Structural Shape and Topology Optimization with Implicit and Parametric Representations

ZHANG, Jiwei

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Mechanical and Automation Engineering

December 2010

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Thesis/Assessment Committee

Professor Wang, Changling Charlie (Chair)

Professor Wang, Yu Michael (Thesis Supervisor)

Professor Hui, Kin-chuen (Committee Member)

Professor Chen, Yong Hua (External Examiner)

Abstract of thesis entitled:

Structural Shape and Topology Optimization with Implicit and Parametric Representations

Submitted by ZHANG, Jiwei

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in December 2010

Engineers have utilized CAE technique as an analysis tool to refine the engineering design over decades. However, CAE alone is not the key to open the door for the final goal. In order to achieve the practical solution to the real-time engineering problem, we need to integrate CAD, CAE and optimization techniques into a single framework.

In the problem of the structural optimizations, three categories of the approaches can be identified: *size*, *shape* and *topology optimizations*. For size optimization, explicit dimensions are usually chosen as the design variables, for example, the thickness of a beam or the diameter of a cylinder. For shape optimization, the shape related parameters of the geometrical boundary are always considered to be the design variables, like the positions of the control points for a Bezier curve. However, these two methods are lack of the capability to handle the topological changes of the geometry. On the contrary, topology optimization is the generalization of size and shape optimizations, which offers a more flexible and powerful tool to determine the best layout

of the materials and the topology to the design problem, and it is becoming increasingly important in the conceptual design phase. In other words, topology optimization gives one the inspiration for the locations where we put holes to reach the best design.

The material density method and the boundary-variation method are the popular methods adopted in both academia and industrial community. Even though the former method is dominant in industry, the latter method is more preferable these years owing to its boundary description nature. Undoubtedly, the level set based method is the most promising technique of the boundary-variation type. Scientists successfully developed the optimization algorithms based on the level set method (LSM) in the past few years. With the implicit representation of the LSM, topological changes of the design can be handled easily and the geometrical complexity is then reserved.

In this thesis, we put forward the algebraic level set (ALS) model with the consideration of the constructive solid geometry (CSG) model so that it is consistent with half-space primitive concept in CSG. Based on general shape derivative, we propose the general shape design sensitivity analysis (SDSA) formulations for general geometric primitives that are represented implicitly, such as line and circle primitives in two-dimensional space and plane primitive in three-dimensional space. We then extend the relevant formulations into corresponding parametrically represented primitives as they are widely used in today's mainstream CAD systems.

In the optimization algorithm part, apart from the general parametric steepest descent (ST) algorithm, we also study the least square (LSQ) based optimization algorithm. As a result, we can solve the

problem arisen from the variant dimensional sizes of the different design variables by using the weighted sensitivity information.

The optimal result given by conventional topology optimization usually involves tedious post-processing to form CAD geometry. Using our parameterizations with basic primitives and the proposed optimization algorithms, we can deliver comparatively complicated shapes with rich topological information. Therefore, the detail design could be conducted directly later.

The numerical examples for the design optimization problem are successfully implemented with both the implicit geometric representation (2D cases) and the parametric geometric representation (3D cases), which proves the feasibility of the proposed framework. The results show that both shape and topology optimizations of a design could be accomplished in a natural way.

摘要

雖然工程師利用 CAE 分析技術來提高產品設計品質已有多年的歷史，但 CAE 分析技術本身並不是實現最優設計的根本解決方案。爲了在工程實踐中獲得設計最優可行解，我們需要將 CAD、CAE 及優化技術集成到一個統一的框架下。

在結構優化領域中有三類基本的優化分支，即尺寸優化、形狀優化以及拓撲優化。對於尺寸優化而言，顯式表達的尺寸等參數可以作爲設計變數，例如梁的厚度或者是圓柱體的直徑等。對於形狀優化而言，幾何邊界所關聯的幾何參數可以作爲設計變數，例如貝賽爾曲線的控制點的位置等。但上述兩類方法的局限性在於無法有效實現幾何形狀的拓撲變化。與此相反，結構拓撲優化提供了一種決定結構設計中材料分佈及拓撲形狀定義的靈活而強大的工具，它是尺寸優化和形狀優化基礎上的一般性的優化方法並已經在產品概念設計中變得越來越爲重要。換而言之，結構拓撲優化爲我們如何在結構中開洞從而達到最優性能設計提供了靈感。

材料密度法以及變邊界法是學術界及工業界所廣泛採用的方法。儘管前者在工業界中處於主導地位，但後者由於其直接的邊界描述特性而越來越廣爲大家所接受。毫無疑問，在變邊界法當中，基於水平集方法的結構優化是最有前途的一種方法。在過去數年裡，科學家已經成功開發出了基於水平集方法的拓撲結構優化方法。由於水平集方法的隱式邊界描述特性，結構邊界的拓撲變化可以輕易的實現從而使得優化結果可以保留複雜的幾何特徵。

在本文中，在參考構造實體幾何模型（CSG）的基礎上我們提出一種代數水平集模型，其特點是與 CSG 中的半平面模型一脈相承。基於一般的形狀敏度的推導之上，我們提出了針對於一般性的基本隱式表達的幾何體素的形狀設計敏度分析列式。這些隱式表達體素包括二維空間中的直線體素和圓體素以及三維空間中的平面體素等。此外，我們還將形狀靈敏度分析推導推廣到在當今 CAD 系統中所廣泛使用的相應的參數化表達的幾何體素中。

在優化算法部分，除一般性的參數化最速下降法之外，我們還研究了基於最小二乘法的優化算法。其結果是利用了加權的敏度信息來解決了優化中不同類型的優化變數所產生的量綱不同的問題。

採用傳統的拓撲優化方法所帶來的問題之一就是結果需要繁瑣後處理以形成 CAD 幾何模型。採用我們所提出的參數化方法以及我們的優化算法，我們可以獲得具有豐富拓撲結構信息的複雜的結構形狀，而詳細設計可以在此之後直接進行。

對於結構優化設計問題，在本文中我們實現了採用隱式幾何表達的二維結構優化問題的算例及採用參數化幾何表達的三維結構優化算例。算例證明了所提出方法的可行性，同時也展示了結構設計中的形狀和拓撲變化可以自然的方式同時實現。

Acknowledgement

I would like to thank my supervisor, Professor Michael Yu Wang, for his patient guidance, helpful suggestions and great support during the past three years. One of his many ideas allowed this work to begin and his insight helps the work to be on the right track. I feel very fortunate to be one of his students.

I would like to extend my thanks to all members of the Computational Modeling and Design Laboratory at the Chinese University of Hong Kong. They are Dr. LIU Tong, Dr. ZHOU Shiwei, Dr. CHEN Shikui, Dr. WEI Peng, Dr. XIA Qi, Dr. XING Xianghua, Dr. HO Hon Shan, Dr. XIA Hongjian, Dr. LUO Junzhao, Dr. WANG Nianfeng, Dr. LUO Zhen, Dr. LIANG Sen, Mr. MA Jie, Mr. POON Ming Ki, Mr. ZHANG Weisheng, Mr. SHU Lei, Mr. TIAN Xuefeng, Mr. ZHOU Mingdong, Ms. LUI Fung Yee and Ms. LI Li.

This work is dedicated to my families and my dear friends,

for their support and patience,

and for their always being there with me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

From the industrial point of view, it is important for the developing new products using best material layout for any given objectives while satisfying various constraints at an early stage of a design process. This requirement actually urged the blossom of structural optimization methods during the last three decades. Structural optimization now is becoming a powerful tool in both academia and industrial community. Efficient procedures even commercial softwares for different types of engineering problems had been put forward consistently.

Three types of structural optimizations can be identified, which are size, shape and topology optimizations. As for size optimization, it is to find out the optimal design by changing the size feature variables such as the diameter of a circular hole or cross-sectional thickness of beam. Shape optimization, which could be considered as the generalization of size optimization, is mainly performed on continuum structures by modifying the predetermined boundaries with fixed topology to achieve the optimal designs. Topology optimization is to search for the optimal designs by determining the best locations and geometries of cavities in

the design domains, and it is a generalization of both size and shape optimizations.

1.1 Background

The emergence of computer-aided design (CAD) and the finite element method (FEM) has greatly enhanced engineer's ability to evaluate potential designs and predict the performance before actual fabrication (Figure (1.1)). Researchers and engineers have been devoted to the integration of the shape and topology optimization methods fully into the existing CAD and CAE environments for decades. The key reason for doing this is to take full advantages of existing CAD and CAE environments to realize genuine design automation. However, till today, no single software can fully achieve this ultimate goal. To achieve the optimal design in a structural optimization problem, it usually depends on the modification of the boundaries of a structure, or the material distribution of the model. To date, three of the most influential structural shape and topology approaches are: conventional parametric shape optimization, material based topology optimization and boundary-variation based shape and topology optimization.

1.1.1 Conventional Parametric Shape Optimization

In the conventional CAD-based shape optimization, due to the history-based mechanism of CAD system (one feature built on another), it is not easy to handle topological changes during the optimization, therefore the topology of a design mostly remains the same as defined at the beginning.

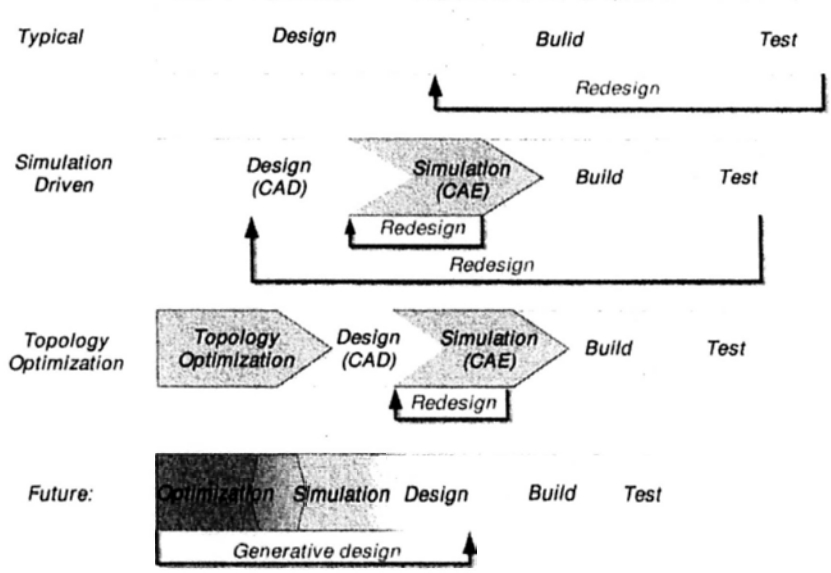


Figure 1.1: Performance evaluation methodologies during different phases

Originally, the shape design parameters were associated directly with the finite element nodal coordinates, however, the mesh irregularities and massive number of shape design parameters caused the decadence of this approach followed by the rising of three more sophisticated methods, namely, the mesh parameterization method [1–3], the natural design variable method [2, 4], and the solid modeling method [5–7].

In the mesh parameterization method, the shape design variables are reduced to a set of parameters that define the positions of master points, which are then used to define geometric entities of the mesh. However, it suffers two major drawbacks. Firstly, designer must define the model with master points, which turns out to be a very tedious job with complicated geometric models. Secondly, mesh degradation might occur since the topology of finite element mesh is usually predetermined.

The natural design variable method requires defining the fictitious

boundary conditions which will be computationally expensive. Moreover, the optimal shape is defined by deformation mode shapes and therefore not compatible with CAD data structures.

The solid modeling method, which is built upon CAD systems, is advantageous compared with the previous two methods. It is a natural integral part of CAD system. Therefore, it could be intuitively interpreted by designers. In [5, 6], the shape design sensitivity is discussed with both discrete method and continuum method, but only two-dimensional cases are studied. Notably, in [7], more general three-dimensional cases are given using the proposed design sensitivity calculation with respect to dimensional constraints under the assumption that one can calculate the dimensional variation through variational geometry equations [8–15]. This approach, however, still demands relating FE nodes to geometric boundaries. Moreover, the feature-based model construction process sets the limitations on topological changes during shape optimization process.

1.1.2 Material Based Topology Optimization

Due to the difficulty of handling topological changes during structural optimization process, topology optimization is regarded as one of the most challenging tasks in conceptual design and only a few methods have been successfully implemented for industrial use. The birth of the finite element based structural topology optimization in the late 1980s opened a new era in structural optimization field, and various approaches have been developed since then.

The homogenization method is introduced in [16] (also see [17, 18]), it is one of the main approach which employs micro-perforated

composites as permissible designs to relax the originally ill-posed 0/1 (void/solid) optimization problem [19]. A relationship between microstructure parameters and effective material properties is established by the homogenization theory. The optimal material distribution may be obtained by determining the size parameters of the microstructure using an optimality criteria procedure. Even though homogenization method has a strong mathematical background, the results given by this theory have little practical value since the shape of the optimal structures are too complicated that they can not be fabricated at all. Furthermore, numerical instabilities may introduce "non-physical" artifacts in the results and make the designs sensitive to variations in the loading. Nevertheless, the homogenization approach can be considered as a milestone in topology optimization because the solution provides mathematical bounds to the theoretical performance of a structure.

Based on the homogenization method, further research works have nurtured more fruitful results. One of the most famous and well accepted methodology is the "solid isotropic material with penalization" (SIMP) approach for its conceptual simplicity [19, 20]. It is based on the assumption that each element contains an isotropic material (density assumed to be constant within each element). The elements are used to discretize the design domain with element densities considered as the design variables. The material properties are modeled to be proportional to the relative material density raised to some power. The power law-based approach has been widely applied to topology optimization problems with multiple constraints, multiple physics, and multiple materials. However, the adoption of finite element grids to represent the shape of structures causes step-like boundary which needs

tedious post-processing works if one wants to use the optimal results directly in existing CAD systems.

Another well-known method called the "evolutionary structural optimization" (ESO) has been proposed by Xie and Steven [21,22], which is based on the concept of gradually removing inefficient material from structure to achieve an optimal design. The method was developed for various problems of structural optimization including stress considerations, frequency optimization, and stiffness constraints. The ESO method uses a fixed model with standard finite elements to represent the initial design domain while the so-called optimum design is found as a subset of the initial set of finite elements. A key process of this method is to use an appropriate criterion to assess the contribution of each element to the specified behavior (response) of the structure and subsequently to remove some elements with the least contribution (usually known as hard kill). A similar approach called "reverse adaptivity" was proposed by Reynolds et al. [23], in which a fixed percentage of relatively under-stressed material is removed to find approximately fully stressed structures. Essentially, both the ESO and the reverse adaptivity are homotopy methods based on material hard kills. Recently, a modified version of the ESO called the "bi-directional evolutionary structural optimization method" (BESO) has been well studied by Huang and Xie [24]. This approach allows material to be removed and added simultaneously (usually known as soft kill), and is similar to the SIMP method to some extent. However, one important difference is that the soft-kill BESO uses discrete design variables while the SIMP allows for continuous material densities. Both the ESO and the BESO are essentially based on an evolutionary strategy focusing

on local consequences but not on the global optimum, and they are typically computationally expensive.

The "bubble method" is proposed by Eschenauer and co-workers [25, 26]. In this method, the so-called characteristic functions of the stresses, strains and displacements are employed to determine the placements or insertion of holes in known shape at optimal positions in the structure, namely, it modifies the structural topology in a prescribed manner. In such case, the design for a given topology is settled before its further changes.

A common characteristic of abovementioned methods is the focus on material properties and the adoption of finite element mesh to represent geometry, the performances of these methods are greatly limited by the weak capability of boundary representation. These element-based methods cannot represent an exact boundary of a design. Real geometry information of the design boundary of a structure during the optimization cannot be obtained directly, and "step effect" is always an issue with the optimal results which makes it hard to become a true integral part of CAD systems.

1.1.3 Boundary Variation Based Shape and Topology Optimization

Instead of using material density as design variables directly, boundary variation methods, typically, the "level set method" (LSM) is another popular type of approach in structural optimization community, which has been widely used in various fields because of its strong geometry handling capabilities. In this method, the design variable is actually the boundaries of a given structure. Instead of using explicit bound-

ary representation under Lagrangian perspective, in the level set based structural topology optimization, boundaries are represented by the implicit level set function and are propagated by the level set equation under the Eulerian framework. Since the LSM can handle the topological changes of moving interfaces naturally (splitting or merging) and flexibly, it offers a tool for simultaneous shape and topology optimization.

Sethian and Wiegmann [27] are the pioneers who extend the level set method developed by Osher and Sethian [28–30] into the structural optimization field. In their works, the boundaries are captured on a fixed Eulerian mesh and the Von Mises equivalent stress is used to improve the structural rigidity. The LSM is employed for tracking the motion of the structural boundaries under a speed function and handling the presence of potential topological changes. An explicit jump immersed interface method is used for computing the solution of the elliptic problem (the Lamé equations) in complex geometries with a regular mesh. Osher and Santosa [31] propose a level set based computational framework for frequencies optimization problems. They use functional gradients to calculate the velocity of the level set function and deal with optimization problems with geometrical constraints.

In the series of papers [32–35], the level set based structural optimization method with shape gradient are developed, and these significant works laid solid foundation for more general structural problems. Even though the shape sensitivity theory itself does not offer the mechanism of topological changes, the LSM fills the gap between conventional shape optimization and topology optimization and gives a solution to unify these two different categories. As stated in [34],

a boundary-based method with the capability of handling topological changes has the most promising potential. It is a more direct approach than material-based methods. For example, in general it allows more explicit representation of any features to be incorporated in the design.

Although level set based structural optimization is quite promising, till now only steepest descent optimization algorithm could be used as the optimization strategy, conventional mathematical programming methods are not quite useful within this framework. Due to this limitation, some variants of the level set based optimization method are developed, which usually focus on the parameterization of the level set equation.

In [36,37], the radial basis functions (RBFs) are used to parameterize the level set function. By assuming the artificial time only dependent on coefficients, the level set function is converted from the partial differential equation (PDE) into a system of ordinary differential equations (ODEs). In later works [38–41] RBFs based level set is employed in a different manner. Instead of solving the level set equation, this variant parameterizes the level set function with RBFs interpolations. The interpolated level set equation is then substituted into the conventional shape derivative formulation, which makes it possible to calculate the sensitivity regarding design parameters. This parameterization enables the adoption of a lot of mature optimization algorithms like Optimality Criteria (OC) and Method of Moving Asymptote (MMA). Using the chain rule, design sensitivities with respect to parameters are obtained, and the level set function can be updated by varying parameters according to sensitivity analysis and the optimization algorithms. This method differs from the conventional discrete LSM because it needs

boundary velocity only and requires boundary integration. The latest variant regarding RBFs level set based structural optimization is found in [42, 43], by dividing the design domain into multiple overlapped sub-domains, the large scale linear system solving procedure could be replaced by the solving process of a set of small scale linear equations at each optimization step.

A piecewise constant level set (PCLS) method is implemented to solve the structural optimization problems in [40, 44, 45]. In this approach, a piecewise density function is defined over the design domain. This function is regarded as the link between the level set function and the objective function. The PCLS method retains advantages of the conventional level set method and it is free of the Courant-Friedrichs-Lewy (CFL) condition and reinitialization. More importantly, this method allows new holes to nucleate so it is useful in two-dimensional topology optimization.

A finite element based level set method (FELSM) is proposed in [46, 47] which uses FEM other than finite difference method (FDM) to solve the PDE. The merit is that mesh grids could be much more flexible compared with the uniform grid requirement in FDM based level set method, and this set of mesh could be used for both front evolution and structural analysis purpose. Therefore it enables arbitrary initial domain with any kind of predefined fixed obstacles (like a circle). The drawback of this method is obvious, too. In this method, only first order accurate solving scheme has been found, higher ordered solving schemes are not available. In terms of accuracy, traditional FDM based solving scheme is far superior to the proposed FELSM.

In [48, 49], B-spline functions are used to parameterize the level set

function. Also, the shape sensitivities with respect to shape parameters as well as to B-spline coefficients are obtained. This makes it capable of handling free form shapes with topological changes, meanwhile, fits the shape maintenance and layout problem into the same framework.

Besides the LSM, some researchers are concentrating on using other boundary variation approaches such as utilizing the CAD geometry to perform shape and topology optimization directly. One of the most notable approach is Armstrong's work [50]. In this work, sensitivity based formulations regarding dimensional parameters are derived with the adjoint method. Furthermore, both shape and small topological changes are observed in the numerical examples. However, no dramatic topological changes are achievable in this work because the topology changing capability relies heavily on the commercial CAD system.

1.2 Research Objectives

Modern performance-driven product development circle can be broken down into three chains: specifications, conceptual design and detailed design. The conceptual design phase and detailed design phase could actually be considered to correspond to topology optimization and CAD based shape optimization problems respectively. However, these two approaches seem incompatible in some sense mainly because the most popular topology optimization approaches currently adopted by industry are material density based which are unable to offer CAD ready geometry in a direct manner. Even though boundary variation based shape and topology methods are now gaining popularity, the boundary representations of the optimal models delivered with these

methods are still not CAD based geometry.

Instead of using complex geometric entities to represent a model, engineers tend to choose more simple primitives to construct a solid prototype at conceptual design phase. Unfortunately, current mainstream history-based systems are mostly detail-oriented. CAD based shape optimization process can not be conducted until after the complex CAD models are conceived. This actually constrains design edits and limits model configuration. Size and shape optimizations performed thereafter can not change the performance based on the current design, and prediction of even better designs cannot be explored within the same scope. Due to these limitations, the current methodology often delays discovery of configuration issues in new designs until much later in the design process; consequently, expensive engineering workarounds and costly redesign are required to correct such issues in some critical designs.

The emerging direct modeling tools foster creativity and substantive engineering early in the design process by offering more flexible geometry manipulations compared with the feature-based CAD systems in which the "top-down" design feature tree of the CAD model is a necessity. This kind of direct editing capabilities help engineers evaluate the best approaches before detailed design commences, which provides us a new solution to unify shape and topology optimizations within CAD environment. The improved design methodology presented in this work attempts to bridge the methodological gap between conceptual and detailed design, or we can say the gap between current topology optimization and shape optimization. By restricting shape representations to a few types of simple geometric entities, the shape design sensitivity anal-

ysis (SDSA) using gradient information could be obtained accordingly. Topological changes as a critical issue in traditional shape optimization in feature-based CAD environment now could be greatly relaxed. This proposed CAD geometry based shape and topology optimization approach allows an optimal solid to be delivered at conceptual design phase using simple primitives and leaves all detailed design behind.

In CAD systems, the two popular approaches for representing a solid are: the constructive solid geometry (CSG) approach that was first introduced by Requicha and Voelcker at the University of Rochester [51]; and the boundary representation (B-Rep) approach by Braid in Cambridge [52]. Comparing to CSG representation, B-Rep is more flexible and has a much richer operation set, this makes B-Rep a more favorable choice for modern CAD systems.

To conduct shape and topology optimization within CAD context, we need to solve two issues: the selection of appropriate primitives to represent a solid and the shape design sensitivity analysis (SDSA) calculation with respect to individual primitive's parameters.

One distinction between mesh model and CAD model is that, CAD model is comprised of multiple regular geometric entities, such as planar surfaces and circular surfaces which are preferable to designers for representing complex solid. Due to this reason, the primitives chosen in this dissertation are simple analytical primitives, which are implicitly represented with CSG or parametrically represented with B-Rep. The critical issue in hindering the conventional shape optimization from the topological changes is solved.

The shape optimization within continuum structure framework using the SDSA is widely accepted. It normally involves the derivative

of a real-valued objective function regarding a set of selected parameters used to define a design. The SDSA calculation provides a rigorous basis for most deterministic optimization algorithms. One should note that the dependence of objective functions on design is not always explicit except for a small set of geometrical objectives, for example, the cross-sectional area of a circular beam that can be explicitly expressed as $S = \pi r^2$ with r indicating the radius of the beam. The sensitivity analysis of S with respect to r is quite straightforward and it can be explicitly derived as $dS/dr = 2\pi r$. For most of the structural problems such as stiffness and stress optimization, or other physical objectives, the dependence is usually expressed implicitly. Supposedly, we are given with a general objective function J, the SDSA regarding the design could be obtained by the FEM based structural analysis, as shown in [53–55]. This is the most time-consuming chain in the optimization algorithm, and the SDSA computation with respect to our chosen parameterizations will be elaborated in later context.

1.3 Contributions and Outline

For the boundary-variation based shape and topology optimization, the parameterization method adopted is critical to the SDSA computation. If the discrete LSM is chosen as the geometric representation and the evolution impulsion, then the free boundary Γ is the design parameter itself. If the parametric LSM is used, such as RBF LSM, then the coefficients or the knot positions as internal parameters could be considered as the design variables and the SDSA can be carried out accordingly. Notably, the SDSA for very simple constructively represented shapes

can be referred to in [48,49,56–59] but no further discussions about the extension and generalization of more flexible shape and topology optimizations are available with implicitly represented geometries, not to mention the unified shape and topology optimization framework with parametric representation. Thus, it motivates us to establish a theoretical and computational frameworks for both geometric representations that are suitable for shape and topology optimization.

In this dissertation, we proposed a more general algebraic level set framework, in which the geometry is constructively and dynamically represented. Also, with the proposed geometric representation and the shape derivative theory, we derived the corresponding SDSA formulations with respect to the chosen internal design parameters. "Internal" means the variables are governing the analytic equation, in contrast with the "external" parameters which are commonly dimensional constraints. We also show that the SDSA formulations are always computable no matter how the solid is represented (implicitly or parametrically), as long as the normal design velocity field regarding design parameters is available. These proposed SDSA formulations for our parameterizations could naturally be extended to CAD-based parametric primitives with B-rep data structure under the condition of a proper geometry updating scheme.

