\phi) = \frac{1}{2}\int_\Omega H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} d\Omega \tag{A-1}$$

$$f_2(\phi) = \frac{1}{2}\int_{\Omega_{unb}} (\phi_{desired} - \phi)^2 d\Omega_{unb} \tag{A-2}$$

$$\text{s.t.} \quad \nabla \cdot (H(\phi) E \boldsymbol{\sigma}) = \boldsymbol{p} \tag{A-3}$$

$$\int_\Omega H(\phi) d\Omega = V_0 \tag{A-4}$$

where Young's Modulus, Poisson's ratio and loading are represented by E, ν and $\boldsymbol{p}$ respectively. The domain Ω_{unb} refers to the user-specified narrowband where $\phi_{desired}$ values are defined. For more details on Ω_{unb}, please refer to Chapter 4.4.2. Detailed explanations regarding the formulation of the objective functions can be found in Chapter 4.4.

The solution of this problem required the equations to be rewritten discretely and solved for node by node; the level set method essentially evolves the boundary by updating the level set function values defined at the nodes. Therefore, each node has a discrete objective

function given by:

$$f_1'(\phi) = \frac{1}{2} H(\phi) E \boldsymbol{\varepsilon}^T \boldsymbol{D} \boldsymbol{\varepsilon} \tag{A-5}$$

$$f_2'(\phi) = \frac{1}{2} (\phi_{desired} - \phi)^2. \tag{A-6}$$

Presented with these two discrete objective functions, the real valued function described in Chapter 4.2.1 is used to aggregate them into a scalar function for optimization. The aggregating procedure is two-fold. Individual, i^{th}, discrete objective functions are first normalized by using the individual real valued function,

$$\phi_i' = \frac{f_i' - f_{i,min}'}{f_{i,max}' - f_{i,min}'}, \tag{A-7}$$

where $f_{i,min}'$ and $f_{i,max}'$ represent the minimum and maximum values in a population sample defined by the nodes in the Cartesian grid within their respective domains. For objective function $f_1'(\phi)$, the selection of maximum and minimum values is straightforward as they are selected globally from within Ω. However, in the case of $f_2'(\phi)$, the maximum and minimum values are only selected from within Ω_{unb}. This is done to limit boundary interaction with the signed least square error objective function, $f_2'(\phi)$, to within the specified narrowband; the choice of narrowband width is entirely left to user selection. Since topological preferences are defined locally with $\phi_{desired}$, this limitation, also applied locally, is justifiable. If there are additional signed least square error objective functions, the same procedure can easily be employed for selecting the maximum and minimum values. The procedure to implement the real valued function into level set based topology optimization is quite different from regular sizing and shape optimization. Please see Chapter 4.4.2 for more details.

The normalized objective functions are evaluated at each iteration. In other words, as the boundary moves, objective function values change. Therefore, new maximum and minimum values have to be selected again. Since the objective function values are located at the nodes, they can easily be stored in an array. Selecting maximum and minimum values at each iteration will simply be a call to a MAX or MIN function that selects the respective values from the array; this is automatic and no user effort is required in the selection process.

There was some concern that the constant change in maximum and minimum objective function values would require a change in the weight values too. However, this concern is uncalled for as the maximum and minimum values are used to normalize the discrete objective functions and do not have an explicit relationship with the user-defined weight value. The constant change in maximum and minimum objective function values is required to properly represent the current population sample.

Once the discrete objective functions have been normalized, they are aggregated with weight values into the total real valued function described in Equation (4.8) for optimization.

Bibliography

[1] de Weck O., "Multiobjective optimization: history and promise," in *Proceedings of Third China-Japan-Korea Joint Symposium of Structural and Mechanical Systems*, (Kanazawa, Japan), 2004.

[2] E. Triantaphyllou, B. Kovalerchuk, L. Mann, and G. Knapp, "Determining the most important criteria in maintenance decision making," *Journal of Quality in Maintenance Engineering*, vol. 3, no. 1, pp. 16–28, 1997.

[3] E. Triantaphyllou and B. Shu, "On the maximum number of feasible ranking sequences in multicriteria decision making problems," *European Journal of Operational Research*, vol. 130, pp. 665–678, 2001.

[4] J. Knowles, M. Oates, and D. Corne, "Multi-objective evolutionary algorithms applied to two problems in telecommunications," *BT Technology Journal*, vol. 18, no. 4, pp. 51–64, 2000.

[5] C. J. K. Lee, T. Furukawa, and H. Noguchi, "Multi-objective and constrained gradient-based approach to parametric design," in *Proceedings of Computational Engineering Conference JSCES*, vol. 10, (Japan), May 2005. Computational Engineering Conference Proceedings, JSCES.

[6] C. J. K. Lee, H. Noguchi, and T. Furukawa, "A multi-objective truss design system using a gradient-based optimization method," in *Conference Proceedings*, (Chennai,

India), December 2005. International Conference on Computational & Experimental Engineering and Sciences, ICCES.

[7] T. Furukawa, C. J. K. Lee, and J. G. Michopoulos, "Regularization for parameter identification using multi-objective optimization," in *Multi-Objective Machine Learning* (Y. Jin, ed.), vol. 16 of *Studies in Computational Intelligence*, ch. 6, pp. 125–149, Springer Berlin / Heidelberg, 2006.

[8] C. J. K. Lee, T. Furukawa, and S. Yoshimura, "Multi-objective human-like optimization for computational engineering design," in *Congress Proceedings*, (Beijing, China), Tsinghua University Press & Springer-Verlag, September 2004. Proceedings of the Sixth World Congress on Computational Mechanics, WCCM VI.

[9] I. Kim and O. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural Multidisciplinary Optimization*, vol. 29, pp. 149–158, 2005.