Since the structural optimization problems are always coupled with constraints (volume, local stress, etc.), an augmented Lagragian multiplier method (ALM) is used to manage the extra constraints in the optimization procedure. Based on the SDSA formulations, we developed the problem-specific steepest decent (ST) optimization algorithms. Due to the problem of the variant dimensions in the design

parameters, a least square (LSQ) gradient-based optimization method is studied. The efficiencies of both proposed algorithms are compared and we prove that LSQ based algorithm is efficient. Comparisons are made between the results (2D cases) and the benchmarks delivered with discrete level set. The results prove the correctness and efficiency of our proposed framework. With our methods, one can avoid the reinitialization process and velocity extension process in the traditional level set method.

This dissertation is organized as follows. In Chapter 2, the shape derivative and the structural optimization problems are discussed with the review of the level set based structural optimization framework. In Chapter 3, we present a parameterization method for solving the level set equation, i.e., the algebraic level set model. Formulations are studied in details and the corresponding SDSA are given. The extension of the sensitivity calculation for the parametric primitives is also elaborated in this chapter. In Chapter 4, the ALM is first studied to impose constraints, followed by the study of the parametric ST algorithm. Since the internal parameters either embedded with the translational or the rotational characteristic, we further propose a LSQ based optimization algorithm to resolve the dimensional problems with fixed time step size. The numerical examples are given in Chapter 5, both 2D (implicitly represented) and 3D (parametrically represented) examples with hybrid primitives are investigated with the structured finite element mesh. Efficiencies of the parametric ST based algorithm and the LSQ based algorithm are compared. The performance difference between the proposed ALS framework and discrete level set framework is discussed. Conclusions and future work are given in the last chapter.

□ **End of chapter.**

Chapter 2

Structural Optimization Problem and Level Set Method

In this chapter, we first review the definition of the optimization problem for continuum structures, and the basic ideas of shape derivatives derived from material derivatives. Our concentration in this chapter is the introduction of shape-variation based shape derivative formulation using material derivatives, which is the foundation of shape design sensitivity analysis. Also, the level set method (LSM) will be discussed. The shape and topology optimization framework using LSM will be reviewed in the last section of this chapter.

2.1 Structural Optimization Problems for Continuum Structure

There is an important class of structural optimization problem needed to determine the shape and material layout of a specific structure. Several kinds of structural objective functionals, such as the minimum mean compliance problem, the maximum natural frequency problem, and the minimum stress problem, subject to certain constraints like volume constraint or stress constraint, have practical and strong engineering backgrounds and are always the favorite subjects in both academia and industry.

As mentioned in last chapter, these kinds of problems cannot always be reduced to a formulation that can express the structural shape as a design function in an explicit way. In this case, the design variable is not a function but the geometry of the physical domain of interest [53–55].

2.1.1 Structural Optimization Problem

The shape design sensitivity analysis (SDSA) discussed in this dissertation only concerns linear structural systems. For nonlinear structural systems one can refer to [60]. Generally speaking, the general structural optimization problem is defined as:

$$\text{minimize} \quad J(\boldsymbol{u}, \Omega) = \int_{\Omega} F(\boldsymbol{u})\, d\Omega \tag{2.1}$$

The goal of structural optimization is to minimize the objective functional $J(\boldsymbol{u})$ for a specific physical or geometric type described by $F(\boldsymbol{u})$, where $\Omega \subseteq R^d$ ($n = 2$ or 3) is the domain occupied by isotropic elastic material, $\boldsymbol{u}$ is the state variable which may be the kinematically

admissible displacement field or its higher order derivatives like strain field or stress field, and this is a standard notion of structural optimization [54]. Here, we only present the optimization formulation for minimum mean compliance objective, but the approach would apply to other general problems.

For mean compliance problem, $F(\boldsymbol{u})$ could be expressed as:

$$F(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{u})^T \boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{u}). \tag{2.2}$$

where $\boldsymbol{u}$ is the displacement field of the structure under certain loads and boundary conditions, $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ is the strain field, and $\boldsymbol{D}$ is the elasticity matrix. The integrand $F(\boldsymbol{u})$ is actually the strain energy density.

The design variable is the shape of the structure, Ω, and the objective function J depends on Ω in two ways: the explicit dependence since the integral is defined on Ω and the implicit dependence through $\boldsymbol{u}$ which is the solution of the following state equations defined on Ω:

$$\begin{aligned} -\operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}) &= \boldsymbol{f} \quad && \text{in} \quad \Omega, \\ \boldsymbol{u} &= 0 \quad && \text{on} \quad \Gamma_D, \\ \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \boldsymbol{g} \quad && \text{on} \quad \Gamma_N. \end{aligned} \tag{2.3}$$

where $\boldsymbol{\sigma}$ is the stress field, $\boldsymbol{f}$ the body force, $\boldsymbol{g}$ the traction force acting on Neumann (or force) boundary Γ_N. The boundary of structure is denoted by Γ which could be subcategorized into three different characteristic parts and following relationships are satisfied:

$$\begin{aligned}
\Gamma &= \Gamma_D \cup \Gamma_N \cup \Gamma_f, \\
\Gamma_D \cap \Gamma_N &= \emptyset, \\
\Gamma_N \cap \Gamma_f &= \emptyset, \\
\Gamma_D \cap \Gamma_f &= \emptyset, \\
\Gamma_D &\neq \emptyset.
\end{aligned}$$

where Γ_D is the Dirichlet (or displacement) boundary and Γ_f means the traction free boundary. The general boundary and force condition can be illustrated as shown in Figure (2.1).

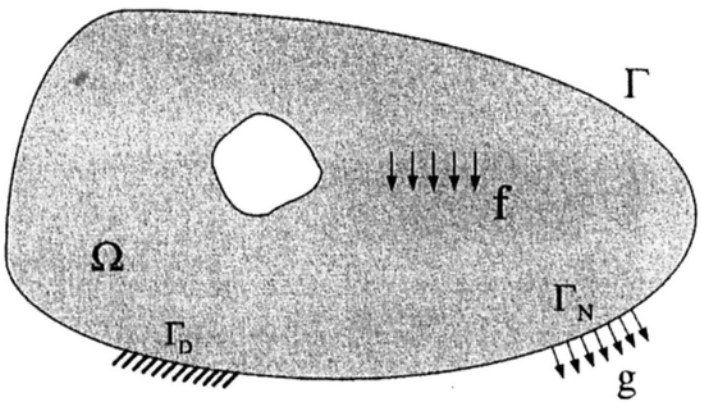


Figure 2.1: A general setting of boundary and force condition

Eq. (2.3) can be expressed in the energy bilinear and the load linear form of $\boldsymbol{u} \in U$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in U. \tag{2.4}$$

with

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega c(\boldsymbol{u}, \boldsymbol{v})\, d\Omega = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{u})^T \boldsymbol{D} \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega \tag{2.5}$$

$$l(\boldsymbol{v}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega + \int_{\Gamma_N} \boldsymbol{g} \cdot \boldsymbol{v}\, d\Gamma, \tag{2.6}$$

where U is the space of kinematically admissible displacements and can be defined as:

$$U = \{\boldsymbol{u} : u_i \in H^1(\Omega), \boldsymbol{u} = 0 \text{ on } \Gamma_D\} \tag{2.7}$$

2.1.2 Material Derivatives of State Variable

To carry out boundary-variation based shape and topology optimization, the shape design sensitivity analysis (SDSA) framework should be developed in the first place to build the relationship between the variation in shape and the resulting variations in functionals. Under this context, the shape of the domain Ω is considered as a continuous medium and also the design variable for any type of structural problems. Using the material derivative idea of continuum mechanics and the adjoint variable method, shape derivative as the consequence can be derived.

By introducing the scalar variable t, the initial structural shape Ω is changed to the deformed shape Ω_t with t denoting the amount of shape change in the design variable direction. This shape perturbation can be construed as a mapping from Ω to Ω_t with the mapping denoted as T, as shown in Figure (2.2). The mapping T is expressed as:

$$\begin{aligned} \boldsymbol{x}_t &= T(\boldsymbol{x}, t), \\ \Omega_t &= T(\Omega, t). \end{aligned} \tag{2.8}$$

The deforming process of the domain from Ω to Ω_t may be viewed as a dynamic process of deforming a continuum, with t playing the role of artificial time. The time design velocity can be defined as:

$$\boldsymbol{V}(\boldsymbol{x}, t) = \frac{d\boldsymbol{x}_t}{dt} = \frac{dT(\boldsymbol{x}, t)}{dt} = \frac{\partial T(\boldsymbol{x}, t)}{\partial t}. \tag{2.9}$$

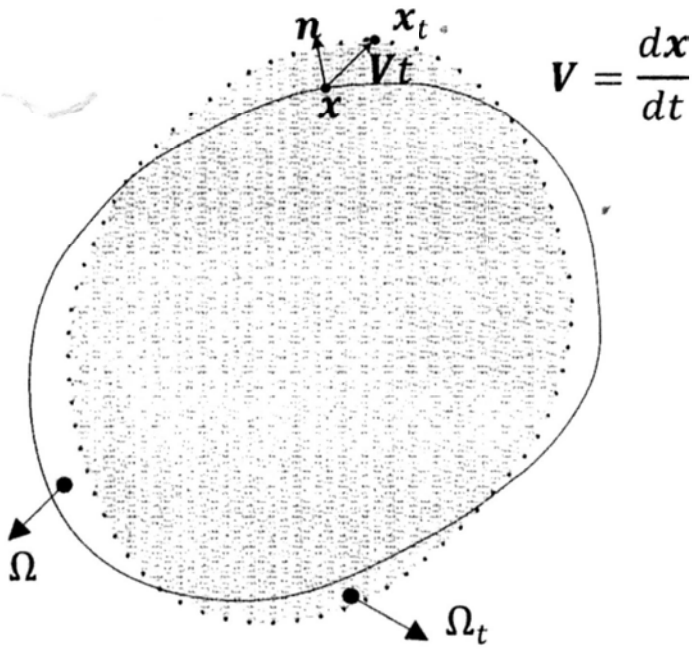


Figure 2.2: Domain perturbation induced by mapping T

where the last equality is because of the fact that the initial point $\boldsymbol{x}$ does not depend on t. Under sufficient regularity conditions, such as that T^{-1} exists, then the design velocity is given as:

$$\boldsymbol{V}(\boldsymbol{x}, t) = \frac{\partial T}{\partial t}(T^{-1}(\boldsymbol{x}_t, t), t). \tag{2.10}$$

Therefore, the shape deformation can be described by the initial-value problem

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{V}(\boldsymbol{x}, t), \qquad \boldsymbol{x}_0 = \boldsymbol{x}. \tag{2.11}$$

When $d\boldsymbol{x}_t/dt = 0$ is reached, the optimal result is obtained.

Suppose $\boldsymbol{z}$ is a smooth solution to Eq. (2.3) defined in Ω. Then the material derivative of z is defined as:

$$\dot{\boldsymbol{z}} = \boldsymbol{z}' + \nabla \boldsymbol{z} \cdot \boldsymbol{V}, \tag{2.12}$$

where $\boldsymbol{z}'$ is the partial derivative of $\boldsymbol{z}$ with respect to t and $\boldsymbol{V}$ is the

velocity vector:

$$\boldsymbol{V} = \frac{d\boldsymbol{x}}{dt},$$

or in three-dimensional form,

$$\boldsymbol{V} = \left[\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}\right]^T.$$

One attractive feature of the partial derivative with respect to t is that, with an assumption of smoothness, the differentiation order between it and the spatial derivative are interchangeable, i.e.,

$$\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}\right)' = \left(\frac{\partial \boldsymbol{z}'}{\partial \boldsymbol{x}}\right). \tag{2.13}$$

2.1.3 Shape Derivative of Objective Functionals

In order to find the shape derivative of an objective functional, two Lemmas regarding the computation of the material derivative of both domain functional and boundary functional are to be given. Both functionals are integrals and depend on the parameter t.

Lemma 1 *Let ψ_1 be a domain integral over Ω*

$$\psi_1 = \int_\Omega f(\boldsymbol{u}) d\Omega$$

where f is a regular function defined on Ω. The material derivative of ψ_1 is

$$\dot{\psi}_1 = \int_\Omega f'(\boldsymbol{u}) d\Omega + \int_\Gamma f(\boldsymbol{u}) V_n d\Gamma$$

where V_n is the normal velocity on boundary Γ.

Note that the conclusion given in the above Lemma only requires the design velocity information on the boundary which provides a simple and easy way for the computation of sensitivity information. Further,

for the design velocity $\boldsymbol{V}$, only the normal velocity V_n is needed, which accounts for the shape deformation and is the bridge between shape optimization theory and level set model to be discussed later. The tangential part of $\boldsymbol{V}$, however, has no effect on the variation effect of a domain.

Lemma 2 *Consider a boundary integral over* Γ,

$$\psi_2 = \int_\Gamma g(\boldsymbol{u}) d\Gamma$$

where g *is a regular function defined on* Γ. *The material derivative of* ψ_2 *is*

$$\dot{\psi}_2 = \int_\Gamma g'(\boldsymbol{u}) d\Gamma + \int_\Gamma \big(\nabla g(\boldsymbol{u}) \cdot \boldsymbol{n} + \kappa g(\boldsymbol{u})\big) V_n d\Gamma.$$

where $\boldsymbol{n}$ *is the unit normal in* Γ *and* κ *the mean curvature of* Γ.

According to Lemma 1, we take the material derivative of the objective functional J:

$$\dot{J} = \int_\Omega F'(\boldsymbol{u}) d\Omega + \int_\Gamma F(\boldsymbol{u}) V_n d\Gamma, \tag{2.14}$$

where

$$F'(\boldsymbol{u}) = \frac{\partial F}{\partial \boldsymbol{u}} \cdot \boldsymbol{u}'. \tag{2.15}$$

Computing $\boldsymbol{u}'$ is generally nontrivial but fortunately this term can be eliminated if we resort to the material derivative of state equations and the adjoint method.

Firstly, we take the material derivative of both sides of the variational equation Eq. (2.4) by applying Lemma 1 and Lemma 2,

$$\dot{a}(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega c'(\boldsymbol{u},\boldsymbol{v})d\Omega + \int_\Gamma c(\boldsymbol{u},\boldsymbol{v})V_n d\Gamma, \tag{2.16}$$

$$\begin{aligned}\dot{l}(\boldsymbol{v}) = &\int_\Omega \boldsymbol{f}\cdot\boldsymbol{v}'d\Omega + \int_\Gamma \boldsymbol{f}\cdot\boldsymbol{v}V_n d\Gamma + \int_{\Gamma_N} \boldsymbol{g}\cdot\boldsymbol{v}'d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\boldsymbol{g}\cdot\boldsymbol{v})\cdot\boldsymbol{n} + \kappa\boldsymbol{g}\cdot\boldsymbol{v}\big)V_n d\Gamma\end{aligned} \tag{2.17}$$

It should be noted that $\boldsymbol{f}' = 0$ and $\boldsymbol{g}' = 0$ are used in the derivation because we assume that both $\boldsymbol{f}$ and $\boldsymbol{g}$ are independent on the design.

Consider the integrand of the first integral on the right hand side of Eq. (2.16):

$$\begin{aligned}c'(\boldsymbol{u},\boldsymbol{v}) &= \big(\boldsymbol{\varepsilon}(\boldsymbol{u})^T\boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{v})\big)' \\ &= \big(D_{ijkl}\varepsilon_{ij}(\boldsymbol{u})\varepsilon_{kl}(\boldsymbol{v})\big)' \\ &= D_{ijkl}\big(\varepsilon'_{ij}(\boldsymbol{u})\varepsilon_{kl}(\boldsymbol{v}) + \varepsilon_{ij}(\boldsymbol{u})\varepsilon'_{kl}(\boldsymbol{v})\big)\end{aligned} \tag{2.18}$$

According to Eq. (2.13), the shape derivative of ε can be transformed as follows:

$$\begin{aligned}\varepsilon'_{ij}(\boldsymbol{u}) &= \frac{1}{2}\left(\left(\frac{\partial u_i}{\partial x_j}\right)' + \left(\frac{\partial u_j}{\partial x_i}\right)'\right) \\ &= \frac{1}{2}\left(\left(\frac{\partial u'_i}{\partial x_j}\right) + \left(\frac{\partial u'_j}{\partial x_i}\right)\right) \\ &= \varepsilon_{ij}(\boldsymbol{u}'),\end{aligned} \tag{2.19}$$

and similarly

$$\varepsilon'_{kl}(\boldsymbol{v}) = \varepsilon_{kl}(\boldsymbol{v}'), \tag{2.20}$$

Substituting Eq. (2.19) and (2.20) into Eq. (2.18) produces

$$\begin{aligned}c'(\boldsymbol{u},\boldsymbol{v}) &= D_{ijkl}\big(\varepsilon_{ij}(\boldsymbol{u}')\varepsilon_{kl}(\boldsymbol{v}) + \varepsilon_{ij}(\boldsymbol{u})\varepsilon_{kl}(\boldsymbol{v}')\big) \\ &= c(\boldsymbol{u}',\boldsymbol{v}) + c(\boldsymbol{u},\boldsymbol{v}'),\end{aligned} \tag{2.21}$$

so the material derivative of the bilinear form becomes

$$\dot{a}(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega c(\boldsymbol{u}',\boldsymbol{v})d\Omega + \int_\Omega c(\boldsymbol{u},\boldsymbol{v}')d\Omega + \int_\Gamma c(\boldsymbol{u},\boldsymbol{v})V_n d\Gamma\,. \tag{2.22}$$

From Eq. (2.22) and Eq. (2.17), we have

$$\begin{aligned}
&\int_\Omega c(\boldsymbol{u}',\boldsymbol{v})d\Omega + \int_\Omega c(\boldsymbol{u},\boldsymbol{v}')d\Omega + \int_\Gamma c(\boldsymbol{u},\boldsymbol{v})V_n d\Gamma \\
&= \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v}'d\Omega + \int_\Gamma \boldsymbol{f}\cdot\boldsymbol{v}V_n d\Gamma \\
&\quad + \int_{\Gamma_N} \boldsymbol{g}\cdot\boldsymbol{v}'d\Gamma + \int_{\Gamma_N} \big(\nabla(\boldsymbol{g}\cdot\boldsymbol{v})\cdot\boldsymbol{n} + \kappa\boldsymbol{g}\cdot\boldsymbol{v}\big)V_n d\Gamma.
\end{aligned} \tag{2.23}$$

Note that

$$\int_\Omega c(\boldsymbol{u},\boldsymbol{v}')d\Omega = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v}'d\Omega + \int_{\Gamma_N} \boldsymbol{g}\cdot\boldsymbol{v}'d\Gamma, \tag{2.24}$$

which is the variational identity. Hence these terms can be canceled out from Eq. (2.23) and the following equation is obtained:

$$\begin{aligned}
a(\boldsymbol{u}',\boldsymbol{v}) &= \int_\Omega c(\boldsymbol{u}',\boldsymbol{v})d\Omega \\
&= \int_\Gamma \big(\boldsymbol{f}\cdot\boldsymbol{v} - c(\boldsymbol{u},\boldsymbol{v})\big)V_n d\Gamma \\
&\quad + \int_{\Gamma_N} \big(\nabla(\boldsymbol{g}\cdot\boldsymbol{v})\cdot\boldsymbol{n} + \kappa\boldsymbol{g}\cdot\boldsymbol{v}\big)V_n d\Gamma\,.
\end{aligned} \tag{2.25}$$

Next, we construct the adjoint equation:

$$a(\boldsymbol{u}_a,\boldsymbol{v}_a) = \int_\Omega \frac{\partial F}{\partial \boldsymbol{u}}\cdot\boldsymbol{v}_a d\Omega, \quad \forall\,\boldsymbol{v}_a \in U\,, \tag{2.26}$$

where $a(\cdot,\cdot)$ is the bilinear form defined in Eq. (2.5) and the subscript a means "adjoint" variables. Since the test function $\boldsymbol{v}_a$ can be selected arbitrarily as long as it belongs to U, we can replace it with $\boldsymbol{u}'$ ($\boldsymbol{u}' \in U$) and change Eq. (2.26) into

$$a(\boldsymbol{u}_a,\boldsymbol{u}') = \int_\Omega \frac{\partial F}{\partial \boldsymbol{u}}\cdot\boldsymbol{u}'d\Omega\,. \tag{2.27}$$

Similarly, we can replace $\boldsymbol{v}$ in Eq. (2.25) with $\boldsymbol{u}_a$:

$$\begin{aligned} a(\boldsymbol{u}', \boldsymbol{u}_a) = &\int_\Gamma \big(\boldsymbol{f} \cdot \boldsymbol{u}_a - c(\boldsymbol{u}, \boldsymbol{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\boldsymbol{g} \cdot \boldsymbol{u}_a) \cdot \boldsymbol{n} + \kappa \boldsymbol{g} \cdot \boldsymbol{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.28}$$

Comparing Eq. (2.28), (2.27), and (2.14) and noting that the bilinear functional is symmetric, we can eliminate all the terms related to $\boldsymbol{u}'$ from Eq. (2.14):

$$\begin{aligned} \dot{J} = &\int_\Gamma \big(F(\boldsymbol{u}) + \boldsymbol{f} \cdot \boldsymbol{u}_a - c(\boldsymbol{u}, \boldsymbol{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\boldsymbol{g} \cdot \boldsymbol{u}_a) \cdot \boldsymbol{n} + \kappa \boldsymbol{g} \cdot \boldsymbol{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.29}$$

It is worthwhile to note that in the derivation we assume the Dirichlet boundary Γ_D cannot move in its normal direction. This means that V_n is zero on Γ_D and the shape derivative should be

$$\begin{aligned} \dot{J} = &\int_{\Gamma_N \cup \Gamma_f} \big(F(\boldsymbol{u}) + \boldsymbol{f} \cdot \boldsymbol{u}_a - c(\boldsymbol{u}, \boldsymbol{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\boldsymbol{g} \cdot \boldsymbol{u}_a) \cdot \boldsymbol{n} + \kappa \boldsymbol{g} \cdot \boldsymbol{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.30}$$

Substituting Eq. (2.2) into the adjoint equation Eq. (2.26), we obtain

$$a(\boldsymbol{u}_a, \boldsymbol{v}_a) = a(\boldsymbol{u}, \boldsymbol{v}_a), \quad \forall\, \boldsymbol{v}_a \in U, \tag{2.31}$$

which means that the adjoint variable $\boldsymbol{u}_a$ is same as the state variable $\boldsymbol{u}$. Therefore, the minimum mean compliance problem is a self-adjoint problem and the shape derivative can be derived as:

$$\begin{aligned} \dot{J} = &\int_{\Gamma_N \cup \Gamma_f} \big(\boldsymbol{f} \cdot \boldsymbol{u} - \frac{1}{2} \boldsymbol{\varepsilon}(\boldsymbol{u})^T \boldsymbol{D} \boldsymbol{\varepsilon}(\boldsymbol{u})\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\boldsymbol{g} \cdot \boldsymbol{u}) \cdot \boldsymbol{n} + \kappa \boldsymbol{g} \cdot \boldsymbol{u}\big) V_n d\Gamma \,. \end{aligned} \tag{2.32}$$

It has been assumed that Γ_N is fixed while Eq. (2.32) is being derived, so the integral over Γ_N vanishes. This assumption will be used

in this study, unless otherwise specified. From now on, we will use the symbol Γ to replace Γ_N for simplification reason, thus Eq. (2.32) will be rewritten as:

$$\dot{J} = \int_{\Gamma} GV_n d\Gamma \tag{2.33}$$

where G is known as the shape gradient density [53] and its expression is

$$G = \boldsymbol{f} \cdot \boldsymbol{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{u})^T \boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{u}). \tag{2.34}$$

2.2 Discrete Level Set Method for Free-Form Shape Evolution

The level set method (LSM) is a numerical technique developed for capturing front propagation. It was first introduced in [28], and further developed in [29, 30, 61]. One of the notable merits of the LSM is the employment of Hamilton-Jacobi approach that enables numerical computations of a time-dependent equation for involving free-form shapes (curves for 2D and surface for 3D) on a fixed Cartesian grid by embedding the shapes into a higher dimensional geometry. Known as the *Eulerian formulation*, the LSM overcomes the conventional *Lagrangian formulation* of boundary propagation which evolves boundary in an explicit way and is incapable of handling topological changes of shapes (splitting or merging). The advantages of the LSM make it widely applied in different fields, from image processing to fluid mechanics, from computer animation to combustion [30]. The fundamental concept of the LSM is described here to provide necessary background. Later in Chapter 3, the basic formulations will be obtained starting from the basics of the level set equation.

2.2.1 Level Set Method for Boundary Representation

In the level set framework, the boundary of a shape is embedded as the zero contour of a one-higer dimensional function $\Phi(\boldsymbol{x})$, which is usually stated as the level set function, and the boundary itself is the zero isocontour or the zero level set of the level set function. Using this model, mathematically, three different parts can be defined:

$$\begin{cases} \Phi(\boldsymbol{x}) > 0 & \boldsymbol{x} \in \Omega, \\ \Phi(\boldsymbol{x}) = 0 & \boldsymbol{x} \in \Gamma, \\ \Phi(\boldsymbol{x}) < 0 & \boldsymbol{x} \in D \backslash \bar{\Omega}, \end{cases} \tag{2.35}$$

where D is the design domain, $\bar{\Omega} = \Omega \cup \Gamma$ is the union of the boundary and its inner part. Figure (2.3) shows the implicit representation of a level set function. Throughout this dissertation, we include the boundary with the interior region Ω where $\Phi(\boldsymbol{x}) > 0$, unless otherwise specified.

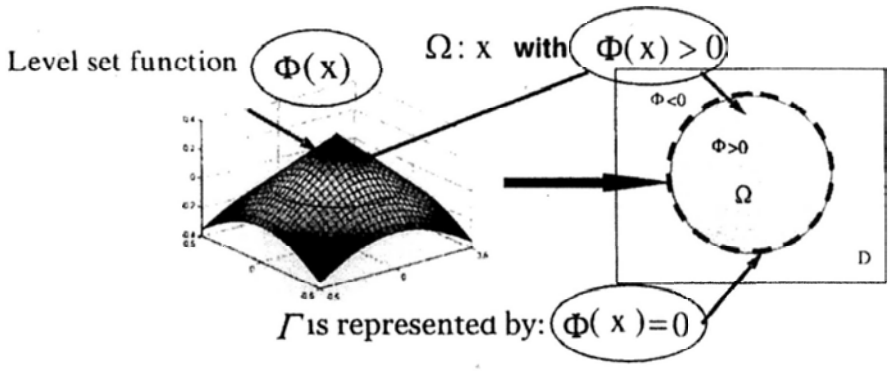


Figure 2.3: Level set function definitions

If above definition is adopted, then the gradient of Φ can be expressed as:

$$\nabla\Phi = \left(\frac{\partial\Phi}{\partial x}, \frac{\partial\Phi}{\partial y}, \frac{\partial\Phi}{\partial z}\right). \tag{2.36}$$

One should note that for any point on the front, the vector $\nabla\Phi$ points to the same direction as the unit outward normal does on the same point. Therefore the unit normal vector of any point on the front can be stated as:

$$\boldsymbol{n} = -\frac{\nabla\Phi}{|\nabla\Phi|}. \tag{2.37}$$

For our definition of the level set equation requires $\Phi > 0$ representing the inside of shape, thus the minus is a necessity, but this actually depends on how one chooses to define the inside and outside of the geometry.

The mean curvature of the interface is defined as the divergence of the normal $\boldsymbol{n}$:

$$\kappa = \nabla \cdot \left(-\frac{\nabla\Phi}{|\nabla\Phi|}\right). \tag{2.38}$$

Actually, since the level set function defines the interface in a domain of one higher-dimension, instead of defining $\boldsymbol{n}$ and κ on the interface only, we can use Eq. (2.37) and (2.38) to define them everywhere on the domain.

2.2.2 Hamilton-Jacobin Equation

To drive the one-dimensional higher function Φ to move over the Euclidean grids, we need to add dynamics. To do this, we need to introduce the velocity to excite this evolution, which is the chain that bonds the structural optimization and level set framework together. During the entire propagating process of the implicit surface, the zero isocontour will always be the boundary of the shape, which means any point $\boldsymbol{x}$ on the interface can be identified from

$$\Phi(\boldsymbol{x}(t), t) \equiv 0, \qquad \forall \boldsymbol{x} \in \Gamma \tag{2.39}$$

The time differentiation of Eq. (2.39) with respect to t gives the so-called level set equation:

$$\frac{\partial \Phi}{\partial t} + \boldsymbol{V} \cdot \nabla \Phi = 0, \tag{2.40}$$

which is Hamilton-Jacobin type partial differential equation and is also referred to as convection equation. In structural optimization problems, only the motion in the normal direction has influence on the change of geometry, we can rewrite the $\boldsymbol{V} \cdot \nabla \Phi$ as $V_n |\nabla \Phi|$, then Eq. (2.40) becomes:

$$\frac{\partial \Phi}{\partial t} - V_n |\nabla \Phi| = 0. \tag{2.41}$$

This partial differential equation defines the motion of the interface where $\Phi(\boldsymbol{x}) = 0$ under the velocity V_n. It is an Eulerian formulation of the interface evolution, since the interfaces is captured by the implicit function Φ as opposed to being tracked in the Lagrangian formulation. The greatest advantage of implicit representation lies in the fact that it is able to deal with topological changes, such as splitting and merging of the boundary, in a natural manner. In addition, with implicit representation, boolean operations can be carried out quite easily.

2.2.3 Discrete Computational Schemes

In Eq. (2.40), the level set equation only contains the first order derivative of Φ, which leads to a lot of numerical schemes were devised to solve this type of hyperbolic PDE. Lax-Friedrichs scheme, Roe-Fix scheme, Godunov scheme and upwind differencing scheme [29, 30] are the most famous ones, and the last two schemes are identical. Here, we only introduce the upwind differencing scheme. The underlying idea of this scheme is that the information always spreads from the known area to

the unknown area, and what the scheme does is to use the information to propagate the implicit surface.

A simple first-order upwind scheme for two-dimensional problems is given by

$$\Phi_{ij}^{n+1} = \Phi_{ij}^{n} - \triangle t(max((V_n)_{ij}, 0)\nabla^{+} + min((V_n)_{ij}, 0)\nabla^{-}), \tag{2.42}$$

where

$$\begin{aligned}\nabla^{+} = \{&max(D_{ij}^{-x}, 0)^2 + min(D_{ij}^{+x}, 0)^2 \\ &+max(D_{ij}^{-y}, 0)^2 + max(D_{ij}^{+y}, 0)^2\}^{1/2},\end{aligned} \tag{2.43}$$

$$\begin{aligned}\nabla^{-} = \{&max(D_{ij}^{+x}, 0)^2 + min(D_{ij}^{-x}, 0)^2 \\ &+max(D_{ij}^{+y}, 0)^2 + max(D_{ij}^{-y}, 0)^2\}^{1/2},\end{aligned} \tag{2.44}$$

Δt is the time step, and it should satisfy the Courant-Friedrichs-Lewy (CFL) conditions [30]:

$$\Delta t \leq \frac{min(\Delta x, \Delta y)}{max|(V_n)_{ij}|}, \tag{2.45}$$

Δx and Δy are grid spaces in horizontal and vertical directions, D_{ij}^{x} , D_{ij}^{y} are forward (+) and backward (-) finite difference operators defined as follows:

$$\begin{aligned} D_{ij}^{+x} &= (\Phi_{i+1,j} - \Phi_{ij})/\Delta x, \\ D_{ij}^{-x} &= (\Phi_{ij} - \Phi_{i-1,j})/\Delta x, \\ D_{ij}^{+y} &= (\Phi_{i,j+1} - \Phi_{ij})/\Delta y, \\ D_{ij}^{-y} &= (\Phi_{ij} - \Phi_{i,j-1})/\Delta y. \end{aligned} \tag{2.46}$$

This is a first-order accurate discretization of the spatial operator. To improve the accuracy, the idea of essentially non-oscillatory (ENO) introduced in [62] has been used [63–65]. In ENO, the velocity is still used to decide the upwind direction but the accuracy can be improved significantly. In [66], Liu et al. pointed that the ENO is overkill

in smooth regions where the data are well behaved and proposed a weighted ENO (WENO) method that takes a convex combination of three ENO approximations. Later, Jiang and Shu [67] improved the WENO method to obtain the optimal fifth-order accuracy in smooth regions of the flow. In [68], Jiang and Peng extended WENO to the Hamilton-Jacobi framework.

2.2.4 Reinitialization

It is obvious to notice that a given boundary can be represented by infinite number of level set functions, as long as the smoothness of these functions are good enough. Therefore, theoretically, one can choose any kind that satisfies the stability and accuracy criteria. But in practice, if we intend to get highly accurate numerical results, the selection of level set function must be very careful, namely, it should be neither too steep nor too flat, especially near the interface [28, 30]. According to these, the so-called signed distance function (SDF) is the best option.

A signed distance function is an implicit function Φ with

$$\Phi(\boldsymbol{x}) = \begin{cases} d & \boldsymbol{x} \in \Omega, \\ 0 & \boldsymbol{x} \in \Gamma, \\ -d & \boldsymbol{x} \in D \backslash \bar{\Omega}, \end{cases} \tag{2.47}$$

where

$$d = |\boldsymbol{x} - \boldsymbol{x}_c|, \tag{2.48}$$

with $\boldsymbol{x}_c$ indicating the closest point on the boundary to $\boldsymbol{x}$. The noticeable characteristic of SDF is that $|\Phi(\boldsymbol{x})| = 1$.

As the boundary evolves, the shape of the level set function will generally drift away from the feature as a signed distance function. In other words, it might cause numerical instability, thus rise the question on how to maintain the SDF characteristic. "Reinitialization" is the key to this question.

There are various techniques for reinitialization [29,30]. In this section, the method of solving the reinitialization equation is introduced. The reinitialization equation [65] is

$$\frac{\partial \Phi}{\partial t} = S(\Phi_0)(1 - |\nabla \Phi|), \tag{2.49}$$

where S is the sign function. It is evident that when this equation is solved to a steady state, which means

$$\frac{\partial \Phi}{\partial t} = 0, \tag{2.50}$$

the signed distance function is rebuilt since

$$|\nabla \Phi| = 1. \tag{2.51}$$

Equation (2.49) can be written in the following form

$$\frac{\partial \Phi}{\partial t} + \boldsymbol{w} \cdot \nabla \Phi = S(\Phi_0), \tag{2.52}$$

where

$$\boldsymbol{w} = S(\Phi_0)(\nabla \Phi / |\nabla \Phi|). \tag{2.53}$$

For numerical purposes it is useful to smooth the sign function. In [65], S is smoothed as:

$$S(\Phi_0) = \frac{\Phi_0}{\sqrt{\Phi_0^2 + \alpha^2}}, \tag{2.54}$$

where α is a small constant, which can be specified as the mesh size. Peng et al. [61] suggest that

$$S(\Phi) = \frac{\Phi}{\sqrt{\Phi^2 + \alpha^2 |\nabla \Phi|^2}} \tag{2.55}$$

is a better choice, especially when the initial Φ_0 is too flat or steep, i.e., when $|\nabla \Phi_0|$ is far from 1.

2.3 Structural Optimization with Discrete Level Set Method

In boundary-variation based structural shape and topology optimization such as minimum mean compliance optimization, the boundary is often the most crucial part. Actually, the interior region and exterior region of the design domain are of no value and the velocity field does not impose any physical meaning. We only need to evaluate the boundary integral with design velocity information when we compute the shape derivative. Nevertheless, within level set framework, if we want to add dynamics to the level set equation, the design velocity field has to be defined across the entire domain or at least within a narrow bound along the shape's interface. To conquer this obstacle, it is often more convenient to introduce the one-dimensional *Heaviside function* with the one-dimensional variable Φ. The Heaviside function defined here has the form of

$$H(\Phi(\boldsymbol{x})) = \begin{cases} 1 & if \quad \Phi(\boldsymbol{x}) \geq 0 \\ 0 & if \quad \Phi(\boldsymbol{x}) < 0 \end{cases} \tag{2.56}$$

The first-order directional derivative of the Heaviside function in the normal direction $\boldsymbol{n}$ is defined as the *Dirac delta function*, which can be

stated as:

$$\hat{\delta}(\Phi(\boldsymbol{x})) = \nabla H(\Phi(\boldsymbol{x})) \cdot \boldsymbol{n}. \tag{2.57}$$

Note that this delta function is nonzero only on the boundary Γ, thus we have

$$\hat{\delta}(\Phi(\boldsymbol{x})) = H'(\Phi(\boldsymbol{x}))|\nabla\Phi(\boldsymbol{x})|. \tag{2.58}$$

In one spatial dimension, the delta function is defined as the derivative of the one dimensional Heaviside function:

$$\delta(\Phi) = H'(\Phi). \tag{2.59}$$

This allows us to rewrite Eq. (2.58) as:

$$\hat{\delta}(\Phi(\boldsymbol{x})) = \delta(\Phi)|\nabla\Phi(\boldsymbol{x})|. \tag{2.60}$$

Therefore, the shape derivative in Eq. (2.33) which involves the boundary integral can be reformulated as

$$\dot{J} = \int_D \delta(\Phi)G(\Phi)V_n|\nabla\Phi|d\Omega. \tag{2.61}$$

where G can also be considered as a function of Φ. If steepest descent schemes is employed, we can get the steepest descent direction by setting

$$V_n = -G(\Phi), \tag{2.62}$$

this would yield

$$\dot{J} = -\int_D \delta(\Phi)G(\Phi)^2|\nabla\Phi|d\Omega \leq 0. \tag{2.63}$$

This is actually a natural extension of design velocity field, as illustrated in Figure. (2.4)

Once the finite element structural analysis has been performed and the velocity has been calculated based on stresses and strains, one

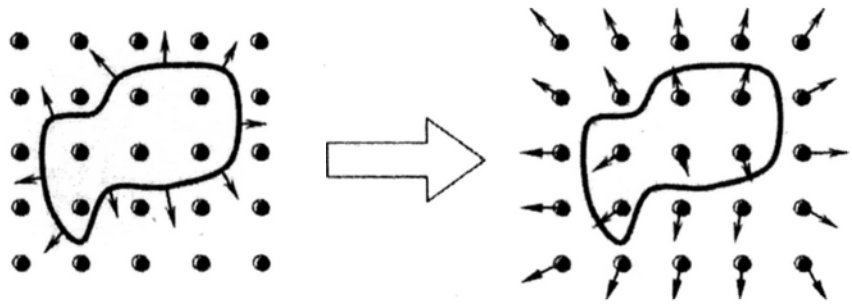


Figure 2.4: Natural design velocity extension

can evolve the boundary by solving the level set equation, and a new boundary of the structure is obtained. If this new structure is not satisfying, a new analysis step begins; otherwise, the optimization process terminates. For details, readers are referred to [32–35].

Two observations are worth to be noticed. First of all, for different structural problems, the only thing that is different in the shape derivative is the form of shape gradient density in Eq. (2.34). Secondly, it is worth to notice that the shape derivative we derived earlier does not give any capability or information to change the topology of structural geometry, it is the dissipation mechanism offered by the level set method gives the flexibility of strong topology changing capacity.

□ **End of chapter.**

Chapter 3

Sensitivity Analysis for Implicit and Parametric Representations

The classical discrete level set based shape and topology optimization framework offers a versatile approach to change structural shape and topology simultaneously. Since the entire design domain must be discretized with Euler grids for level set evolution, the Courant-Friedrichs-Lewy (CFL) condition [69] must be satisfied to ensure the stability of the numerical scheme. In applying the classical level set method for structural topology optimization, the implementation also requires a careful choice of upwind scheme, natural extension of design velocity and reinitialization algorithm [28, 30]. Hence, the numerical considerations of discrete computation have severely restricted the primary advantages of the LSM in structural optimization.

From the engineering point of view, even though the level set based optimization produces optimal solutions with clear boundary informa-

tion as opposed to the density based results given by the SIMP method, there is still no CAD-ready geometric information available but discretized mesh. On the one hand, one of the key observations shows that, for most parts of the optimal solutions given by discrete level set based optimization, their shapes can be approximated by basic geometries such as linear and circular primitives in 2D, or planar and cylindrical primitives in 3D, as shown in Figure (3.1). Therefore, it is natural to think about making full use of basic geometric primitives and their combinations to perform shape optimizations while keeping the topology changing capability. The rest of this chapter will concentrate on the derivation of the shape design sensitivity analysis (SDSA) formulations for different geometric representations.

Figure 3.1: Beam-like optimal structure

3.1 Introduction to Solid Modeling

Solid modeling techniques adopted by contemporary CAD community started in 1970s. With the constructive solid geometry (CSG) representation, a solid is represented as a set of theoretic regularized boolean expressions of bounded primitive solid objects [70–72] such as block, sphere or cylinder. The boundary representation (B-rep), on the other

hand, describes only the oriented surface of a solid as a data structure that is composed of vertices, edges, and faces.

In later sections of this chapter, we will introduce the computational framework for the sensitivity analysis formulations with implicitly represented geometric primitives under the CSG framework. By adding "dynamics", we come up with the algebraic level set (ALS) model. Moreover, we will extend the sensitivity analysis formulations to parametrically represented primitives under the B-Rep framework.

3.2 Shape Design Sensitivity Analysis for Implicit Geometric Representation

3.2.1 Algebraic Level Set Parameterization

For any intricate solids, they are actually defined as boolean combinations of a set of half-space primitives governed by implicit equations. The half-space model itself offers a mathematically rigorous, easy-to-understand approach to solid modeling [72].

A half-space primitive in R^2 or R^3 is such one that divides the spatial space into two semi-infinite regions. For instance, an unbounded straight line in 2D space divides R^2 into two parts. Such a straight line half-space model can be expressed as $h(\boldsymbol{x}) = ax + by + c$ [71, 73]. This definition is consistent with the level set definition by considering the parameters in the half-space model as artificial time dependent variables to introduce dynamics, therefore, we come up with the concept of the ALS model. Based on typical 2D and 3D half-space primitives, we can have a set of basic ALS primitives that could represent various

kinds of planar and spatial shapes needed in most mechanical design applications.

Suppose we have two independent ALS functions $\varphi_1(\boldsymbol{x})$ and $\varphi_2(\boldsymbol{x})$, the intersection of the interior regions of them can be expressed as: $\Omega = \{\boldsymbol{x} \in D | \Phi(\boldsymbol{x}) \geq 0\}$, where $\Phi(\boldsymbol{x}) = min(\varphi_1, \varphi_2)$. Also, the union of the interior regions of the said functions could be written as $\Omega = \{\boldsymbol{x} \in D | \Phi(\boldsymbol{x}) \geq 0\}$, where $\Phi(\boldsymbol{x}) = max(\varphi_1, \varphi_2)$. With the above boolean description, complex geometries could be easily constructed. Figure (3.2) shows a rectangular shape with a centered hole delivered with the boolean operators given at the bottom part of the figure, where $\Omega_i, \quad i = 1, \ldots, 5$ are primitive shapes represented im-

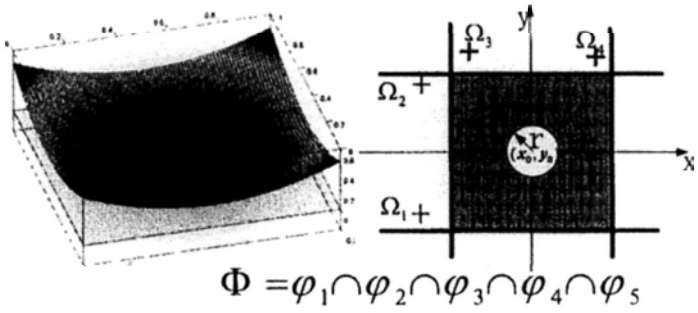


Figure 3.2: Complex shape defined by multiple ALS primitives

plicitly by the corresponding ALS function (primitive). Here, in terms of algebraic, we mean the corresponding implicit primitive could be represented algebraically.

Generally speaking, an ALS primitive could be expressed as $\varphi = \varphi(\boldsymbol{x}, \boldsymbol{s})$, where $\boldsymbol{x}$ is the spatial coordinate vector and $\boldsymbol{s} = \{s^1, s^2, \ldots, s^M\}$ is the vector containing all its relevant parameters with M as the number of all parameters.

Similar to the SDF in the conventional level set framework, here we

choose the normalized ALS with the SDF property to make it consistent with the level set concept. The extra benefit for $|\varphi(\boldsymbol{x})| = 1$ is that we could reduce the computational errors when conducting boolean operations between different ALS primitives during the geometry construction process.

The basic ALS primitives chosen here are line and circle primitives in 2D, and plane, cylinder and sphere primitives in 3D. A set of the possible ALS expressions for the selected primitives are

Line primitive,

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{\sqrt{a^2 + b^2}}(ax + by + c); \tag{3.1}$$

Circle shape primitive,

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = \alpha[\sqrt{(x - x_0)^2 + (y - y_0)^2} - r]; \tag{3.2}$$

Plane primitive,

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{\sqrt{a^2 + b^2 + c^2}}(ax + by + cz + d); \tag{3.3}$$

Sphere primitive,

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = \alpha[\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} - r]; \tag{3.4}$$

Cylinder primitive,

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = \alpha[\sqrt{(x - x_0)^2 + (y - y_0)^2} - r]; \tag{3.5}$$

The cylinder primitive shown in Eq. (3.5) is just a particular situation with axis parallel to z axis, and he general configurations are not discussed here. Primitives with coordinate transformation will not be discussed here for simplicity.

In above equations, $\alpha = 1$ indicates the inside of the primitive is void, whereas $\alpha = -1$ indicates the inside of the primitive is solid. Therefore, for any given complex shape $\Omega = \cup \cap \Omega_i$, it could be simply expressed implicitly via a single piecewise defined level set function Φ constructed from the corresponding group of ALS primitives $\{\varphi_i, i = 1, \cdots, N\}$ with the same set of boolean operators for building Ω. Moreover, every primitive only gets involved once during the construction process. This Φ, however, is only piecewise differentiable. Suppose there is such a group of differentiable boolean operators that all ALS primitives could be shown in a single constructed equation, then Φ could be expressed as $\Phi(\boldsymbol{x}, \boldsymbol{S}) = \Phi(\varphi_1, \cdots, \varphi_i, \cdots, \varphi_N)$, and Ω could be expressed as $\Omega = \{\boldsymbol{x} \in D | \Phi(\boldsymbol{x}, \boldsymbol{S}) \geq 0\}$. Here $\varphi_i = \varphi_i(\boldsymbol{x}, \boldsymbol{s}_i)$, $\boldsymbol{s}_i = \{s_i^j, j = 1, \cdots, M_i\}$, and $\boldsymbol{S} = \{s_1, \cdots, s_{N_s}\}$ is the parameter vector with N_s indicating the number of all the parameters collected from all individual ALS primitives. In later context, all primitives will be considered as independent of other primitives unless otherwise mentioned, which means no geometric constraints are imposed between primitives and the design parameters of each primitive are independent. Mathematically, this can be expressed as:

$$\boldsymbol{S} = \boldsymbol{s}_1 \cup \boldsymbol{s}_2 \cup \cdots \cup \boldsymbol{s}_{N-1} \cup \boldsymbol{s}_N$$

$$\boldsymbol{s}_i \cap \boldsymbol{s}_k = \emptyset, \quad i \neq k,$$

$$\boldsymbol{s}_i \cap \boldsymbol{s}_k = \boldsymbol{s}_i, \quad i = k,$$

where $i = 1, \ldots, N$, $k = 1, \ldots, N$.

3.2.2 Sensitivity Analysis Formulations

Since Hamilton-Jacobin PDE equation is time dependent, we could assume that the constructed Φ is dependent on t only via $\boldsymbol{S}$, so that we can separate the spatial coordinates from the time dependent variables. With this assumption, the resulting constructed level set function $\Phi(\boldsymbol{x})$ can be rewritten as:

$$\begin{aligned} \Phi(\boldsymbol{x}, t) &= \Phi(\boldsymbol{x}, \boldsymbol{S}(t)) \\ &= \Phi(\varphi_1, \varphi_2, \cdots, \varphi_N). \end{aligned} \tag{3.6}$$

For the assumption of the differentiability of Φ that we made earlier is certifiable, by substituting Eq. (3.6) into Eq. (2.41), we could get

$$\begin{aligned} V_n &= \frac{1}{|\nabla\Phi|}\frac{\partial\Phi}{\partial t} \\ &= \frac{1}{|\nabla\Phi|}\left(\frac{\partial\Phi}{\partial \boldsymbol{S}}\right)\cdot\dot{\boldsymbol{S}} \\ &= \frac{1}{|\nabla\Phi|}\sum_{I=1}^{N_s}\left(\frac{\partial\Phi}{\partial s_I}\right)\cdot\dot{s}_I. \end{aligned} \tag{3.7}$$

Therefore, Eq. (2.33) becomes

$$\begin{aligned} \dot{J} &= \int_\Gamma G(\boldsymbol{x}, \boldsymbol{u}) V_n d\Gamma \\ &= \int_\Gamma G\frac{1}{|\nabla\Phi|}\sum_{I=1}^{N_s}\left(\frac{\partial\Phi}{\partial s_I}\right)\cdot\dot{s}_I d\Gamma. \end{aligned} \tag{3.8}$$

Note that the differentiability of Φ on the entire domain now is relaxed to the differentiability on the boundary. Since J is dependent on t via $\boldsymbol{S}$, using the chain rule, one could get

$$\dot{J} = \sum_{I=1}^{N_s}\frac{\partial J}{\partial s_I}\dot{s}_I. \tag{3.9}$$

By comparing Eq. (3.8) with Eq. (3.9), we have

$$\frac{\partial J}{\partial s_I} = \int_\Gamma G \frac{1}{|\nabla \Phi|} \frac{\partial \Phi}{\partial s_I} d\Gamma. \tag{3.10}$$

Now let us introduce a Lemma obtained in [48, 49]

Lemma 3 *Suppose* $\Phi = \Phi(\varphi_1, \cdots, \varphi_N)$ *is a R-function of a set of implicit primitives* $\varphi_1, \cdots, \varphi_N$, *if* $\varphi_i(x_0) = 0$ *and* $\varphi_j(x_0) \neq 0$ *for* $\forall j \neq i, \quad j = 1, \cdots, N$, *then* $D(\Phi)|_{x_0} = D(\varphi_i)|_{x_0}$.

Using this Lemma, if s_I is related to φ_i and x_0 is a point on the corresponding Γ_i of the primitive, we have $\frac{\partial \Phi}{\partial s}|_{x_0} = \frac{\partial \varphi_i}{\partial s}|_{x_0}$. Therefore, by denoting Γ_i as the portion of boundary Γ corresponding to a primitive shape Ω_i and by assuming that the construction of Φ preserves the gradient of each ALS primitive on the boundary, i.e., $\nabla \Phi|_{\Gamma_i} = \nabla \varphi_i|_{\Gamma_i}$, the Eq. (3.10) can be written as

$$\frac{\partial J}{\partial s_I} = \int_{\Gamma_i} G \frac{1}{|\nabla \varphi_i|} \frac{\partial \varphi_i}{\partial s_I} d\Gamma = \sum_{k=1}^{T} \int_{\Gamma_k} G \frac{1}{|\nabla \varphi_i|} \frac{\partial \varphi_i}{\partial s_I} d\Gamma \tag{3.11}$$

with $\Gamma_i = \cup \Gamma_k$. Here we refer to these s_I-related boundary segments Γ_k as *active boundary segments* for parameter s_I and refer to the implicit representations of the corresponding boundary segments as *active primitives* for s_I. Further, by denoting an arbitrary $s_I \in S$ as s, the SDSA formulation with respect to this parameter could be reformulated as

$$\frac{\partial J}{\partial s} = \sum_{k=1}^{T} \int_{\Gamma_k} G \frac{1}{|\nabla \varphi_i|} \frac{\partial \varphi_i}{\partial s} d\Gamma \tag{3.12}$$

where φ_i is the ALS primitive that contains s, and T is the total number of boundary segments that belong to φ_i.