[10] I. Kim and O. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 31, pp. 105–116, 2006.

[11] D. van Veldhuizen and G. Lamont, "Multiobjective evolutionary algorithms: analyzing the state of the art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

[12] F. C.M. and P. Fleming, "Genetic algorithms for multi-objective optimization: formulation, discussion and generalization," in *Proceedings of the fifth International Conference on Genetic Algorithms*, (San Mateo, CA), 1993.

[13] M. Li, J. Kou, and L. Dai, "Ga-based multi-objective optimization," in *Proceedings of the third World Congress on Intelligent Control and Automation*, (Hefei, PR China), 2000.

[14] S. Obayashi, D. Sasaki, Y. Takeguchi, and N. Hirose, "Multiobjective evolutionary computation for supersonic wing-shape optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 4, July 2000.

[15] S. Obayashi, D. Sasaki, and A. Oyama, "Finding tradeoffs by using multiobjective optimization algorithms," in *Trans. Japan Soc. Aero. Space Science*, vol. 37, pp. 51–58, 2004.

[16] C. Coello Coello and A. Christiansen, "Multi-objective optimization of trusses using genetic algorithms," *Computers and Structures*, vol. 75, no. 6, pp. 647–660, 2000.

[17] M. P. Bendsøe and O. Sigmund, *Topology Optimization*. Springer, 2004.

[18] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, New York, 2003.

[19] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2005.

[20] G. Allaire, F. Jouve, and A.-M. Toader, "A level-set method for shape optimization," *Comptes Rendus Mathematique*, vol. 334, no. 12, pp. 1125–1130, 2002.

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

[22] G. Allaire and F. Jouve, "A level-set method for vibration and multiple loads structural optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 30-33, pp. 3269–3290, 2005.

[23] S. Amstutz and H. Andrä, "A new algorithm for topology optimization using a level-set method," *Journal of Computational Physics*, vol. 216, August 2006.

[24] T. Belytschko, C. Parimi, N. Moës, N. Sukumar, and S. Usui, "Structured extended finite element methods for solids defined by implicit surfaces," *International Journal for Numerical Methods in Engineering*, vol. 56, pp. 609–635, 2003.

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

[26] J. Chen, V. Shapiro, K. Suresh, and I. Tsukanov, "Parametric and topological control in shape optimization," in *Proceedings of IDETC/CIE*, (Philadelphia, Pennyslvania, USA), September 2006.

[27] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing," *Journal of Computational Physics*, vol. 183, pp. 83–116, November 2002.

[28] L. He, C.-Y. Kao, and S. Osher, "Incorporating topological derivatives into shape derivatives based level set methods," *Journal of Computational Physics*, vol. 225, pp. 891–909, July 2007.

[29] J. A. Norato, M. P. Bendsøe, R. B. Haber, and D. A. Tortorelli, "A topological derivative method for topology optimization," *Structural and Multidisciplinary Optimization*, vol. 33, pp. 375–386, 2007.

[30] M. Y. Wang and X. Wang, "Level set models for structural topology optimization," in *Proceedings of DETC'03*, (Chicago, Illinois, USA), September 2003.

[31] M. Y. Wang, X. Wang, and D. Guo, "A level set method for structural topology optimization," *Computational Methods in Applied Mechanics and Engineering*, vol. 192, pp. 227–246, 2003.

[32] X. Wang, M. Y. Wang, and D. Guo, "Structural shape and topology optimization in a level-set-based framework of region representation," *Structural and Multidisciplinary Optimization*, vol. 27, pp. 1–19, May 2004.

[33] 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*, vol. 221, pp. 395–421, January 2007.

[34] P. Papalambros and D. Wilde, *Principles of optimal design.* Cambridge University Press, 2 ed., 2000.

[35] M. Aoki, *Introduction to optimization techniques.* New York: Macmillian, 1971.

[36] C. J. K. Lee, T. Furukawa, and S. Yoshimura, "A human-like technique for design of engineering systems," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 14, pp. 1915–1943, 2005.

[37] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[38] 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*, vol. 227, pp. 5561–5581, 2008.

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

[40] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, no. 2, pp. 269–277, 1995.

[41] M. Sussman and E. Fatemi, "An efficient interface preserving level set redistancing algorithm and its applications to interfacial incompressible fluid flow," *SIAM J. Sci. Comput.*, vol. 20, pp. 1165–1191, 1999.

[42] J. Andersson, *Multiobjective optimization in engineering design*. PhD thesis, Department of Mechanical Engineering, Linkopings University, Sweden, 2001.

[43] Z. Liu, J. G. Korvink, and R. Huang, "Structural topology optimization: fully coupled level set method via femlab," *Structural and Multidisciplinary Optimization*, vol. 29, pp. 407–417, June 2005.

[44] T. Hisada, "Recent progress in nonlinear fem-based sensitivity analysis," *JSME International Journal*, vol. 38, no. 3, pp. 301–310, 1995.

[45] M. Tanaka and H. Noguchi, "Structural shape optimization of hyperelastic material by discrete force method," *Theoretical and Applied Mechanics Japan*, vol. 53, pp. 83–91, 2004.

[46] T. E. Bruns and D. A. Tortorelli, "An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms," *Int. J. Numer. Meth. Engng.*, vol. 57, pp. 1413–1430, 2003.

[47] J. Koski, *Multicriteria truss optimization*, ch. 9. Multicriteria optimization in engineering and in the sciences, New York: Plenum, 1988.

[48] M. Shimoda, H. Azegami, and T. Sakurai, "Multiobjective shape optimization of linear elastic structures considering multiple loading conditions," *JSME international journal. Ser. A, Mechanics and material engineering*, vol. 39, no. 3, pp. 407–414, 1996.