This localized SDSA formulation with respect to parameter s in Eq. (3.12) requires that the active boundary point is only related to

one primitive except for a set of measure zero, but it is not a necessary condition for the validity of the SDSA computation regarding a parameter.

In later sections, we will see this conclusion not only applies to the proposed ALS model but applies to the B-Rep model with boundary primitives represented parametrically as well.

3.2.3 Sensitivity for Basic 2D Algebraic Level Set Primitives

In the previous section, we have come up with a set of ALS functions. We hope that, by combining the ALS functions through boolean operators and by introducing artificial time t to add dynamics, the shape and topology of a structure could evolute over time simultaneously.

The ALS line primitive in Eq. (3.1) has three independent parameters. Obviously there exists a dependency between the three parameters, we should eliminate the extra degree of freedom (DOF) to compute the SDSA, as we have done with the set of implicit ALS primitives in the following subsections.

Algebraic Level Set Line Primitive

If we only consider the two DOFs of the ALS line primitive, according to Figure (3.3), we can reformulate Eq. (3.1) as:

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = x \sin\alpha + y \cos\alpha + c, \tag{3.13}$$

the corresponding parameters are α and c.

The derivatives of φ with respect to the design parameters and the

norm of $\nabla\varphi$ are:

$$\begin{aligned} \frac{\partial\varphi}{\partial\alpha} &= x\cos\alpha - y\sin\alpha, \\ \frac{\partial\varphi}{\partial c} &= 1, \\ |\nabla\varphi| &= 1. \end{aligned} \tag{3.14}$$

For any objective function J with corresponding shape gradient density G , we have the SDSA formulations with respect to all parameters

$$\begin{aligned} \frac{\partial J}{\partial\alpha} &= \int_\Gamma G(x\cos\alpha - y\sin\alpha)d\Gamma, \\ \frac{\partial J}{\partial c} &= \int_\Gamma G d\Gamma. \end{aligned} \tag{3.15}$$

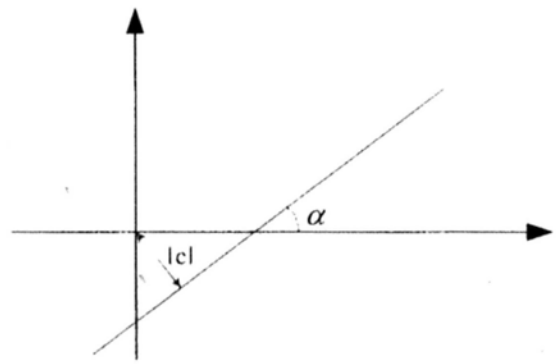


Figure 3.3: Two DOFs of ALS line primitive

Algebraic Level Set Circle Primitive

The ALS circle (inside is void) function is

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = -r + \sqrt{(x - x_0)^2 + (y - y_0)^2}, \tag{3.16}$$

the corresponding parameters are x_0, y_0 and r.

The derivatives of φ with respect to the design parameters and the norm of $\nabla\varphi$ are:

$$\begin{aligned}
\frac{\partial\varphi}{\partial x_0} &= \frac{x_0 - x}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}, \\
\frac{\partial\varphi}{\partial y_0} &= \frac{y_0 - y}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}, \\
\frac{\partial\varphi}{\partial r} &= -1, \\
|\nabla\varphi| &= 1.
\end{aligned} \tag{3.17}$$

For any objective function J with corresponding shape gradient density G , we have the SDSA formulations with respect to all parameters

$$\begin{aligned}
\frac{\partial J}{\partial x_0} &= \int_\Gamma G \frac{x_0 - x}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} d\Gamma = \frac{1}{r}\int_\Gamma G(x_0 - x) d\Gamma, \\
\frac{\partial J}{\partial y_0} &= \int_\Gamma G \frac{(y_0 - y)}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} d\Gamma = \frac{1}{r}\int_\Gamma G(y_0 - y) d\Gamma, \\
\frac{\partial J}{\partial r} &= -\int_\Gamma G d\Gamma.
\end{aligned} \tag{3.18}$$

3.2.4 Sensitivity for Basic 3D Algebraic Level Set Primitives

Algebraic Level Set Plane Primitive

The ALS plane function is

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = (x\cos\alpha + y\sin\alpha)\cos\beta + z\sin\beta + c, \tag{3.19}$$

the corresponding design parameters are α , β and c.

The derivatives of φ with respect to the design parameters and the

norm of $\nabla\varphi$ are:

$$\begin{aligned}
\frac{\partial\varphi}{\partial\alpha} &= -x\sin\alpha\cos\beta + y\cos\alpha\cos\beta, \\
\frac{\partial\varphi}{\partial\beta} &= -(x\cos\alpha + y\sin\alpha)\sin\beta + z\cos\beta, \\
\frac{\partial\varphi}{\partial c} &= 1, \\
|\nabla\varphi| &= 1.
\end{aligned} \tag{3.20}$$

For any objective function J with corresponding shape gradient density G , we have the SDSA formulations with respect to all parameters

$$\begin{aligned}
\frac{\partial J}{\partial\alpha} &= \int_\Gamma G(y\cos\alpha\cos\beta - x\sin\alpha\cos\beta)d\Gamma, \\
\frac{\partial J}{\partial\beta} &= \int_\Gamma G[z\cos\beta - (x\cos\alpha + y\sin\alpha)\sin\beta]d\Gamma, \\
\frac{\partial J}{\partial c} &= \int_\Gamma G d\Gamma.
\end{aligned} \tag{3.21}$$

Algebraic Level Set Cylinder Primitive

The ALS cylinder (inside of the cylinder is void, and we only consider the type with rotation axis z fixed for simplicity) function is

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = -r + \sqrt{(x - x_0)^2 + (y - y_0)^2}, \tag{3.22}$$

the corresponding design parameters are x_0, y_0 and r.

The derivatives of φ with respect to the design parameters and the norm of $\nabla\varphi$ are:

$$\begin{aligned}
\frac{\partial\varphi}{\partial x_0} &= \frac{x_0 - x}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}, \\
\frac{\partial\varphi}{\partial y_0} &= \frac{y_0 - y}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}, \\
\frac{\partial\varphi}{\partial r} &= -1, \\
|\nabla\varphi| &= 1.
\end{aligned} \tag{3.23}$$

For any objective function J with corresponding shape gradient density G , we have the SDSA formulations with respect to all parameters

$$\begin{aligned}
\frac{\partial J}{\partial x_0} &= \int_\Gamma G \frac{x_0 - x}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} d\Gamma = \frac{1}{r}\int_\Gamma G(x_0 - x) d\Gamma, \\
\frac{\partial J}{\partial y_0} &= \int_\Gamma G \frac{(y_0 - y)}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} d\Gamma = \frac{1}{r}\int_\Gamma G(y_0 - y) d\Gamma, \\
\frac{\partial J}{\partial r} &= -\int_\Gamma G d\Gamma.
\end{aligned} \tag{3.24}$$

Algebraic Level Set Sphere Primitive

The ALS sphere (inside is void) function is

$$\varphi(\boldsymbol{x}, \boldsymbol{s}) = -r + \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}, \tag{3.25}$$

the corresponding design parameters are x_0, y_0, z_0 and r.

The derivatives of φ with respect to the design parameters and the norm of $\nabla\varphi$ are:

$$\begin{aligned}
\frac{\partial \varphi}{\partial x_0} &= \frac{x_0 - x}{\sqrt{(x-x_0)^2 + (y-y_0)^2 + (y-y_0)^2}}, \\
\frac{\partial \varphi}{\partial y_0} &= \frac{y_0 - y}{\sqrt{(x-x_0)^2 + (y-y_0)^2 + (y-y_0)^2}}, \\
\frac{\partial \varphi}{\partial z_0} &= \frac{z_0 - z}{\sqrt{(x-x_0)^2 + (y-y_0)^2 + (y-y_0)^2}}, \\
\frac{\partial \varphi}{\partial r} &= -1, \\
|\nabla\varphi| &= 1.
\end{aligned} \tag{3.26}$$

For any objective function J with corresponding shape gradient density G , we have the SDSA formulations with respect to all parameters

$$\begin{aligned}
\frac{\partial J}{\partial x_0} &= \int_\Gamma G \frac{(x_0 - x)}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} d\Gamma = \frac{1}{r} \int_\Gamma G(x_0 - x) d\Gamma, \\
\frac{\partial J}{\partial y_0} &= \int_\Gamma G \frac{(y_0 - y)}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} d\Gamma = \frac{1}{r} \int_\Gamma G(y_0 - y) d\Gamma, \\
\frac{\partial J}{\partial z_0} &= \int_\Gamma G \frac{(z_0 - z)}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} d\Gamma = \frac{1}{r} \int_\Gamma G(z_0 - z) d\Gamma, \\
\frac{\partial J}{\partial r} &= -\int_\Gamma G d\Gamma. \qquad (3.27)
\end{aligned}$$

Till now, all the SDSA formulations for the basic ALS primitives are listed, and the preparation work for shape and topology optimization with ALS framework is accomplished.

3.3 Shape Design Sensitivity Analysis for Parametric Geometric Representation

Of the implicit and parametric geometric representations, it is difficult to declare that one is always more appropriate than the other. Modern CAD systems use B-Rep data structure and parametric representation to represent models instead of utilizing implicit representation. The reasons for this are that implicit primitives are unable to represent bounded curves and surfaces (like line segment) and they do not easily transform between coordinate systems. Also, some free-form curves and surfaces such as B-spline or NURBs type are easily to be represented in parametric forms.

We choose the basic parametric primitives instead of free-form primitives like NURBs surface to represent a solid, including 2D parametric primitives (straight line, line segment, circle), and 3D parametric primitives (plane patch, cylinder and sphere). For 2D primitives, they can

be expressed in the general parametric form of $\boldsymbol{x} = \boldsymbol{x}_e(u, \boldsymbol{g})$, where u is the parametric coordinate, $\boldsymbol{g} = \{g^j, j = 1, \ldots, M\}$ is the geometric vector. For 3D primitives, their general parametric expression is $\boldsymbol{x} = \boldsymbol{x}_s(u, v, \boldsymbol{g})$, where u, v are the parametric coordinates. One will note that a primitive could be parameterized in many different ways, so the method chosen here is not the only one. With the selected set of basic parametric primitives and the topology changing capability of the geometry kernels, we expect the shape and topology optimization within genuine CAD environments could be conducted.

3.3.1 Sensitivity Analysis Formulations

If we consider $\boldsymbol{g}$ as a function of t, for a given parametric primitive, its design velocity can be expressed in a direct way as:

$$\boldsymbol{V} = \frac{d\boldsymbol{x}}{dt} = \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{g}}\right) \cdot \left(\frac{d\boldsymbol{g}}{dt}\right) = \sum_{j=1}^{M} \frac{\partial \boldsymbol{x}}{\partial g_j} \cdot \dot{g}_j. \tag{3.28}$$

While its normal component V_n can be expressed as:

$$V_n = \boldsymbol{V} \cdot \boldsymbol{n} = \sum_{j=1}^{M} \left(\frac{\partial \boldsymbol{x}}{\partial g_j} \cdot \boldsymbol{n}\right) \cdot \dot{g}_j. \tag{3.29}$$

The next step we consider an arbitrary point $\boldsymbol{x}$ on the boundary of a solid that is composed of N parts of boundary parametric primitives. Here we also have the similar assumption that all parametric primitives are independent, then any point on this boundary could be considered as $\boldsymbol{x} = \boldsymbol{x}(\mathbf{g})$, where $\mathbf{g} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_N\} = \{g_1, \ldots, g_{N_p}\}$ is the parameter vector containing all independent parameters collected from related boundary primitives. Consider $\mathbf{g}$ as the function of t, the time design

velocity can be written as:

$$\boldsymbol{V} = \frac{d\boldsymbol{x}}{dt} = \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{g}}\right) \cdot \left(\frac{d\mathbf{g}}{dt}\right) = \sum_{i=1}^{N_p} \frac{\partial \boldsymbol{x}}{\partial g_i} \cdot \dot{g}_i, \tag{3.30}$$

$$V_n = \boldsymbol{V} \cdot \boldsymbol{n} = \sum_{i=1}^{N_p} \left(\frac{\partial \boldsymbol{x}}{\partial g_i} \cdot \boldsymbol{n}\right) \cdot \dot{g}_i. \tag{3.31}$$

In fact, Eq. (3.30) and Eq. (3.31) tell us that time design velocity $\boldsymbol{V}$ and its normal component V_n are determined by the number of parameters chosen to parameterize a primitive.

The shape derivative in Eq. (2.33) could be obtained as:

$$\begin{aligned} \dot{J} &= \int_\Gamma G(\boldsymbol{x}, u) V_n d\Gamma \\ &= \sum_{i=1}^{N_p} \Big[\int_\Gamma G(\frac{\partial \boldsymbol{x}}{\partial g_i} \cdot \boldsymbol{n}) d\Gamma \Big] \cdot \dot{g}_i. \end{aligned} \tag{3.32}$$

If we consider the objective function J as a function of t through $\mathbf{g}$, using the chain rule, a direct calculation is

$$\dot{J} = \sum_{i=1}^{N_p} \frac{\partial J}{\partial g_i} \cdot \dot{g}_i. \tag{3.33}$$

By comparison, we have the sensitivity for each parameter g_i, that is

$$\frac{\partial J}{\partial g_i} = \int_\Gamma G(\frac{\partial \boldsymbol{x}}{\partial g_i} \cdot \boldsymbol{n}) d\Gamma. \tag{3.34}$$

If we denote $\boldsymbol{V}_{g_i} = \frac{d\boldsymbol{x}}{dg_i}$ as the design velocity with respect to a given parameter g_i, then $V_{g_i}^n = \boldsymbol{V}_{g_i} \cdot \boldsymbol{n}$

$$\frac{dJ}{dg_i} = \int_\Gamma G V_{g_i}^n d\Gamma = \sum_{k=1}^{T} \int_{\Gamma_k} G V_{g_i}^{n(k)} d\Gamma, \tag{3.35}$$

where Γ_k is the boundary segment that is related to parameter g_i and T is the total number of such boundary segments.

3.3.2 Sensitivity for Basic 2D Parametric Primitives

Parametric Line Primitive

For the parametric line primitive, its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_e(u, \boldsymbol{g}) = \boldsymbol{x}_0 + u\boldsymbol{e}_1. \tag{3.36}$$

If we use the global coordinate system setup as shown in Figure (3.4),

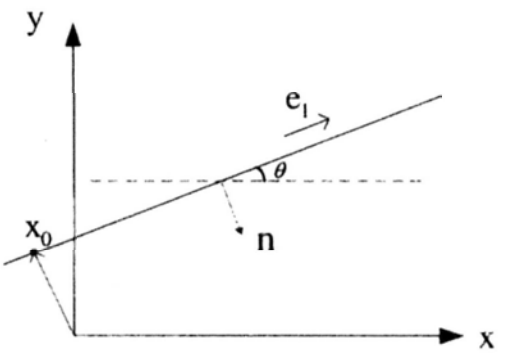


Figure 3.4: Global coordinate setup for the line primitive

the representation can be rewritten as:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c\sin\theta \\ -c\cos\theta \end{bmatrix} + u \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \tag{3.37}$$

where $\boldsymbol{g} = \{c, \theta\}$, $\boldsymbol{n} = [\sin\theta, -\cos\theta]^T$, $\boldsymbol{x}_0 = c\boldsymbol{n} = [c\sin\theta, -c\cos\theta]^T$. Consequently, we have

$$\begin{aligned} \frac{d\boldsymbol{x}}{dc} &= \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}, \\ \frac{d\boldsymbol{x}}{d\theta} &= \begin{bmatrix} c\cos\theta - u\sin\theta \\ c\sin\theta + u\cos\theta \end{bmatrix}, \end{aligned} \tag{3.38}$$

and

$$\begin{aligned} V_n^c &= \frac{d\boldsymbol{x}}{dc} \cdot \boldsymbol{n} = 1, \\ V_n^\theta &= \frac{d\boldsymbol{x}}{d\theta} \cdot \boldsymbol{n} = -u. \end{aligned} \tag{3.39}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned}\frac{\partial J}{\partial c} &= \int_\Gamma G V_n^c d\Gamma = \int_{\Gamma_u} G|\boldsymbol{r}_u| du, \\ \frac{\partial J}{\partial \theta} &= \int_\Gamma G V_n^\theta d\Gamma = -\int_{\Gamma_u} Gu|\boldsymbol{r}_u| du,\end{aligned} \tag{3.40}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}]^T$.

Parametric Line Segment Primitive

For the bounded parametric line segment primitive as shown in Figure (3.5), its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_e(u, \boldsymbol{g}) = \boldsymbol{x}_e(u, \boldsymbol{x}_1, \boldsymbol{x}_2) = (1-u)\boldsymbol{x}_1 + u\boldsymbol{x}_2, \tag{3.41}$$

where $\boldsymbol{g} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, $\boldsymbol{n} = [\frac{\bar{y}}{d}, -\frac{\bar{x}}{d}]^T$, $d = |\boldsymbol{x}_2 - \boldsymbol{x}_1|$, $\bar{x} = x_2 - x_1, \bar{y} = y_2 - y_1$.

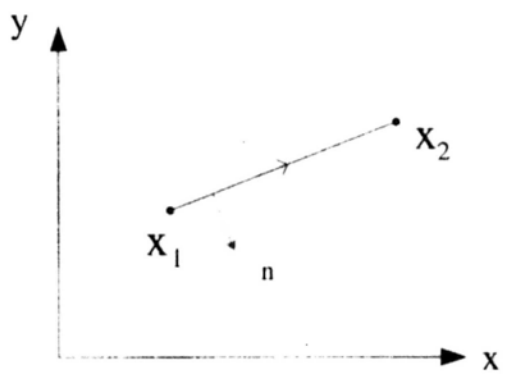


Figure 3.5: Configuration for the parametric line segment primitive

Consequently, we have

$$\begin{aligned}\frac{d\boldsymbol{x}}{d\boldsymbol{x}_1} &= (1-u)\boldsymbol{I}_{2\times 2}, \\ \frac{d\boldsymbol{x}}{d\boldsymbol{x}_2} &= u\boldsymbol{I}_{2\times 2},\end{aligned} \tag{3.42}$$

and

$$\begin{aligned} V_{x_1}^n &= \frac{\bar{y}}{d}(1-u), \\ V_{y_1}^n &= -\frac{\bar{x}}{d}(1-u), \\ V_{x_2}^n &= \frac{\bar{y}}{d}u, \\ V_{y_2}^n &= -\frac{\bar{x}}{d}u. \end{aligned} \tag{3.43}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned} \frac{\partial J}{\partial x_1} &= \int_\Gamma G V_{x_1}^n d\Gamma = \frac{\bar{y}}{d}\int_{\Gamma_u} G(1-u)|\boldsymbol{r}_u|du, \\ \frac{\partial J}{\partial y_1} &= \int_\Gamma G V_{y_1}^n d\Gamma = -\frac{\bar{x}}{d}\int_{\Gamma_u} G(1-u)|\boldsymbol{r}_u|du, \\ \frac{\partial J}{\partial x_2} &= \int_\Gamma G V_{x_2}^n d\Gamma = \frac{\bar{y}}{d}\int_{\Gamma_u} Gu|\boldsymbol{r}_u|du, \\ \frac{\partial J}{\partial y_2} &= \int_\Gamma G V_{y_2}^n d\Gamma = -\frac{\bar{x}}{d}\int_{\Gamma_u} Gu|\boldsymbol{r}_u|du. \end{aligned} \tag{3.44}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}]^T$.

Parametric Circle Primitive

For the parametric circle primitive as shown in Figure (3.6), its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_c(u, \boldsymbol{g}) = R[\boldsymbol{e}_1 \cos u + \boldsymbol{e}_2 \sin u] + \boldsymbol{x}_0, \tag{3.45}$$

where $\boldsymbol{g} = \{R, \boldsymbol{x}_0\}$, $\boldsymbol{n} = \boldsymbol{e}_1 \cos u + \boldsymbol{e}_2 \sin u$, $\boldsymbol{x}_0 = [x_0, y_0]^T$, $\boldsymbol{e}_1 = [e_{11}, e_{12}]^T$, $\boldsymbol{e}_2 = [e_{21}, e_{22}]^T$. Consequently, we have

$$\begin{aligned} \frac{d\boldsymbol{x}}{dR} &= \boldsymbol{e}_1 \cos u + \boldsymbol{e}_2 \sin u, \\ \frac{d\boldsymbol{x}}{d\boldsymbol{x}_0} &= \boldsymbol{I}_{2\times 2}, \end{aligned} \tag{3.46}$$

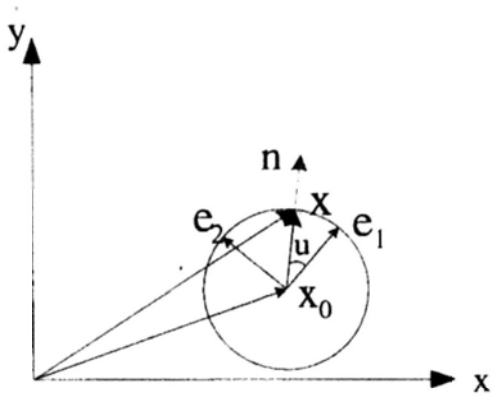


Figure 3.6: Configuration for the parametric circle primitive

and

$$\begin{aligned} V_R^n &= 1, \\ V_{x_0}^n &= e_{11}\cos u + e_{21}\sin u, \\ V_{y_0}^n &= e_{12}\cos u + e_{22}\sin u. \end{aligned} \tag{3.47}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned} \frac{\partial J}{\partial R} &= \int_\Gamma G V_R^n d\Gamma = \int_{\Gamma_u} G|\boldsymbol{r}_u| du, \\ \frac{\partial J}{\partial x_0} &= \int_\Gamma G V_{x_0}^n d\Gamma = \int_{\Gamma_u} G(e_{11}\cos u + e_{21}\sin u)|\boldsymbol{r}_u| du, \\ \frac{\partial J}{\partial y_0} &= \int_\Gamma G V_{y_0}^n d\Gamma = \int_{\Gamma_u} G(e_{12}\cos u + e_{22}\sin u)|\boldsymbol{r}_u| du, \end{aligned}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}]^T$.

3.3.3 Sensitivity for Basic 3D Parametric Primitives

Parametric Plane Primitive

For the parametric plane primitive, its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_s(u, v, g) = \boldsymbol{x}_0 + u\boldsymbol{e}_1 + v\boldsymbol{e}_2, \tag{3.48}$$

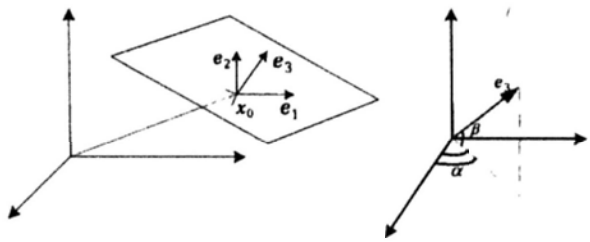


Figure 3.7: Configuration for the parametric plane primitive

If we use the parameterization as shown in Figure (3.7), then the representation can be rewritten as:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c\cos\beta\cos\alpha \\ c\cos\beta\sin\alpha \\ c\sin\beta \end{bmatrix} + u \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix} + v \begin{bmatrix} -\cos\alpha\sin\beta \\ -\sin\alpha\sin\beta \\ \cos\beta \end{bmatrix}, \quad (3.49)$$

where $\boldsymbol{g} = \{c, \alpha, \beta\}$, $\boldsymbol{n} = \boldsymbol{e}_3 = [\cos\beta\cos\alpha, \cos\beta\sin\alpha, \sin\beta]^T$, $\boldsymbol{x}_0 = c\boldsymbol{n} = [c\cos\beta\cos\alpha, c\cos\beta\sin\alpha, c\sin\beta]^T$, $\boldsymbol{e}_1 = [-\sin\alpha, \cos\alpha, 0]^T$, $\boldsymbol{e}_2 = [-\cos\alpha\sin\beta, -\sin\alpha\sin\beta, \cos\beta]^T$. Consequently, we have

$$\begin{aligned}
\frac{d\boldsymbol{x}}{dc} &= \begin{bmatrix} \cos\beta\cos\alpha \\ \cos\beta\sin\alpha \\ \sin\beta \end{bmatrix}, \\
\frac{d\boldsymbol{x}}{d\theta} &= \begin{bmatrix} -c\cos\beta\sin\alpha - u\cos\alpha + v\sin\alpha\sin\beta \\ c\cos\beta\cos\alpha - u\sin\alpha - v\cos\alpha\sin\beta \\ 0 \end{bmatrix}, \\
\frac{d\boldsymbol{x}}{d\beta} &= \begin{bmatrix} -c\cos\alpha\sin\beta - v\cos\alpha\cos\beta \\ -c\sin\alpha\sin\beta - v\sin\alpha\cos\beta \\ c\cos\beta - v\sin\beta \end{bmatrix}, \quad (3.50)
\end{aligned}$$

and

$$\begin{aligned} V_n^c &= \frac{d\boldsymbol{x}}{dc} \cdot \boldsymbol{n} = 1, \\ V_n^\alpha &= \frac{d\boldsymbol{x}}{d\alpha} \cdot \boldsymbol{n} = -u\cos\beta, \\ V_n^\beta &= \frac{d\boldsymbol{x}}{d\beta} \cdot \boldsymbol{n} = -v. \end{aligned} \tag{3.51}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned} \frac{\partial J}{\partial c} &= \int_\Gamma G V_n^c d\Gamma = \int_{\Gamma_{(u,v)}} G|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \\ \frac{\partial J}{\partial \alpha} &= \int_\Gamma G V_n^\alpha d\Gamma = -\int_{\Gamma_{(u,v)}} Gu\cos\beta|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \\ \frac{\partial J}{\partial \beta} &= \int_\Gamma G V_n^\beta d\Gamma = -\int_{\Gamma_{(u,v)}} Gv|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \end{aligned} \tag{3.52}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}]^T$, $\boldsymbol{r}_v = [\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}]^T$.

Parametric Cylinder Primitive

For the parametric cylinder primitive, its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_s(u, v, g) = \boldsymbol{x}_0 + R[\boldsymbol{e}_1 \cos u + \boldsymbol{e}_2 \sin u] + v\boldsymbol{e}_3. \tag{3.53}$$

If we use the parameterization as shown in Figure (3.8), then the representation can be rewritten as:

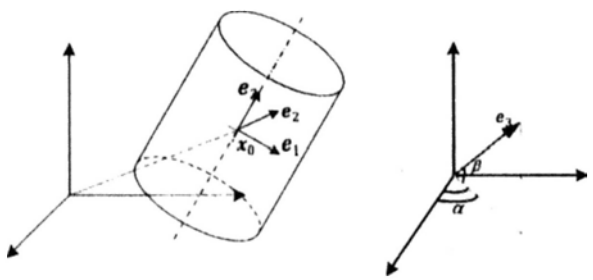


Figure 3.8: Configuration for the parametric cylinder primitive

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -c_1 \sin\alpha \\ c_1 \cos\alpha \\ 0 \end{bmatrix} + \begin{bmatrix} -c_2 \cos\alpha \sin\beta \\ -c_2 \sin\alpha \sin\beta \\ c_2 \cos\beta \end{bmatrix}$$
$$+ \quad R\left(\begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix} \cos u + \begin{bmatrix} -\cos\alpha \sin\beta \\ -\sin\alpha \sin\beta \\ \cos\beta \end{bmatrix} \sin u \right)$$
$$+ \quad v \begin{bmatrix} \cos\beta \cos\alpha \\ \cos\beta \sin\alpha \\ \sin\beta \end{bmatrix}, \qquad (3.54)$$

where $\boldsymbol{g} = \{c_1, c_2, \alpha, \beta\}$, $\boldsymbol{n} = \boldsymbol{e}_1 \cos u + \boldsymbol{e}_2 \sin u$, $\boldsymbol{x}_0 = c_1\boldsymbol{e}_1 + c_2\boldsymbol{e}_2 = [-c_1 \sin\alpha - c_2 \cos\alpha \sin\beta, \quad c_1 \cos\alpha - c_2 \sin\alpha \sin\beta, \quad c_2 \cos\beta]^T$, $\boldsymbol{e}_1 = [-\sin\alpha, \cos\alpha, 0]^T$, $\boldsymbol{e}_2 = [-\cos\alpha \sin\beta, -\sin\alpha \sin\beta, \cos\beta]^T$. Consequently, we have

$$\frac{d\boldsymbol{x}}{dR} = \boldsymbol{n},$$

$$\frac{d\boldsymbol{x}}{dc_1} = \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix},$$

$$\frac{d\boldsymbol{x}}{dc_2} = \begin{bmatrix} -\cos\alpha \sin\beta \\ -\sin\alpha \sin\beta \\ \cos\beta \end{bmatrix},$$

$$\frac{d\boldsymbol{x}}{d\alpha} = \begin{bmatrix} -R\cos u \cos\alpha + R \sin u \sin\alpha \sin\beta - c_1 \cos\alpha + c_2 \sin\alpha \sin\beta \\ -R\cos u \sin\alpha - R \sin u \cos\alpha \sin\beta - c_1 \sin\alpha - c_2 \cos\alpha \sin\beta \\ 0 \end{bmatrix},$$

$$\frac{d\boldsymbol{x}}{d\beta} = \begin{bmatrix} -R\sin u \cos\alpha \cos\beta - v\cos\alpha \sin\beta - c_2 \cos\alpha \cos\beta \\ -R\sin u \sin\alpha \cos\beta - v\sin\alpha \sin\beta - c_2 \sin\alpha \cos\beta \\ -R\sin u \sin\beta + v\cos\beta - c_2 \sin\beta \end{bmatrix}, \qquad (3.55)$$

and

$$\begin{aligned} V_n^R &= \frac{d\boldsymbol{x}}{dR}\cdot\boldsymbol{n} = 1, \\ V_n^{c_1} &= \frac{d\boldsymbol{x}}{dc_1}\cdot\boldsymbol{n} = \cos u, \\ V_n^{c_2} &= \frac{d\boldsymbol{x}}{dc_2}\cdot\boldsymbol{n} = \sin u, \\ V_n^{\alpha} &= \frac{d\boldsymbol{x}}{d\alpha}\cdot\boldsymbol{n} = \sin\beta(c_1\sin u - c_2\cos u) - v\cos u\cos\beta, \\ V_n^{\beta} &= \frac{d\boldsymbol{x}}{d\beta}\cdot\boldsymbol{n} = v\sin u. \end{aligned} \tag{3.56}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned} \frac{\partial J}{\partial R} &= \int_\Gamma GV_n^R d\Gamma = \int_{\Gamma_{(u,v)}} G|\boldsymbol{r}_u\times\boldsymbol{r}_v|dudv, \\ \frac{\partial J}{\partial c_1} &= \int_\Gamma GV_n^{c_1} d\Gamma = \int_{\Gamma_{(u,v)}} G|\boldsymbol{r}_u\times\boldsymbol{r}_v|\cos u dudv, \\ \frac{\partial J}{\partial c_2} &= \int_\Gamma GV_n^{c_2} d\Gamma = \int_{\Gamma_{(u,v)}} G|\boldsymbol{r}_u\times\boldsymbol{r}_v|\sin u dudv, \\ \frac{\partial J}{\partial \alpha} &= \int_\Gamma GV_n^{\alpha} d\Gamma = \int_{\Gamma_{(u,v)}} Gu[\sin\beta(c_1\sin u - c_2\cos u) - v\cos u\cos\beta]|\boldsymbol{r}_u\times\boldsymbol{r}_v|dudv, \\ \frac{\partial J}{\partial \beta} &= \int_\Gamma GV_n^{\beta} d\Gamma = -\int_{\Gamma_{(u,v)}} Gv\sin u|\boldsymbol{r}_u\times\boldsymbol{r}_v|dudv, \end{aligned} \tag{3.57}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}]^T$, $\boldsymbol{r}_v = [\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}]^T$.

Parametric Sphere Primitive

For the parametric sphere primitive as shown in Figure (3.9), its general representation is

$$\boldsymbol{x} = \boldsymbol{x}_s(u,v,g) = \boldsymbol{x}_0 + R[\boldsymbol{e}_1\cos u + \boldsymbol{e}_2\sin u] + R\boldsymbol{e}_3\sin v, \tag{3.58}$$

where $\boldsymbol{g} = \{R, \boldsymbol{x}_0\}$, $\boldsymbol{n} = \boldsymbol{e}_1\cos u + \boldsymbol{e}_2\sin u + \boldsymbol{e}_3\sin v$, $\boldsymbol{e}_1 = [e_{11}, e_{12}, e_{13}]^T$, $\boldsymbol{e}_2 = [e_{21}, e_{22}, e_{23}]^T$, $\boldsymbol{e}_3 = [e_{31}, e_{32}, e_{33}]^T$, $\boldsymbol{x}_0 = [x_0, y_0, z_0]^T$.

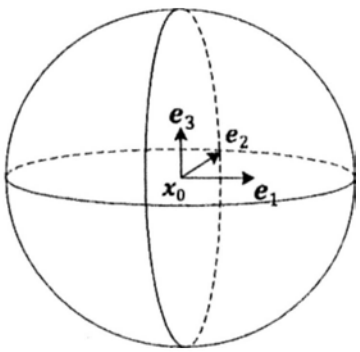


Figure 3.9: Configuration for sphere primitive

Consequently, we have

$$\begin{aligned} \frac{d\boldsymbol{x}}{dR} &= \boldsymbol{n}, \\ \frac{d\boldsymbol{x}}{d\boldsymbol{x}_0} &= \boldsymbol{I}_{3\times 3}, \end{aligned} \tag{3.59}$$

and

$$\begin{aligned} V_R^n &= 1, \\ V_{x_0}^n &= e_{11}\cos u + e_{21}\sin u + e_{31}\sin v, \\ V_{y_0}^n &= e_{12}\cos u + e_{22}\sin u + e_{32}\sin v, \\ V_{z_0}^n &= e_{13}\cos u + e_{23}\sin u + e_{33}\sin v. \end{aligned} \tag{3.60}$$

Therefore, for any cost function J, we can obtain

$$\begin{aligned} \frac{\partial J}{\partial R} &= \int_{\Gamma_{(u,v)}} G|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \\ \frac{\partial J}{\partial x_0} &= \int_{\Gamma_{(u,v)}} G(e_{11}\cos u + e_{21}\sin u + e_{31}\sin v)|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \\ \frac{\partial J}{\partial y_0} &= \int_{\Gamma_{(u,v)}} G(e_{12}\cos u + e_{22}\sin u + e_{32}\sin v)|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \\ \frac{\partial J}{\partial z_0} &= \int_{\Gamma_{(u,v)}} G(e_{13}\cos u + e_{23}\sin u + e_{33}\sin v)|\boldsymbol{r}_u \times \boldsymbol{r}_v| du dv, \end{aligned} \tag{3.61}$$

where $\boldsymbol{r}_u = [\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}]^T$, $\boldsymbol{r}_v = [\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}]^T$.

3.4 Consistency of Shape Design Sensitivity Analysis under Both Frameworks

The implicit function Φ under the ALS framework can be expressed as $\Phi = \Phi(\boldsymbol{x}, \boldsymbol{S})$, where $\boldsymbol{S} = s_I, I = 1 \cdots, N_s$. If we consider $\boldsymbol{x}$ as a function of $\boldsymbol{S}$, for $\forall s \in \boldsymbol{S}$, using the chain rule, the following formulation should be always valid on its boundary Γ,

$$\frac{\partial \Phi}{\partial s} + \nabla \Phi \cdot \frac{d\boldsymbol{x}}{ds} = 0, \tag{3.62}$$

which implies

$$\frac{\partial \Phi}{\partial s} = -\nabla \Phi \cdot \frac{d\boldsymbol{x}}{ds}. \tag{3.63}$$

With $\boldsymbol{n} = -\frac{\nabla \Phi}{|\nabla \Phi|}$, we have

$$\frac{d\boldsymbol{x}}{ds} \cdot \boldsymbol{n} = \frac{1}{|\nabla \Phi|} \cdot \frac{\partial \Phi}{\partial s}. \tag{3.64}$$

Here we denote $\frac{d\boldsymbol{x}}{ds}$ as the design velocity with respect to the parameter s of a given ALS primitive, then $V_n^s = \frac{d\boldsymbol{x}}{ds} \cdot \boldsymbol{n} = \frac{1}{|\nabla \Phi|} \cdot \frac{\partial \Phi}{\partial s}$. This notation is the same as the definition we used in parametric representation part.

Therefore, for any given basic geometric primitive, if the parameters of its implicit ALS representation are the same as the set of parameters in its corresponding parametric representation, the result of the SDSA formulations regarding these parameters under both frameworks are identical. The difference between the two frameworks lies in the global geometry updating scheme, which will be discussed in Chapter 4.

□ **End of chapter.**

Chapter 4

Optimization Schemes

The well-established discrete level set based structural shape and topology optimization framework involves the process of solving the PDE. Therefore only the steepest descent method is found to be applicable to evolve the structural interface so far. With our proposed parameterizations, one can convert the infinite-dimensional shape and topology optimization problem into a parametric optimization problem and hence many mature optimization algorithms can be used. Readers are referred to [39,40,42,43] for more details. The optimization algorithms will be developed in this chapter and the difference between the geometry updating schemes for the ALS model and the parametrically represented B-Rep model will be addressed.

4.1 Augmented Lagrangian Method for Applying Constraints

There are many different kinds of practical engineering constraints in real engineering problems, such as local stress constraint or maximum

displacement constraint. For mean compliance problem, a volume constraint is always applied. Although in practice, minimization of volume as the objective function subject to compliance constraint is more favorable, we still choose volume as the constraint with mean compliance as the cost function because the volume constraint is easy to impose. As has been proved in [76], the abovementioned two problems are dual problems, which means if one can be solved, vise versa. The volume constraint can be stated as:

$$\int_{\Omega} d\Omega \leq vol, \tag{4.1}$$

which describes an upper limit on the amount of material in terms of the maximum admissible volume *vol* of the structure. Since an equality constraint is generally not easy to be imposed during the optimization, we can combine it with the objective function using the augmented Lagrangian method [77] by imposing the volume constraint as a penalty term in the objective function to construct an augmented objective function:

$$L = J + \lambda\Big(\int_{\Omega} d\Omega - vol\Big) + \frac{r}{2}\Big(\int_{\Omega} d\Omega - vol\Big)^2, \tag{4.2}$$

where $\lambda > 0$ is the Lagrange multiplier and $r > 1$ is the penalty parameter.

Similar to what we have done in section 2.1.3, the shape derivative of L is

$$\dot{L} = \int_{\Gamma}\Big(\boldsymbol{f}\cdot\boldsymbol{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{u})^T\boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{u}) + \bar{\lambda}\Big)V_n\, d\Gamma, \tag{4.3}$$

where

$$\bar{\lambda} = \max\Big\{0,\ \lambda + r\Big(\int_{\Omega} d\Omega - vol\Big)\Big\}. \tag{4.4}$$

and V_n can be formulated as shown in Eq. (3.7) for implicit representation or in Eq. (3.29) for parametric representation.

It is proved in [78] that along with the design variables converging to a local optimal solution, the Lagrange multiplier $\bar{\lambda}$ converges to the correct value λ^*. In theory, the penalty parameter r should be large enough to speed up the convergence of λ. However, in structural optimization problems, if r is too large, the volume changes too drastically and some useful intermediate shapes will be missed. Consequently, r should be selected properly.

We can rewrite Eq. (4.3) in a more compact form as:

$$\dot{L} = \int_{\Gamma} \bar{G} V_n d\Gamma, \tag{4.5}$$

where $\bar{G}$ can be denoted as:

$$\bar{G} = \boldsymbol{f} \cdot \boldsymbol{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{u})^T \boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{u}) + \bar{\lambda}. \tag{4.6}$$

From now on, in the rest part of this dissertation, the symbol G will be used to replace $\bar{G}$ for simplicity, unless otherwise specified.

One can picture that every time when the structural geometry is updated, the new displacement field $\boldsymbol{u}$ can be recalculated by solving the equilibrium equation with boundary conditions, and hence the equilibrium equation is satisfied automatically. Further, by utilizing augmented Lagrangian multiplier method to impose the volume constraint, the original constrained optimization problem is turned into an unconstrained optimization problem.

4.2 Parametric Steepest Descent Method for Structural Optimization

4.2.1 Formulations

In the infinite dimensional optimization, to guarantee the reduction of L, we require the boundary to move with the normal design velocity field V_n that satisfies the descent property:

$$\dot{L} < 0. \tag{4.7}$$

A simple descent method is the steepest descent (ST) method in which

$$V_n = -G. \tag{4.8}$$

Substituting Eq. (4.8) into Eq. (4.3), we can see that the descent property is satisfied.

Since we parameterize the structural shape with basic geometric primitives, all the internal parameters could be considered as the design variables. Again, we have to find a descent direction with the SDSA information derived in Chapter 3 to make sure that the objective function decreases. Therefore, with the implicit ALS model, if we choose the ST method to conduct optimization, we can just set $\dot{s}_I$ in Eq. (3.9) as:

$$\dot{s}_I = -\frac{\partial L}{\partial s_I} = -\sum_{k=1}^{T} \int_{\Gamma_k} G \frac{1}{|\nabla \varphi_j|} \frac{\partial \varphi_j}{\partial s_I} d\Gamma. \tag{4.9}$$

This guarantees the time derivatives for the global N_s parameters satisfy the descent property, namely,

$$\dot{L} = -\sum_{I=1}^{N_s} \left(\frac{\partial L}{\partial s_I} \right)^2 < 0. \tag{4.10}$$

We can obtain the similar result for parametric representation with B-Rep model by setting g_i in Eq. (3.33) as:

$$\dot{g}_i = -\frac{\partial L}{\partial g_i} = -\sum_{k=1}^{T} \int_{\Gamma_k} G V_{g_i}^{n(k)} d\Gamma. \tag{4.11}$$

We also can verify that the descent property is satisfied

$$\dot{L} = -\sum_{i=1}^{N_p} \left(\frac{\partial L}{\partial g_i}\right)^2 < 0. \tag{4.12}$$

For an arbitrary parameter with either implicit ALS primitives or parametric primitives, the updating schemes are:

$$s^{n+1} = s^n + \tau \dot{s}, \tag{4.13}$$

or

$$g^{n+1} = g^n + \tau \dot{g}, \tag{4.14}$$

where s^n and p^n are the parameters at previous step for implicit ALS primitives and parametric primitives respectively, whereas s^{n+1} and g^{n+1} are the updated parameters at current step for the corresponding primitives. Here τ is the time step size that one should carefully choose.

It is well known that if we utilize the line search method to determine the optimal time step size τ for each optimization step, the convergence speed will be much faster [79], however, this method requires multiple function evaluations, which could be very expensive in time since for each iteration the FE analysis process is involved. Therefore, we choose to adopt the fixed time step size in stead of trying to determine the optimal time step size even though this may result in a comparatively slower convergence speed.

When the parameters are updated, we can update the geometry accordingly. For the ALS model, we can update the new geometry of

the structure as:

$$\Phi^{n+1} = \Phi(\varphi_1^{n+1}, \ldots, \varphi_i^{n+1}, \ldots, \varphi_N^{n+1}), \tag{4.15}$$

where $\varphi_i^{n+1} = \varphi_i(\boldsymbol{x}, \boldsymbol{s}_i^{n+1})$.

For the B-Rep model with the parametric representation, the updating scheme is a little bit different. In traditional history-based solid modelers, they use a construction approach that is highly dependent on the sequence of instructions. Therefore, changes to early instructions can cause later instructions to fail, which is known as rebuilding or regeneration failure [7]. Different from the traditional history-based modelers, the emergence of direct modeling technique uses local-operations like 'tweak' of the commercial geometry kernels such as ACIS [74] to modify the geometries without the need of understanding the design intent history. These types of functions alter the boundary geometries and then automatically update the topology of the solid (i.e. adjacency relationships) to accommodate the new geometries. This is exactly the way we update the geometries with the ACIS kernel to realize topological changes.

4.2.2 Algorithm

The following algorithm is established to solve the structural optimization problem, which consists of the following steps:

Step 1: Initialize the geometry representation of a given solid at $t = 0$ corresponding to an initial design Ω in terms of its boundary Γ. For the ALS model, the initialization includes the definition of the ALS primitives φ_i and the corresponding boolean operators

acting on them; for B-Rep model with the parametric representation, the initialization includes the current B-rep data structure definition and its related parametric primitives $\boldsymbol{x}_i$. Also, determine the design variables s_I for the ALS primitives or g_i for the parametrically represented primitives, the time step size τ, the Lagrangian multiplier λ and the penalty parameter r.

Step 2: Solve the augmented Lagrangian multiplier problem L for any given cost function and constraint

(2.1) Solve the equilibrium equation with the FEM to get the displacement field $\boldsymbol{u}$, the strain field $\boldsymbol{\varepsilon}$ and the stress field $\boldsymbol{\sigma}$.

(2.2) Calculate the sensitivity of augmented functional L regarding all selected design variables, namely, $\frac{\partial L}{\partial s_I}$ for the ALS model or $\frac{\partial L}{\partial g_i}$ for the B-Rep model with the parametric representation.

(2.3) Update the parameters as $s_I^{n+1} = s_I^n - \tau \frac{\partial L}{\partial s_I}$ or $g_i^{n+1} = g_i^n - \tau \frac{\partial L}{\partial g_i}$.

(2.4) Update the geometry either according to a set of predesignated boolean operators acting on the corresponding set of implicit ALS primitives, or using local operations to update the B-rep data structure with the given set of parametric primitives.

Step 3: Update the Lagrangian multiplier $\bar{\lambda} = \max\left\{0,\ \lambda + r\left(\int_\Omega d\Omega - vol\right)\right\}$.

Step 4: Check if a termination condition is satisfied. If the condition is met, an optimal solution is found. Otherwise, repeat steps 2-3 until convergence. The termination condition is defined as

$|\frac{L^{n+1}-L^n}{L^{n+1}}| \leq \epsilon$, where ϵ is a specified error limit, here we chose it as 10^{-4}.

With this algorithm, one can perform shape and topology optimization simultaneously under the CAD-based framework.

4.3 Least Squares Curve Fitting Method for Structural Optimization

4.3.1 Formulations

The least squares (LSQ) curve fitting method is a mathematical approach mainly used to find the best-fitting curve to a given set of points by minimizing the sum of the squares of the the residuals of the points from the curve. The sum of the squares of the residuals is used instead of the absolute values of the residuals because this allows the residuals to be treated as a continuous differentiable quantity.

In the typical LSQ problem, the objective Q has the following special form:

$$Q = \sum [y_i - f(x_i, a_1, \ldots, a_j, \ldots, a_n)]^2 = \sum r_i^2, \tag{4.16}$$

where r_i is the residual. By minimizing this function, one could select values for the parameters a_j, $j = 1, \cdots, n$, that best match the model to the data.

From Eq. (4.8) we know that, in the infinite dimensional optimization, the special descent direction $V_n = -G$ can guarantee every point on the boundary Γ moves towards the steepest descent direction which can reduce the objective function the most at current step, and this

special V_n can be considered as the criterion for developing our optimization algorithms.

Inspired by the abovementioned LSQ method, here we want to take the residual along the design boundary between the normal component of time design velocity in infinite dimensional domain and normal component of time design velocity in the parameterized domain (either implicit representation or parametric representation), and fit this residual into the LSQ curve fitting model to obtain a different descent direction for either s_I or g_i at each iteration. We denote the normal component of time design velocity in infinite dimensional domain as $V_{n_{Inf}}$ and denote the normal component of time design velocity under ALS or parametric framework as V_{n_P}. Then this initial idea could be illustrated in Figure (4.1)

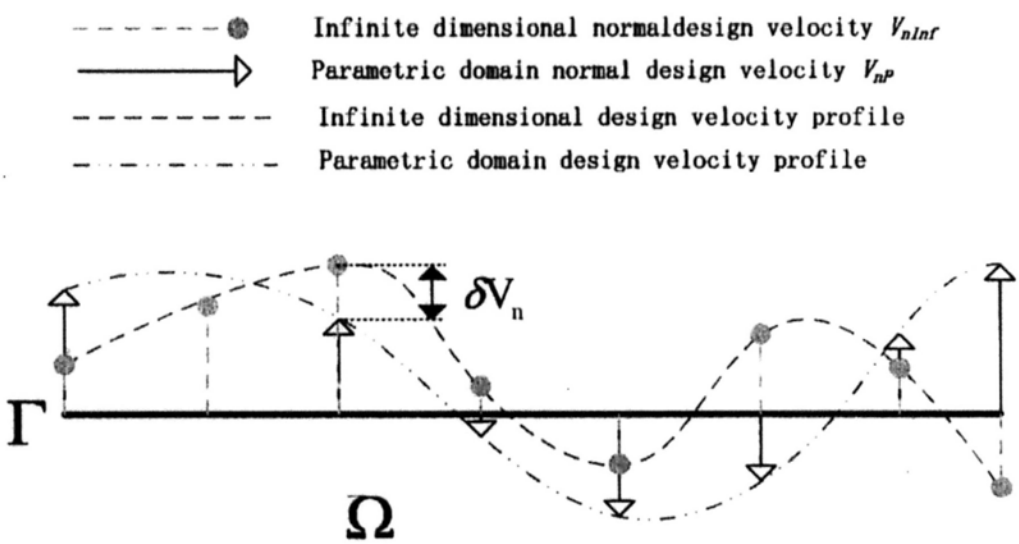


Figure 4.1: δV_n as the residual for the LSQ based optimization

For any shape represented by ALS primitives, the abovementioned

intermediate subproblem could be formulated as:

$$\begin{aligned} &Find \quad \dot{s}_I \\ &minimize \quad Q = \frac{1}{2}\int_{\Gamma}(V_{n_{Inf}} - V_{n_P})^2 d\Gamma. \end{aligned} \tag{4.17}$$

Suppose the boundary Γ is categorized into M parts $\Gamma_1, \ldots, \Gamma_i, \ldots, \Gamma_N$, each Γ_i belongs to only one specific ALS primitive φ_i, then Q in Eq. (4.18) can be written as:

$$Q = \sum_{i=1}^{N} Q_i = \frac{1}{2}\sum_{i=1}^{N}\int_{\Gamma_i}(V_{n_{Inf}} - V_{n_P})^2 d\Gamma. \tag{4.18}$$

All ALS primitives we chose are normalized, namely, $|\nabla\Phi| = 1$, therefore, along the entire boundary, V_{n_P} in Eq. (3.7) can be rewritten as:

$$V_{n_P} = \sum_{I=1}^{N_s}\left(\frac{\partial\Phi}{\partial s_I}\right)\cdot\dot{s}_I. \tag{4.19}$$

For arbitrary $\boldsymbol{x}$ on the specific boundary Γ_i defined by φ_i, by applying Lemma 3 in Chapter 3, its normal component of time design velocity can be further simplified as:

$$V_{n_P} = \sum_{j=1}^{M_i}\left(\frac{\partial\varphi_i}{\partial s_i^j}\right)\cdot\dot{s}_i^j, \tag{4.20}$$

where s_i^j, $j = 1, \ldots, M_i$, are the internal parameters that define φ_i in the same way as in Chapter 3. Therefore, Eq. (4.18) can be written as:

$$Q = \sum_{i=1}^{N} Q_i = \frac{1}{2}\sum_{i=1}^{N}\int_{\Gamma_i}(V_{n_{Inf}} - \sum_{j=1}^{M_i}\left(\frac{\partial\varphi_i}{\partial s_i^j}\right)\cdot\dot{s}_i^j)^2 d\Gamma. \tag{4.21}$$

Clearly, $\frac{dQ}{ds_i^j} = \frac{dQ_i}{ds_i^j}$, and this global LSQ problem can be turned into a set of simpler local LSQ problems, namely, $Min(Q) = \sum Min(Q_i)$.

Substitute Eq. (4.19) into Q_i, we have

$$Q_i = \frac{1}{2}\int_{\Gamma_i}(-G - \sum_{j=1}^{M_i}\dot{s}_i^j\cdot\frac{\partial\varphi_i}{\partial s_i^j})^2 d\Gamma. \tag{4.22}$$

If Q_i reaches its minimum, $\frac{dQ_i}{ds_i^k} = 0$, $k = 1, 2, \cdots, M_i$, then we have

$$\int_{\Gamma_i} (-G - \sum_{j=1}^{M_i} \dot{s}_i^j \cdot \frac{\partial \varphi_i}{\partial s_i^j}) \cdot \frac{\partial \varphi_i}{\partial s_i^k} d\Gamma = 0, \tag{4.23}$$

namely,

$$\sum_{j=1}^{M_i} \int_{\Gamma_i} \dot{s}_i^j \cdot \frac{\partial \varphi_i}{\partial s_i^j} \cdot \frac{\partial \varphi_i}{\partial s_i^k} d\Gamma = -\int_{\Gamma_i} G \cdot \frac{\partial \varphi_i}{\partial s_i^k} d\Gamma. \tag{4.24}$$

If written in matrix form, we have

$$\boldsymbol{H}_i \dot{\boldsymbol{s}}_i = \boldsymbol{b}_i, \tag{4.25}$$

where $H_i^{kj} = \int_{\Gamma_i} \frac{\partial \varphi_i}{\partial s_i^k} \cdot \frac{\partial \varphi_i}{\partial s_i^j} d\Gamma$, $b_i^k = -\int_{\Gamma_i} G \cdot \frac{\partial \varphi_i}{\partial s_i^k} d\Gamma = -\frac{\partial L}{\partial s_i^k}$. Here we have to note that $\boldsymbol{b}_i$ actually is the descent direction we formulated in the parametric ST algorithm part.

For any ALS primitive φ_i, its partial derivatives regarding all inner parameters could be stated as a vector $\frac{\partial \varphi_i}{\partial \boldsymbol{s}_i} = \{\frac{\partial \varphi_i}{\partial s_i^1}, \frac{\partial \varphi_i}{\partial s_i^2}, \cdots, \frac{\partial \varphi_i}{\partial s_i^{M_i}}\}$, where M_i is the total number of the parameters that define the corresponding ALS function.

We define the inner product of two arbitrary partial derivatives $\frac{\partial \varphi_i}{\partial s_i^j}, \frac{\partial \varphi_i}{\partial s_i^k}$ in $\frac{\partial \varphi_i}{\partial \boldsymbol{s}_i}$ as the boundary integral of these two functions over a finite boundary segment Γ_i that belongs to the specific ALS primitive, i.e.

$$\langle \frac{\partial \varphi_i}{\partial s_i^j}, \frac{\partial \varphi_i}{\partial s_i^k} \rangle = \int_{\Gamma_i} \frac{\partial \varphi_i}{\partial s_i^j} \cdot \frac{\partial \varphi_i}{\partial s_i^k} d\Gamma. \tag{4.26}$$

At each optimization step, as long as we can identify the independent parameters that govern the ALS primitive, the partial derivatives with respect to these parameters of the corresponding ALS function should be *linearly independent* on the boundary.

Hereby we give another Lemma from linear algebra:

Lemma 4 *$\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ is a vector over Euclidean space, let the symmetric matrix*

$$D = \begin{bmatrix} \langle \alpha_1, \alpha_1 \rangle & \langle \alpha_1, \alpha_2 \rangle & \cdots & \langle \alpha_1, \alpha_n \rangle \\ \langle \alpha_2, \alpha_1 \rangle & \langle \alpha_2, \alpha_2 \rangle & \cdots & \langle \alpha_2, \alpha_n \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle \alpha_n, \alpha_1 \rangle & \langle \alpha_n, \alpha_2 \rangle & \cdots & \langle \alpha_n, \alpha_n \rangle \end{bmatrix}$$

, then the sufficient and necessary condition for $\alpha_1, \alpha_2, \cdots, \alpha_n$ being linearly independent is $|D| \neq 0$.

Using the above lemma, we know that, theoretically, H_i is a full rank matrix, hence the solution to Eq. (4.25) could be written directly as:

$$\dot{\boldsymbol{s}}_i = \boldsymbol{H}_i^{-1} \boldsymbol{b}_i. \tag{4.27}$$

Further, it is not difficult to prove that $\boldsymbol{H}_i$ is also a positive definite matrix provided that $\alpha_i, \ldots, \alpha_n$ are linearly independent. Since $\boldsymbol{b}_i$ is a descent direction for the augmented functional L, we can prove that $\dot{\boldsymbol{s}}_i = \boldsymbol{H}_i^{-1} \boldsymbol{b}_i$ is also a descent direction for L. For other primitives and the related parameters we selected, the new time derivatives of the parameters could be acquired in the same manner.

Consequently, we can guarantee again that for the global N_s parameters, their time derivatives satisfy the descent property, namely,

$$\dot{L} = \sum_{I=1}^{N_S} (\frac{\partial L}{\partial s_I}) \cdot \dot{s}_I < 0, \tag{4.28}$$

where $\dot{s}_I$ can be calculated from s_I related matrix $\boldsymbol{H}$ and the corresponding descent vector $\boldsymbol{b}$.

Take the ALS line primitive with expression $\varphi(\boldsymbol{x}) = x \sin\alpha + y \cos\alpha + b$, for example. Suppose its corresponding boundary is Γ,

the partial derivatives are $\frac{\partial \phi}{\partial \alpha} = x\cos\alpha - y\sin\alpha$ and $\frac{\partial \phi}{\partial b} = 1$ at any position on Γ. Since these two partial derivatives are independent at all times along the boundary, its corresponding $\boldsymbol{H}_{2\times 2}$ matrix is positive definite.

The same rule can be applied to the B-Rep geometric model with parametric representations to obtain the LSQ based algorithm. For any parametric primitive, each element in $\boldsymbol{H}$ matrix can be expressed as $H_{ki} = \int_\Gamma (\frac{\partial \boldsymbol{x}}{\partial g_k} \cdot \boldsymbol{n})(\frac{\partial \boldsymbol{x}}{\partial g_i} \cdot \boldsymbol{n}) d\Gamma$, and each element in $\boldsymbol{b}$ vector can be expressed as $b_k = -\int_\Gamma G(\frac{\partial \boldsymbol{x}}{\partial g_k} \cdot \boldsymbol{n}) d\Gamma$.

The parameter updating scheme is the same as what we have done in the parametric ST method, i.e. for the implicit ALS primitives

$$s^{n+1} = s^n + \tau \dot{s}, \tag{4.29}$$

or for the parametric primitives

$$g^{n+1} = g^n + \tau \dot{g}. \tag{4.30}$$

The geometry updating schemes adopted here are the same as the schemes we used in the parametric ST method part in the previous sections.

4.3.2 Algorithm

Similar to the parametric ST algorithm in the previous section, here we give the generic algorithm for the LSQ curve fitting based optimization algorithm as:

Step 1: Initialize the geometry representation of a given solid at $t = 0$ corresponding to an initial design Ω in terms of its boundary Γ. For the ALS model, the initialization includes the definition of

the ALS primitives φ_i and the corresponding boolean operators acting on them; for B-Rep model with the parametric representation, the initialization includes the current B-rep data structure definition and its related parametric primitives $\boldsymbol{x}_i$. Also, determine the design variables s_I for the ALS primitives or g_i for the parametrically represented primitives, the time step size τ, the Lagrangian multiplier λ and the penalty parameter r.

Step 2: Solve the augmented Lagrangian multiplier problem L for any given cost function and constraint

(2.1) Solve the equilibrium equation with the FEM to get the displacement field $\boldsymbol{u}$, the strain field $\boldsymbol{\varepsilon}$ and the stress field $\boldsymbol{\sigma}$.

(2.2) LSQ subproblem to determine $\dot{\boldsymbol{S}}$

(2.2.1) Calculate the sensitivity of augmented functional L regarding all selected design variables, namely, $\frac{\partial L}{\partial s_I}$ for the ALS model or $\frac{\partial L}{\partial g_i}$ for the B-Rep model with the parametric representation.

(2.2.2) Update parameters as $\boldsymbol{s}_i^{n+1} = \boldsymbol{s}_i^n - \tau \boldsymbol{H}_i^{-1} \boldsymbol{b}_i$ or $\boldsymbol{g}_j^{n+1} = \boldsymbol{g}_j^n - \tau \boldsymbol{H}_j^{-1} \boldsymbol{b}_j$, where $\boldsymbol{s}_i$ is the parameter vector of the ALS primitive φ_i and $\boldsymbol{g}_j$ is the parameter vector of the parametric primitive $\boldsymbol{x}_j$.

(2.3) Update the geometry either according to a set of predesignated boolean operators acting on the corresponding set of implicit ALS primitives, or using local operations to update the B-rep data structure with the given set of parametric primitives.

Step 3: Update the Lagrangian multiplier $\bar{\lambda} = \max\Big\{0,\ \lambda + r\Big(\int_\Omega d\Omega - vol\Big)\Big\}$.

Step 4: Check if a termination condition is satisfied. If the condition is met, an optimal solution is found. Otherwise, repeat steps 2-3 until convergence. The termination condition is defined as $|\frac{L^{n+1}-L^n}{L^{n+1}}| \leq \epsilon$, where ϵ is a specified error limit, here we chose it as 10^{-4}.

As to be seen in Chapter 5, with the fixed time step size, the LSQ based optimization algorithm is much more efficient compared with the parametric ST based optimization algorithm.

□ **End of chapter.**

Chapter 5

Numerical Examples

In this chapter, the sensitivity analysis formulations for two different representations as well as the optimization algorithms are implemented to solve some compliance optimization problems in both two dimensions (the ALS model with implicit primitives) and in three dimensions (the B-Rep model with parametric primitives). In practice, we found that the convergence criterion is too strict for most cases. Therefore, a maximum number of steps is specified. If this number is reached, the optimization stops. In this chapter, this number is specified case by case. For all the numerical examples tested in this study, a number around a few hundreds is enough and no obvious changes in the designs and the objective functions are found even if more steps are used.

5.1 Finite Element Approximation

A challenge to structural optimization is that the conventional FEM is normally used for the stress analysis, and the finite element mesh will distort after the shape and topology change [7]. Under these cir-

cumstances, the domain must be re-meshed. However, re-meshing is a complicated and time consuming task, and will bring down the efficiency of optimization. To perform the finite element analysis, here we choose a fixed structured mesh, as it is often seen in homogenizaiton-based topology optimization [16, 19, 20]. Also, the so-called "ersatz material" approach [33] which has been widely used in stress analysis of the compliance optimization problem will be adopted in some of the following numerical examples. In this approach, the state equations (2.3) are extended from the structure domain Ω to the whole design domain D. The void domain $D\backslash\Omega$ is assumed to be replaced by a type of "weak" material, whose Young's modulus is very low. For example, Young's modulus of the weak material is often defined as:

$$E_0 = c \cdot E, \tag{5.1}$$

where E is the Young's modulus of the solid material of the structure and c is a small coefficient. In this study, c is selected as 0.001. Note that c can not be too small, otherwise the stiffness matrix will be singular.

For elements cut by structural boundary, Young's modulus is calculated according to the fraction of solid material. For example, if the volume of solid takes one half of the volume of an element, Young's modulus of this element is set to be $0.5E$.

With the assumption of ersatz material, state equations are extended to the whole design domain:

$$\begin{aligned} -\mathrm{div}\,\boldsymbol{\sigma}(\boldsymbol{u}) &= \boldsymbol{f} \quad \text{in} \quad D, \\ \boldsymbol{u} &= 0 \quad \text{on} \quad \Gamma_D, \\ \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \boldsymbol{g} \quad \text{on} \quad \Gamma_N. \end{aligned} \tag{5.2}$$

Because the design domain is fixed, no remeshing is required during the structure evolution process.

This method is simple and can give satisfactory results for compliance optimization. Also, the extended finite element method (XFEM) with abovementioned "weak" material is also employed in later examples to calculate more accurate stresses. Regarding the implementation of XFEM in topology optimization, the reader is referred to [59,82] and references herein.

Unless stated otherwise, in all the following examples, the units are consistent and the following parameters are assumed as: the Young's modulus $E = 1$ for the solid material, $E = 0.001$ for the weak material, and Poisson's ratio $\nu = 0.3$.

5.2 2D Examples with Constructively ALS Model

In this section, several 2D examples with the ALS model will be shown here, and both linear ALS line primitive and second order circle primitive will be involved. The results will be compared with the optimal designs obtained with discrete level set based optimization framework. Also, the efficiency will be compared between the parametric ST based algorithm and the LSQ based algorithm. The influence of the mesh density and the time step size will not be detailed in this chapter. Since the current approach does not include the topological derivative for the nucleation of solid material, the initial design (especially the holes) we construct will take the benchmark results as the references.

5.2.1 Multiple Local Minima of A Structural Optimization Problem

Unlike discrete level set based optimization approach that is close to the infinite-dimensional optimization, with proper incorporation with topological derivatives, local minima can be escaped for many cases [80, 81]. Using our parameterizations without topological derivatives, since we only have very limited set of selective primitives defined by their internal parameters, we will show that optimal results might get stuck at local minima with different initial shape layouts.

The minimum compliance design of a long cantilever beam as shown in Figure (5.1) is used to demonstrate the local minimum problem. The design domain is a rectangle with $L = 2$ and $H = 1$. A vertical concentrated load $F = 1$ acts on the middle of the right edge and the left edge of the cantilever beam is fixed. As in the current problem, the objective function is a linear function of the Young's modulus and the load, their values would not change the final design. With different values of the load and Young's modulus, the minimum point will not change although the value of the objective function at this point will be different.

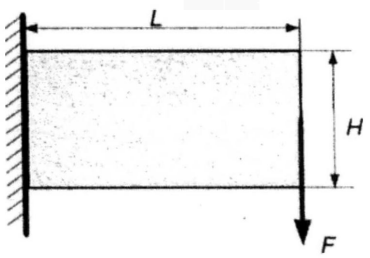
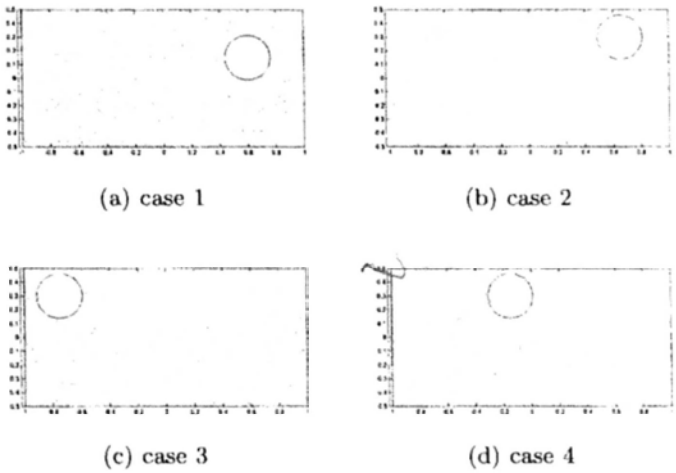


Figure 5.1: Problem setting for a long cantilever beam problem

The ALS circle primitive with the same radius as the initial geom-

etry is pre-collocated within the design domain for four different cases. The position of the circle x_0, y_0 are chosen as the design variables, and no volume constraint is imposed. The only difference between the four cases is the initial position of the circle within the domain is different. The mesh density we choose to perform FE analysis is 80×40 with density-based FEM scheme, and the parametric ST algorithm is chosen as the optimization algorithm with time step size as 1×10^{-4}. The four different initial designs are shown in Figure (5.2).



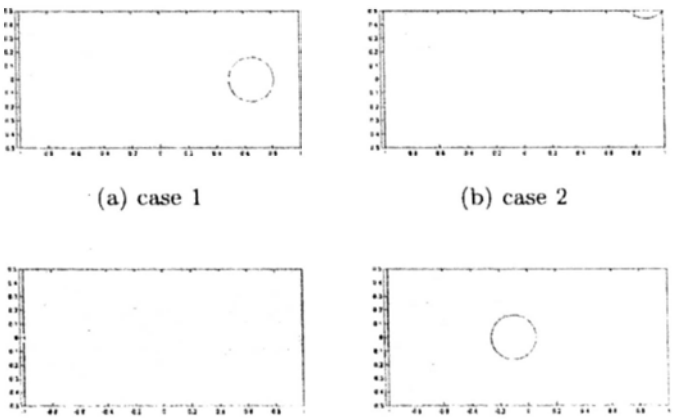
(a) case 1 (b) case 2

(c) case 3 (d) case 4

Figure 5.2: Different initial designs for local minimum problem

Clearly, for the mean compliance problem without volume constraint, the more material we employ in the domain the better the stiffness is. The final designs after a hundred steps are shown in Figure (5.3). As it can be seen that the final results are different with the different initial designs. In Figure (5.3(a)) and Figure (5.3(d)), clearly, the geometries are stuck at the local minima in the middle area, whereas in Figure (5.3(b)) and Figure (5.3(c)), the circles are 'escaping' from the local minima because the initial solutions we give are close to the

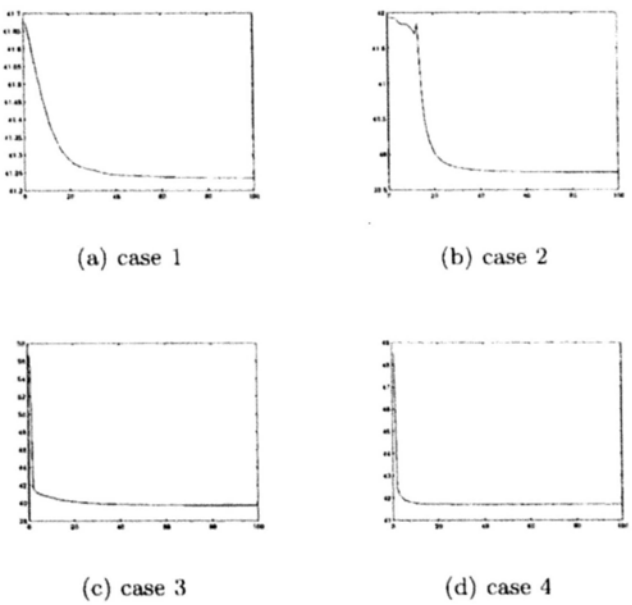
global minimum. Figure (5.4) shows the convergence process of the compliance objective for all four cases and Table (5.1) shows the final mean compliance of all four cases, clearly case 3 has the best mean compliance as expected.



(a) case 1 (b) case 2

(c) case 3 (d) case 4

Figure 5.3: Final designs for local minimum problem



(a) case 1 (b) case 2

(c) case 3 (d) case 4

Figure 5.4: Convergence history for the local minimum problems

Table 5.1: Mean compliance of all four cases

Mean Compliance	Case 1	Case 2	Case 3	Case 4
	41.2351	39.7445	39.7425	41.7144

5.2.2 A Long Cantilever Beam Design Problem

Here, all the geometric settings (length and width of the domain) as well as the loading settings remain the same as the settings of the previous example, and the maximum allowable volume fraction is set to be 0.5. Different initial designs will be tested with both the density-based FEM and the XFEM schemes, also, two different optimization algorithms will be tested separately to show their effectiveness. The optimal result obtained with the discrete LSM is shown in Figure (5.5) with the final mean compliance of 59.7056 [83]. All the other results should consider this as the benchmark.

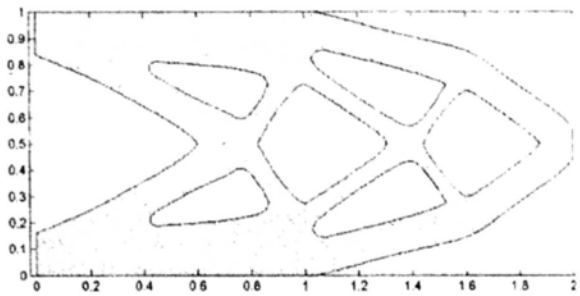


Figure 5.5: Optimal result with discrete level set method [83]

Case 1-1 with The Density Based FEM and the Parametric ST Algorithm

In this example, a complex initial design is constructed using 52 ALS line primitives (7 holes in the middle area) with predefined boolean

operations, as shown in Figure (5.6). Here we expect to use simple primitives to deliver the final complex shape. The time step size τ chosen here is 2×10^{-4} and the maximum allowable iterations is set to be 1600.

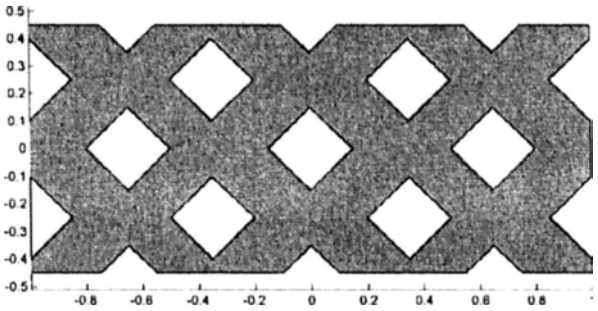
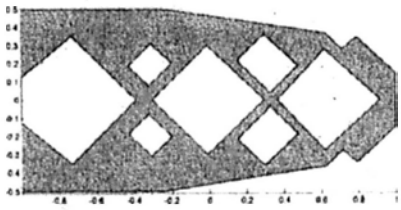


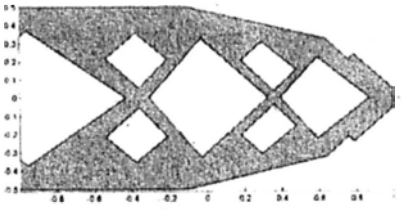
Figure 5.6: Initial design for case 1-1 of a long cantilever beam problem

After 1000 iterations, the design reaches its optimum, and both the topology and the shape are changed. It seems that with the big enough step size, the convergent speed with parametric ST method is not fast enough. The reason for this will be clarified later in this section, and we will realize the LSQ based optimization algorithm a better choice for our optimization problem. Figure (5.7) shows some of the intermediate results and the final result of case 1-1. The converged design has a mean compliance of 59.9330 and is not as good as the value obtained with discrete LSM. Considering our limited parameterization of the design, this result is desirable because we obtained an optimal shape that is represented by a few basic geometric primitives without losing too much performance, and this trade-off is quite favorable.

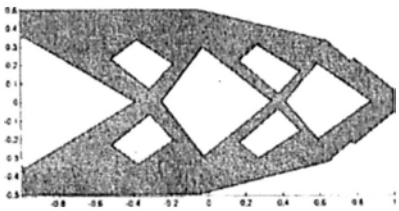
The convergence history of the objective function and the volume ratio are shown in Fig (5.8).



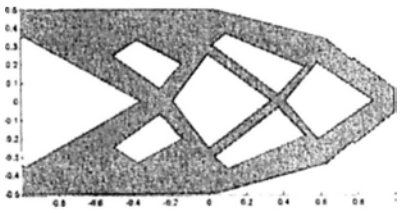
(a) step 10



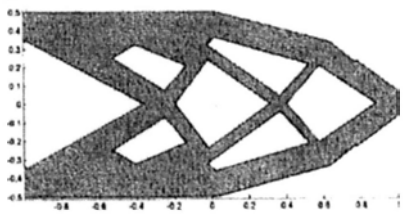
(b) step 80



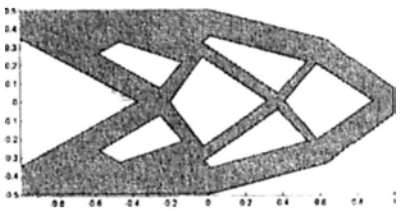
(c) step 250



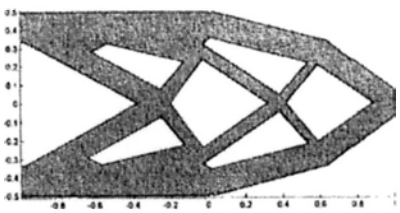
(d) step 400



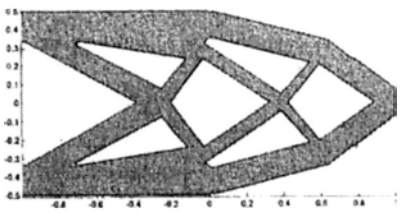
(e) step 550



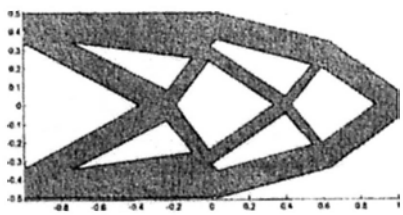
(f) step 650



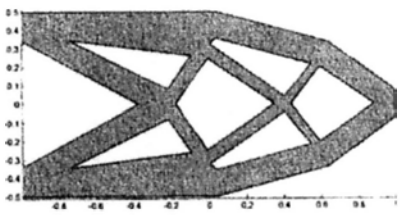
(g) step 750



(h) step 850

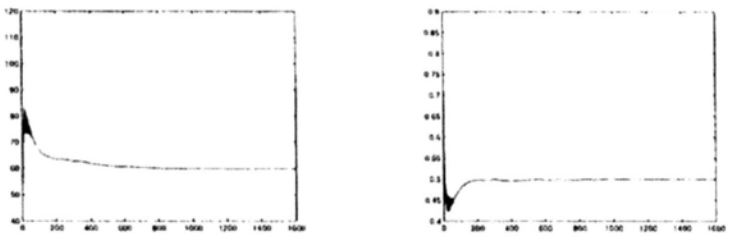


(i) step 1000



(j) step 1600

Figure 5.7: Intermediate results of case 1-1



(a) mean compliance (b) volume ratio

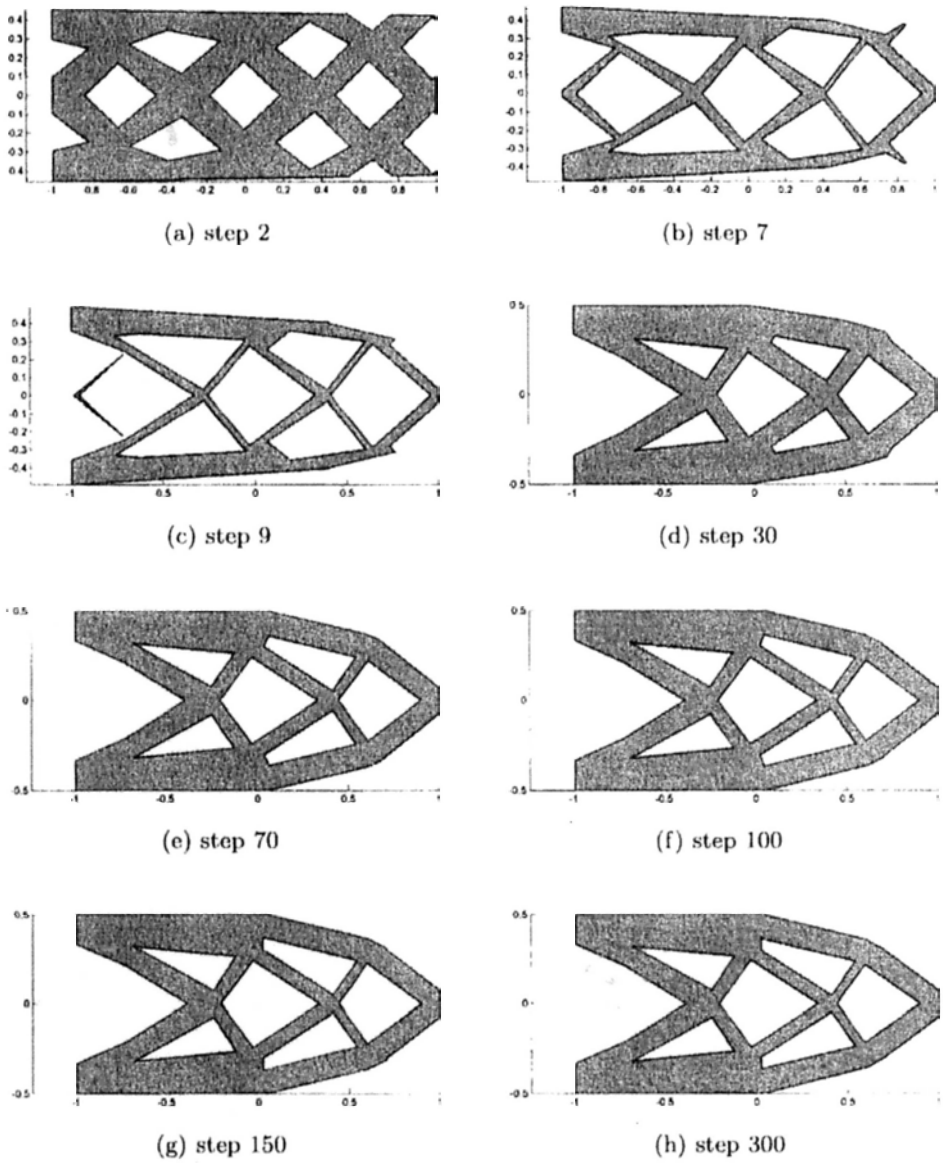
Figure 5.8: Convergence history of case 1-1

Case 1-2 with The Density Based FEM and The LSQ Algorithm

In this example, the initial domain is constructed using the same set of ALS primitives, the optimization algorithm chosen here is the LSQ based algorithm, the FEM scheme is the density-based FEM with the mesh density of 80×40, the time step size τ chosen here is 2×10^{-5} and the maximum allowable iterations is 300.

After about 150 iterations, the final design converges to the similar geometry as shown in case 1-1. Obviously, with a smaller time step size and the same initial design, the LSQ based algorithm is more efficient than the parametric ST algorithm. We can easily explain this by considering the line primitive with infinite length as a rigid body, then the translation and rotation of this primitive will determine the overall gesture of this line, which precisely correspond to the two design variables in Eq. (3.13). Therefore, there exists a dimensional problem for different design variables if we use the fixed time step size with the parametric ST algorithm because we cannot determine automatically the step size for design variables with different dimensions. However, with the fixed time step, if we choose the LSQ based optimization

algorithm, the $\boldsymbol{H}$ matrix in Eq. (4.25) adds weights between different design variables, and this accounts for the "harmonic" optimization process we observed. The intermediate results of case 1-2 are shown in Figure (5.9), and the final mean compliance is 59.9196.



(a) step 2 (b) step 7

(c) step 9 (d) step 30

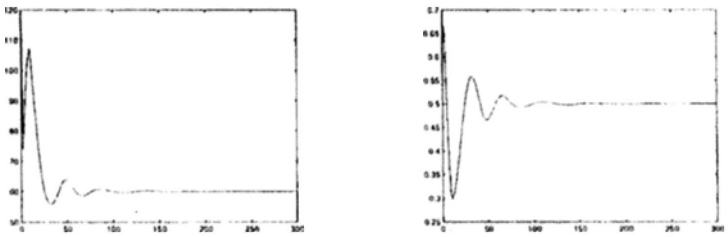
(e) step 70 (f) step 100

(g) step 150 (h) step 300

Figure 5.9: Intermediate results of case 1-2

By considering the residual between the infinite-dimensional time

velocity and the parametric time velocity, we can resolve the issue of dimensional problem for different design variables, and this will lead to a much smoother convergence process, as shown in Figure (5.10).



(a) mean compliance (b) volume ratio

Figure 5.10: Convergence history of case 1-2

Case 1-3 with The XFEM and The LSQ Algorithm

In this example, we have a domain resembling case 1-1 and 1-2, the difference is that we add eight more circle primitives into the initial design as shown in Figure (5.11). To acquire more accurate strain energy density along the structural boundary, we use the XFEM scheme instead with the mesh density of 80×40. The LSQ algorithm is used because its convergence speed is far better than the parametric ST algorithm. The time step size τ chosen here is 2×10^{-5} and the maximum allowable iterations is 300.

After about 300 iterations, the final design converges to the similar shape as shown in case 1 series. With the XFEM scheme, the final shape is more accurate than those delivered by the density-based FEM. It is evident that the first order line primitives and second order circles are able to mingle naturally together to deliver the optimal shape. The intermediate results of case 1-3 are shown in Figure (5.12) and

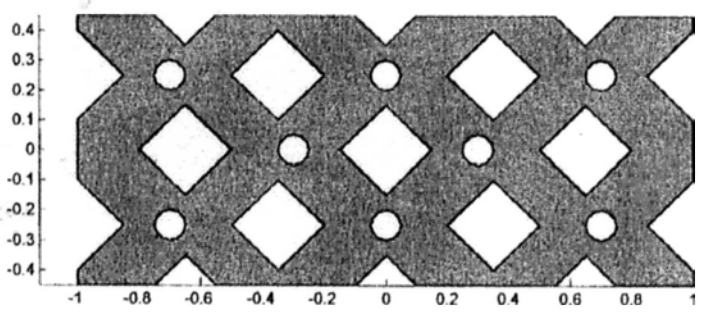
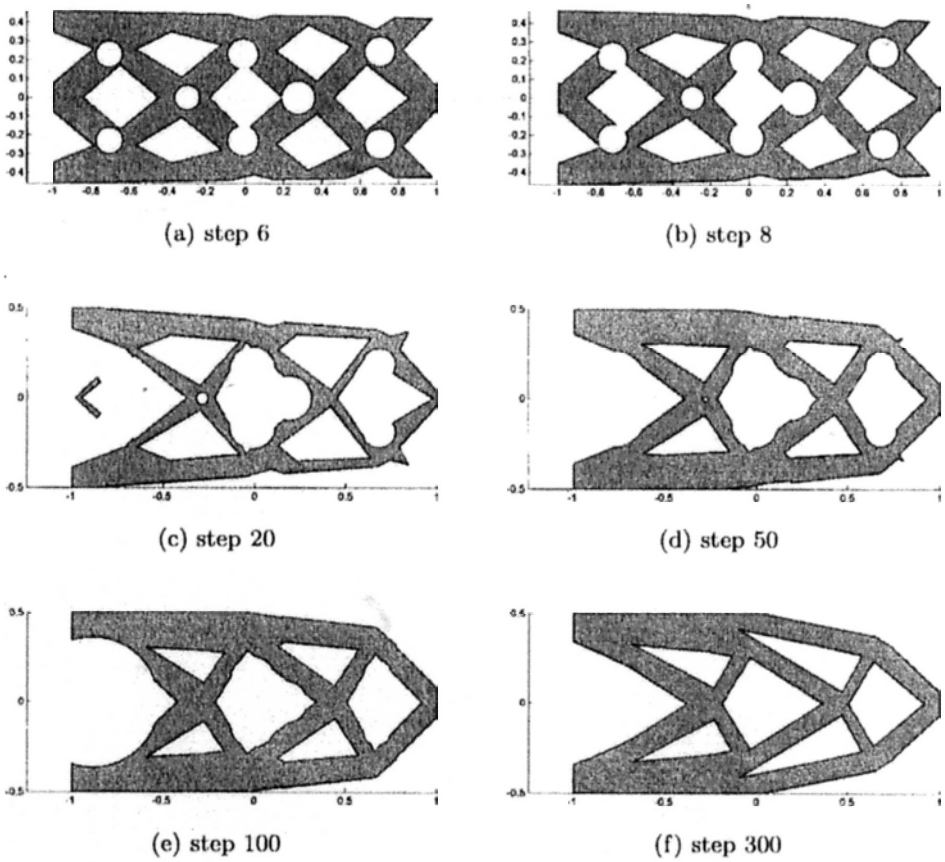


Figure 5.11: Initial design for case 1-3 of a long cantilever beam problem

final mean compliance is 60.1197. The convergence history is shown in Figure (5.13).



(a) step 6

(b) step 8

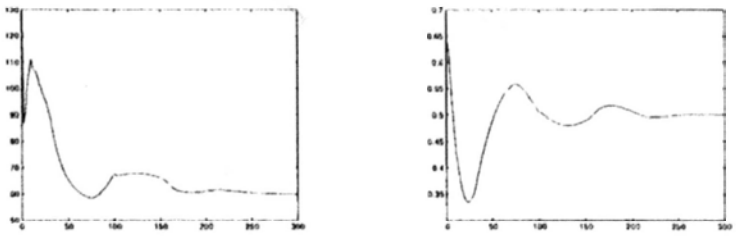
(c) step 20

(d) step 50

(e) step 100

(f) step 300

Figure 5.12: Intermediate results of case 1-3



(a) mean compliance (b) volume ratio

Figure 5.13: Convergence history of case 1-3

Case 2-1 with The Density Based FEM and The Parametric ST Algorithm

In this set of examples, the initial design in shown in Figure (5.14), the FEM scheme is the density-based FEM with the mesh density of 80×40, the time step size τ chosen here is 2×10^{-4} and the maximum allowable iterations is 1000. The optimization algorithm adopted is the parametric ST algorithm.

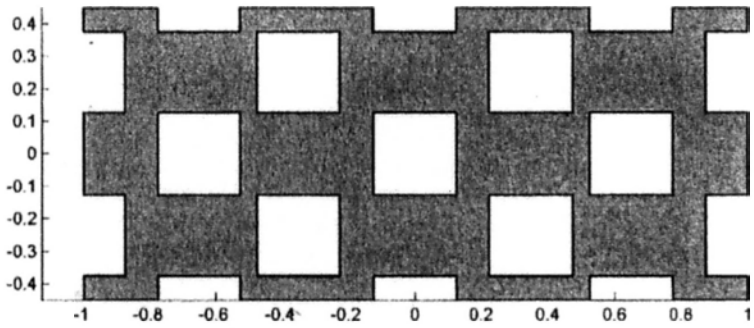
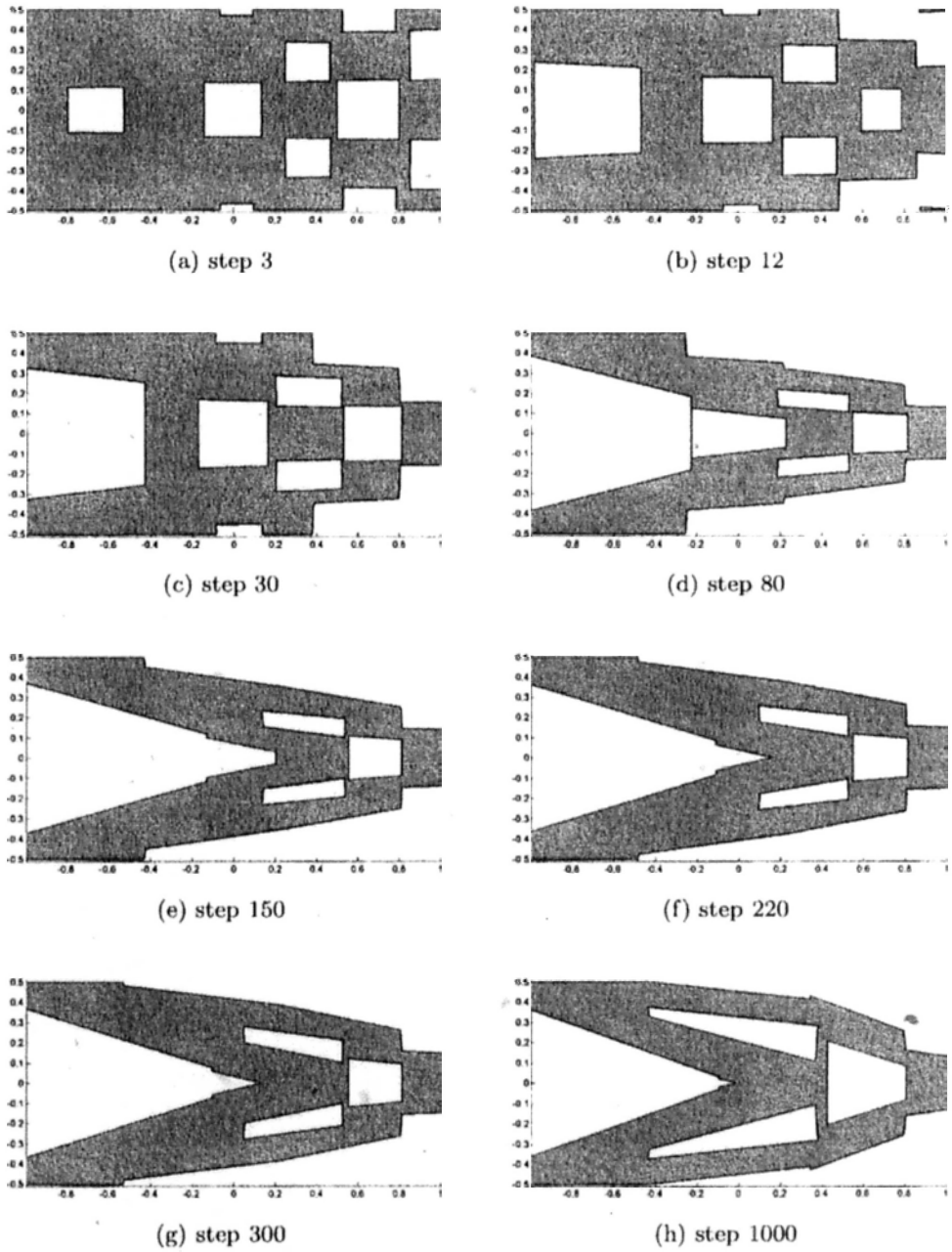


Figure 5.14: Initial design for case 2-1 of a long cantilever beam problem

The intermediate results of case 2-1 are shown in Figure (5.15). The convergence history is shown in Figure (5.16). We can see that since the parametric ST algorithm with fixed time step size cannot resolve the dimensional issue between different design variables, the optimal



(a) step 3

(b) step 12

(c) step 30

(d) step 80

(e) step 150

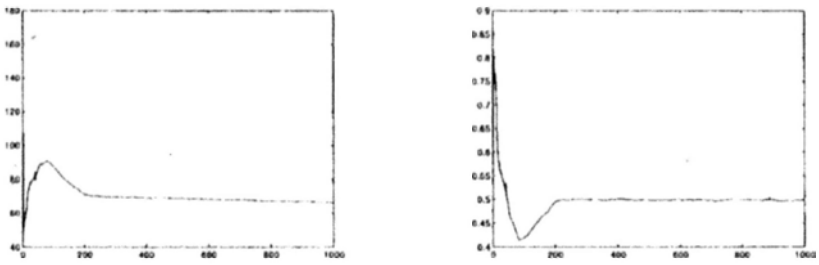
(f) step 220

(g) step 300

(h) step 1000

Figure 5.15: Intermediate results of case 2-1

shape is quite different from the results in case 1 series, and a lot of details are missing. The high mean compliance value of 66.1764 in this case is over 10% higher than the value in the benchmark owing to the abovementioned dimensional issue.

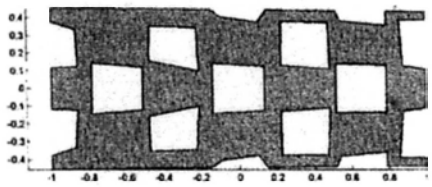


(a) mean compliance (b) volume ratio

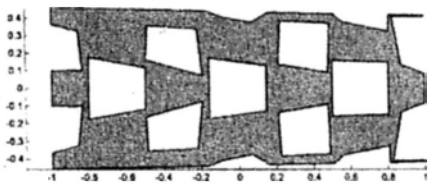
Figure 5.16: Convergence history of case 2-1

Case 2-2 with The Density Based FEM and The LSQ Algorithm

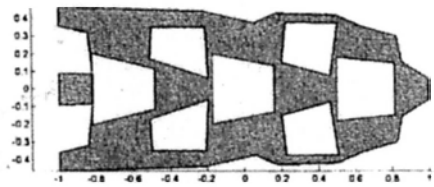
In case 2-2, we replace the parametric ST algorithm with the LSQ algorithm for a better convergence process, other settings are the same as case 2-1 except that the time step size is set to be 2×10^{-5} and the maximum allowable iterations is set to be 300. The intermediate results of case 2-2 are shown in Figure (5.17). We can observe from the final design that, the optimal result might be quite different with various initial settings. Since we only use a few ALS primitives to represent the shape of the design, the representation is not as sophisticated as the discrete LSM, and the finial design is dependent on the initial design. We can consider this phenomenon as the initial-design-dependent problem. The convergence history is shown in Figure (5.18) with the final mean compliance of 60.1197, which is higher than both the op-



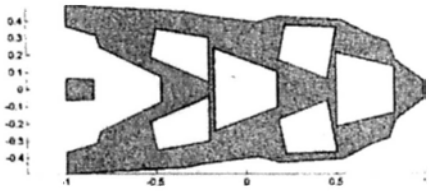
(a) step 5



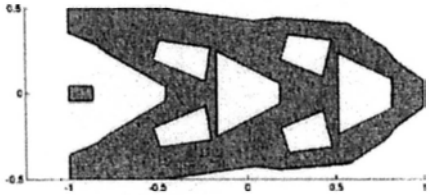
(b) step 12



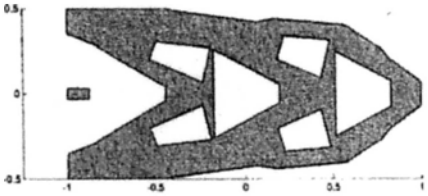
(c) step 18



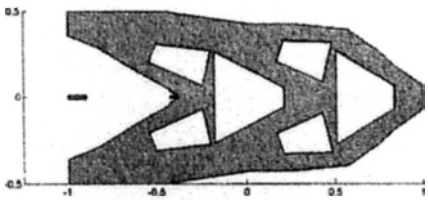
(d) step 30



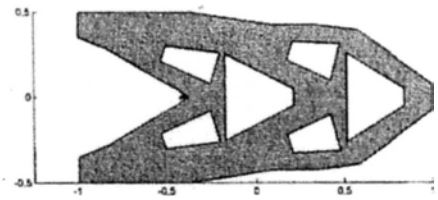
(e) step 70



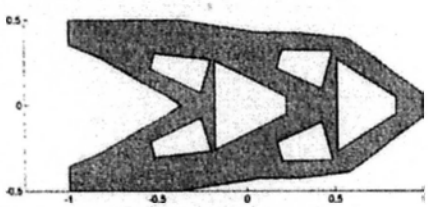
(f) step 100



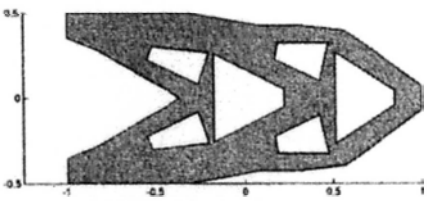
(g) step 150



(h) step 200



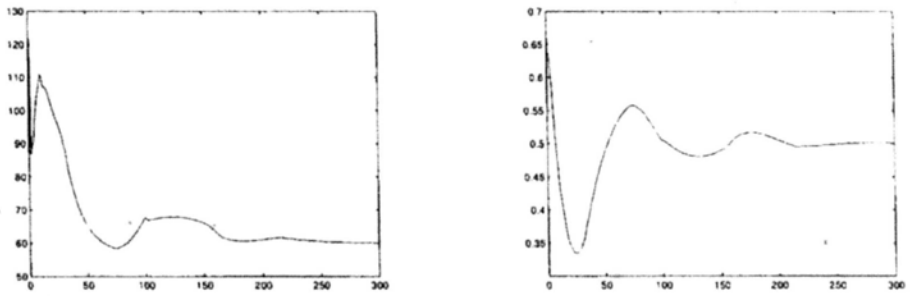
(i) step 230



(j) step 300

Figure 5.17: Intermediate results of case 2-2

timal results delivered by the discrete LSM and the results we get in case 1-1 or case 1-2. While compared with case 2-1, the final result preserves more geometric details during the evolution and the rotation and translation of each ALS line primitive is in harmony during the optimization. The final design is quite similar to the benchmark result in shape and the final mean compliance value is quite reasonable compared with case 2-1 which employs the the parametric ST algorithm.



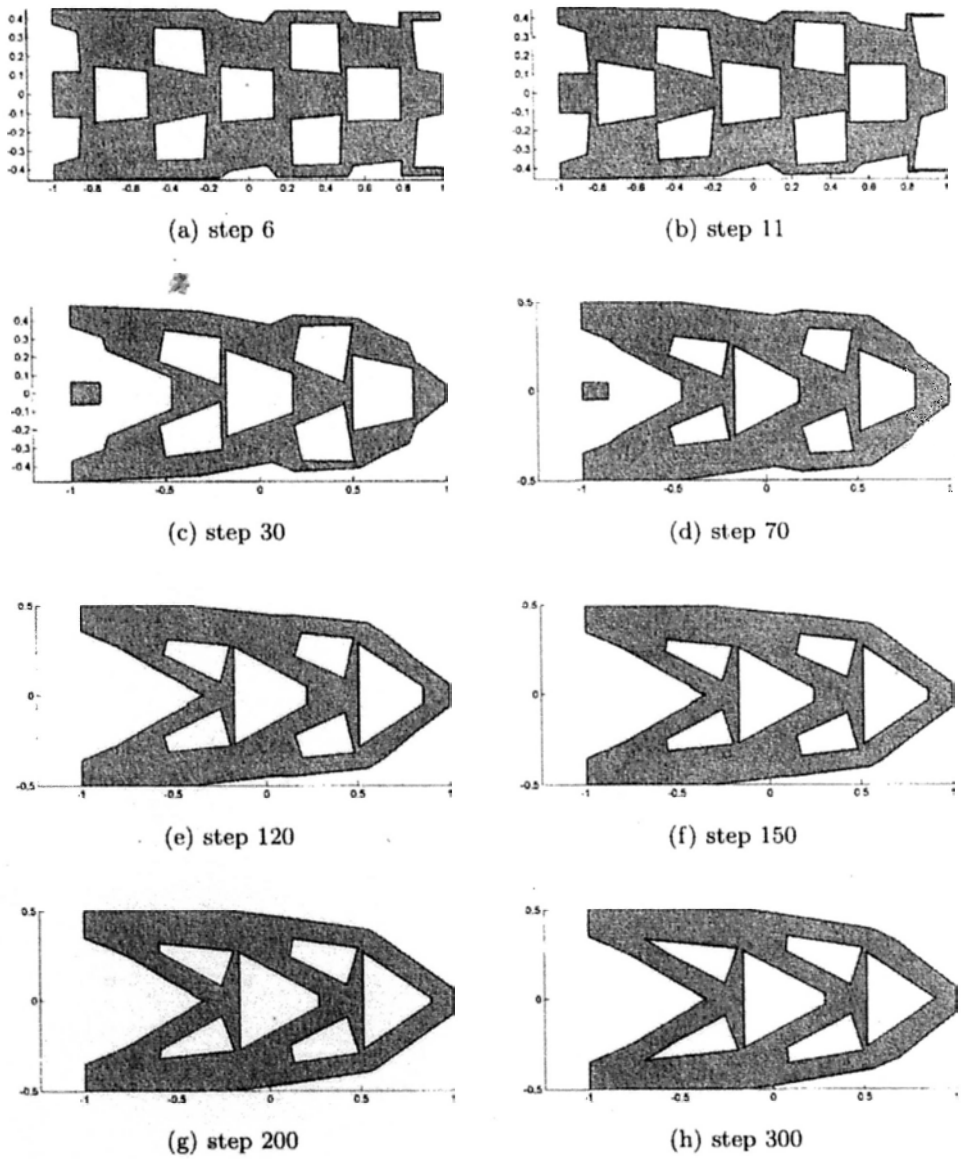
(a) mean compliance

(b) volume ratio

Figure 5.18: Convergence history of case 2-2

Case 2-3 with The XFEM and The LSQ Algorithm

In this example, we have the same initial design as case 2-1, all relevant parameters are remained unchanged except that the XFEM scheme is used instead of the density-based FEM to expect higher accuracy of the stress and strain fields. The intermediate results of case 2-3 are shown in Figure (5.19). As it can be seen, we have a more accurate computation of strain energy density distribution on the boundary with the XFEM scheme, which accounts for the more accurate optimal result as shown in Figure (5.19).



(a) step 6

(b) step 11

(c) step 30

(d) step 70

(e) step 120

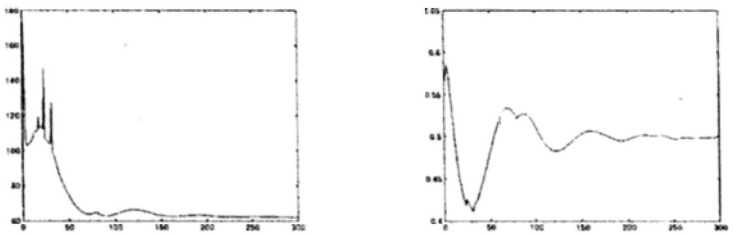
(f) step 150

(g) step 200

(h) step 300

Figure 5.19: Intermediate results of case 2-3

The convergence history is shown in Figure (5.20) with the final mean compliance of 61.6781.



(a) mean compliance (b) volume ratio

Figure 5.20: Convergence history of case 2-3

Case 3-1 with The XFEM and The LSQ Algorithm

In case 3-1, the number of ALS line primitives in the domain is different from the previous examples. The different initial design is shown in Figure (5.21), where 68 ALS line primitives are used and 10 quadrilateral shapes are distributed inside the design domain. The FEM scheme chosen is the XFEM with the mesh density of 80×40, the time step size τ chosen here is 2×10^{-5} and the maximum allowable iterations is 400. The optimization algorithm adopted is the LSQ algorithm.

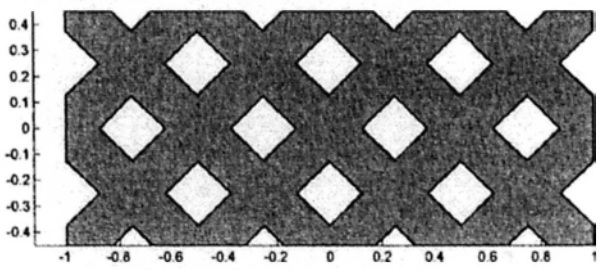


Figure 5.21: Initial design for case 3-1 of a long cantilever beam problem

The final results of case 3-1 is shown in Figure (5.22) and the con-

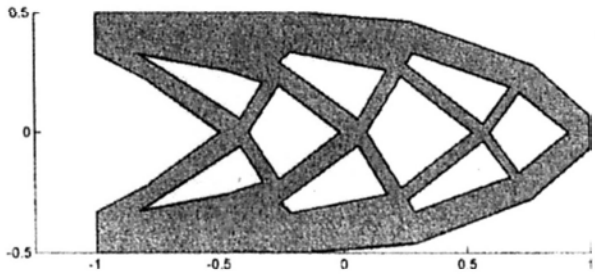
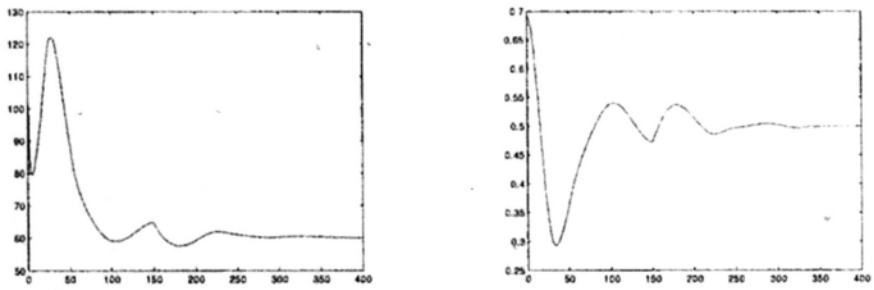


Figure 5.22: Final design for case 3-1

vergence history is shown in Figure (5.23). As it can be seen, the final optimal design has a different shape, more topological details are kept. The final mean compliance value is 59.9438, and it has very tiny difference compared to case 1-1 and case 1-2. For our optimization is based on the boundary variation, it is quite a initial-solution-dependent problem, and this explains the difference in shape of the final design for different cases. In this specific case, there might exist many local minima, and the extra details are actually local minima.



(a) mean compliance (b) volume ratio

Figure 5.23: Convergence history of case 3-1

Case 3-2 with The XFEM and The LSQ Algorithm

In case 3-2, we use hybrid ALS primitives to construct the initial design. The initial design is slightly different from case 3-1. Here 48 ALS line primitives and 3 ALS circle primitives are used and the number of holes in the initial design domain remains unchanged, except that the center holes are ALS circle primitives rather than quadrilateral shapes as shown in Figure (5.24). The FEM scheme is the XFEM with the mesh density of 80×40, the time step size τ chosen is 2×10^{-5} and the maximum allowable iterations is 300. The optimization algorithm adopted is the LSQ algorithm.

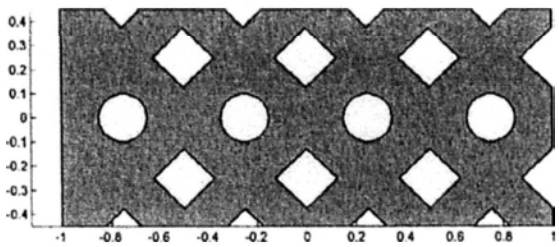


Figure 5.24: Initial design for case 3-2 of a long cantilever beam problem

The final design of case 3-2 is shown in Figure (5.25) and the convergence history is shown in Figure (5.26). The final optimal design has the similar internal details except that some of the holes are circles. The final mean compliance value is 61.7374, which is higher than case 3-1. This is another proof for the statement that the initial design and the local minimums will determine the final shape and topology of a design in a combined manner. The convergence history is shown in Figure (5.26).

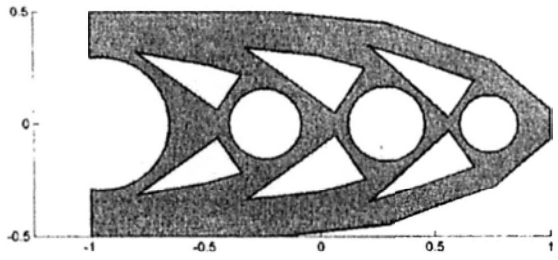
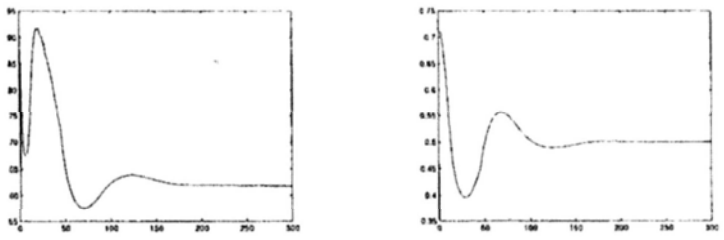


Figure 5.25: Final design for case 3-2



(a) mean compliance (b) volume ratio

Figure 5.26: Convergence history of case 3-2

Observations

From the results given in the series of numerical examples for the long cantilever beam, we have several observations.

1. With the proposed SDSA formulations and the ALS model, the original PDE solving procedure for the level set equation is converted to the sensitivity calculations regarding design parameters, which makes our problem quite a simple one. Compared with other free-form parameterization methods for implicit representation, such as the RFBs level set that usually involves a very large amount of design variables, our initial design domain is constructed with only a small number of primitives with their internal parameters as design variables, and this makes the SDSA computations very efficient.

2. The final optimal design usually includes a subset of primitives of the initial design and has relatively high practical engineering value.

3. The LSQ optimization algorithm is quite efficient if we use the fixed time step size for all design variables as has been proved numerically in related examples.

4. The final optimal design is initial-solution-dependent, and different initial designs may produce different final optimum.

5. Even though the initial design is complex enough to produce complex shape, the final design might look different when compared with the corresponding benchmark results. This can be explained as: with our parameterization, each primitive only has very limited design freedoms, and it is easy to be stuck at many of the local minima.

6. For all examples given above, the final mean compliance values are higher than the value in the benchmark. Take the final optimal result in case 1-3 for example, with this design, there still exists shape gradient residuals on some parts the boundary as shown in the enlarged portions in Figure (5.27), that means if we want to achieve the optimum from infinite-dimensional optimization point of view, more geometric details are needed.

5.2.3 A Short Cantilever Beam Design Problem

The minimum compliance design of a short cantilever beam is given in this section. The settings are shown in Figure (5.28). The design domain is a rectangle with $L = 1.5$ and $H = 1$. A vertical concentrated load $F = 1$ acts on the bottom right conner of the short cantilever beam. The maximum allowable volume fraction here is set to 0.35. The optimal design obtained with the discrete LSM is shown in Figure (5.29)

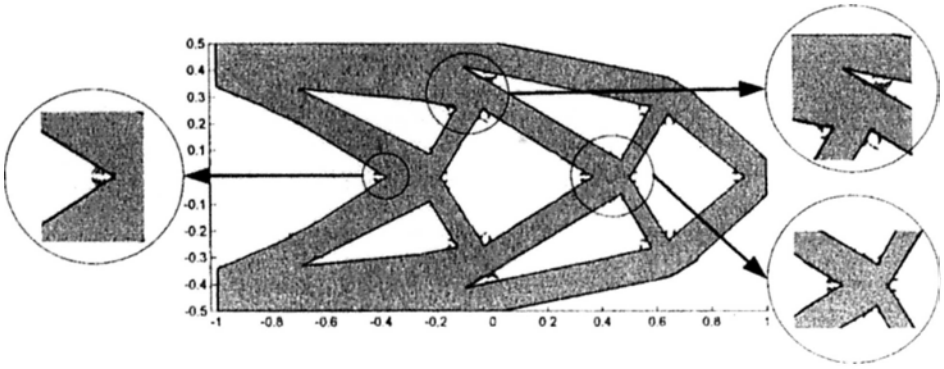


Figure 5.27: Shape gradient residuals for the final design of case 1-3

with the final mean compliance of 51.5395 [35], which can be considered as the benchmark for this problem.

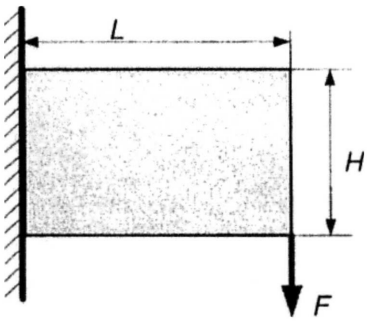


Figure 5.28: Problem setting for a short cantilever beam problem

Only one initial design is used here to show the correctness of the SDSA formulations and the effectiveness of the LSQ algorithm. The initial domain is constructed using 46 ALS line primitives, as illustrated in Figure (5.30). The FEM scheme chosen is the XFEM with the mesh density of 60×40, the time step size τ chosen here is 2×10^{-5} and the maximum allowable iterations is 500. Again, the optimization algorithm adopted is the LSQ algorithm. The intermediate results of this example are shown in Figure (5.31). Similar observations can be made that both shape and topology of this design can be changed si-

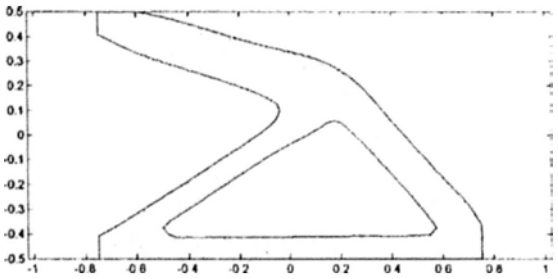


Figure 5.29: Benchmark result for the short cantilever beam problem [35]

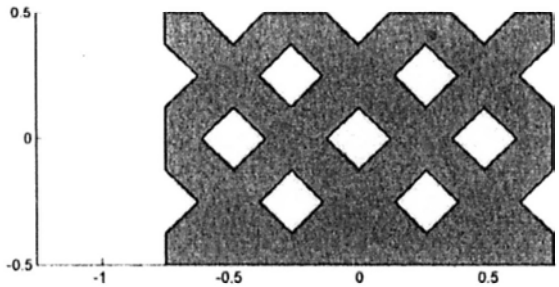
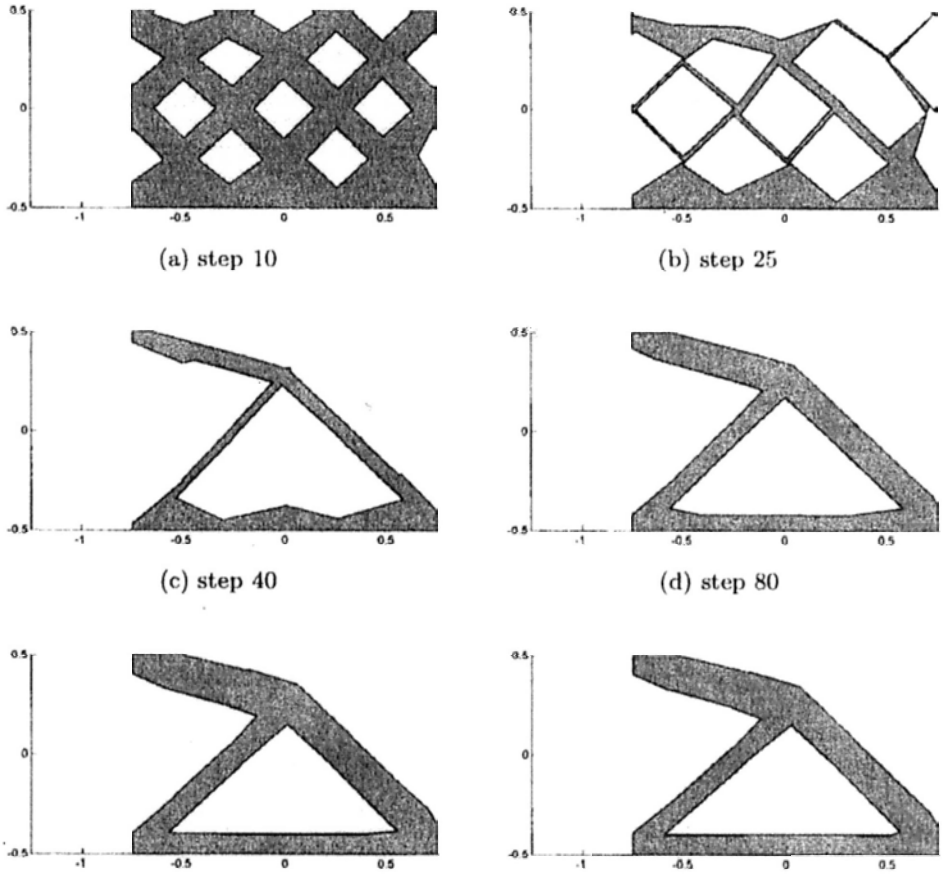


Figure 5.30: Initial design of a short cantilever beam problem

multaneously. The convergence history of this short cantilever beam example is shown in Figure (5.32) with the final mean compliance of 54.3272.

5.2.4 A Short Cantilever Beam with a Fixed Hole

An optimization problem with a fixed hole obstacle is considered in this example. This problem has been studies in [47, 84, 85]. As shown in Figure (5.33), the dimensions of the cantilever beam are: $L = 1.5$ and $H = 1$. The radius of the hole is $1/3$ and its center is determined by $W = 0.5$ and $D = 0.5$. A vertical concentrated load $F = 1$ is applied at the bottom right corner. The maximum allowed volume is

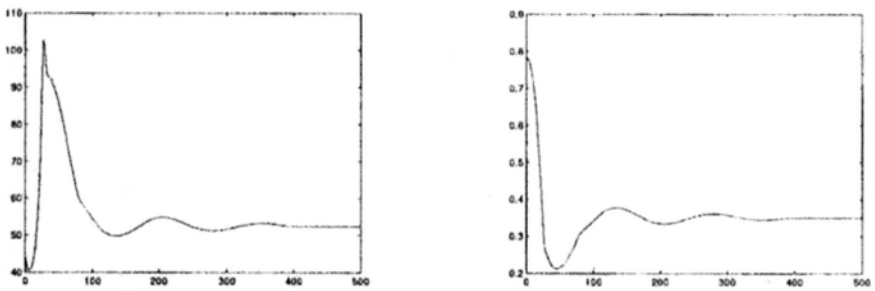


(a) step 10 (b) step 25

(c) step 40 (d) step 80

(e) step 150 (f) step 300

Figure 5.31: Intermediate results of the short cantilever beam problem



(a) mean compliance (b) volume ratio

Figure 5.32: Convergence history of the short cantilever beam problem

half of the volume of the design domain which is $1.5 - \pi/9$. With the traditional discrete level set based optimization, we need the regular mesh to carry out upwind scheme, and shapes as obstacles are not easy to be included in a natural way. Therefore, we use the result in [47] as the benchmark result, as illustrated in Figure (5.34).

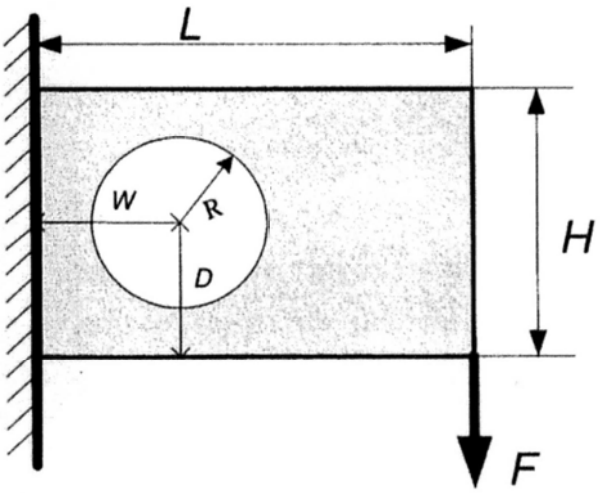


Figure 5.33: Problem setting for a cantilever beam problem with hole

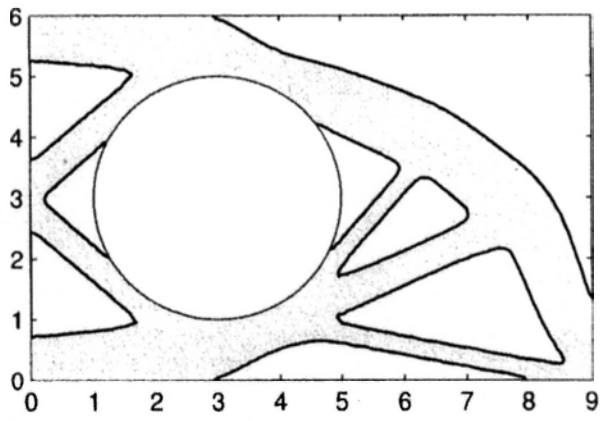


Figure 5.34: Benchmark for the short cantilever beam problem with hole [47]

With our shape parameterization and the SDSA formulations for different geometric primitives, it is easy to select different subset of

primitives as the design primitives, while the rest unselected could be considered as obstacles and the design parameters will not be updated. Here we use one initial design to show the shape and topology changing capabilities. The initial design domain constructed using 42 ALS line primitives and 4 ALS circle primitives, as shown in Figure (5.35). The circle primitive of the biggest radius is considered as the obstacle. In other words, this circle will not be considered as the design primitive and we do not calculate the design sensitivity regarding its internal parameters and do not update its position and radius.

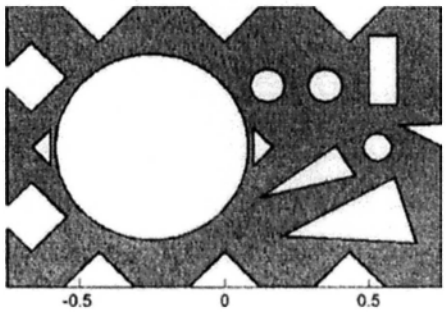
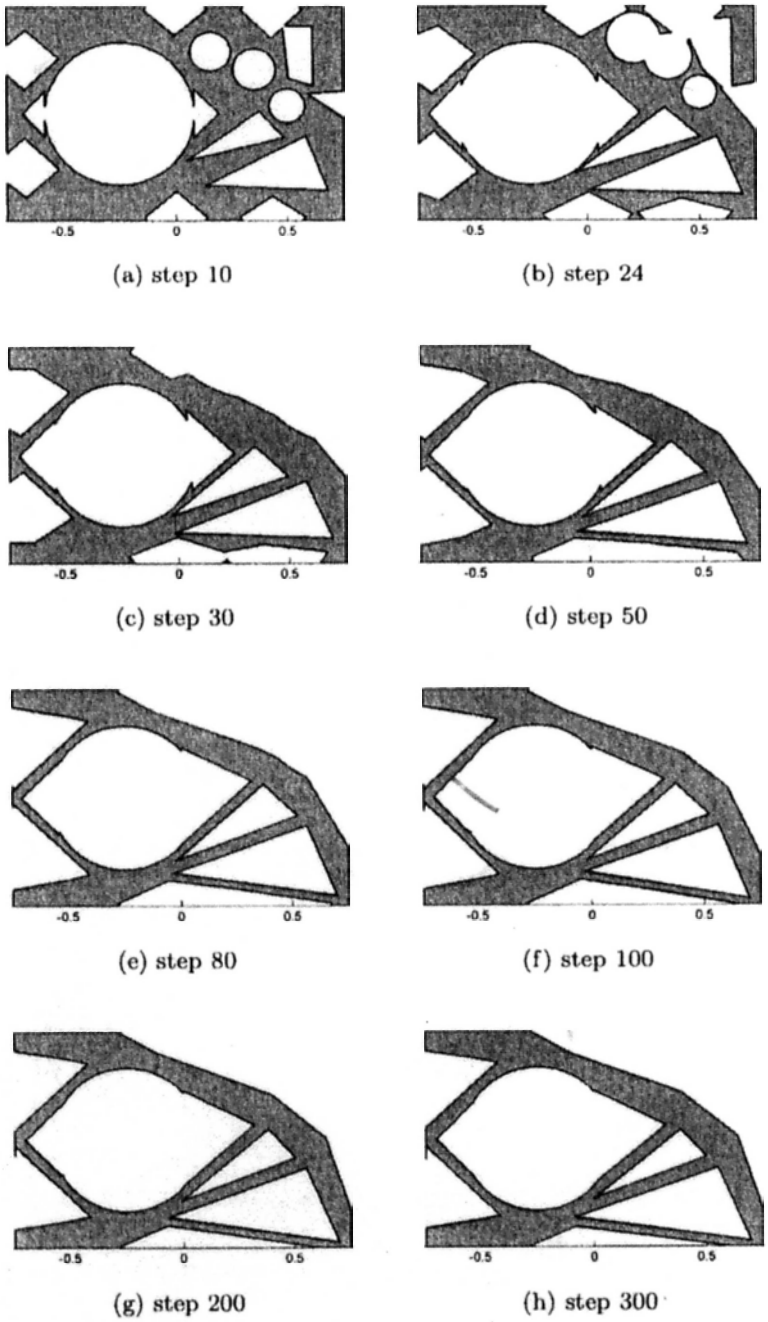


Figure 5.35: Initial design for the short cantilever beam problem with hole

The FEM scheme in this example is the XFEM with the mesh density of 60×40, the time step size τ chosen here is 2×10^{-5} and the maximum allowable iterations is 300. Again, the optimization algorithm adopted is the LSQ algorithm. The intermediate results of this example are shown in Figure (5.36).

Note that the final design has a similar shape and the same topology as the benchmark result, with only a few different details. Figure (5.37) shows the convergence history of this example. Figure (5.38) shows the shape gradient residual on the boundary of the final design. Almost everywhere on the boundary reaches optimum except for the enlarged



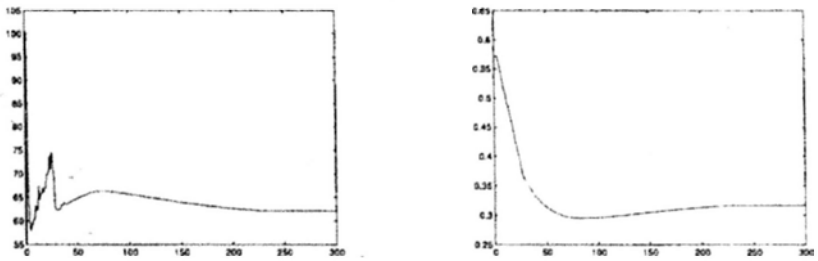
(a) step 10 (b) step 24

(c) step 30 (d) step 50

(e) step 80 (f) step 100

(g) step 200 (h) step 300

Figure 5.36: Intermediate results of the short cantilever beam problem with hole



(a) mean compliance (b) volume ratio

Figure 5.37: Convergence history of the short cantilever beam problem with hole

area. This means primitives in that area could not fully approximate the benchmark design. The enlarged portion actually is a circular curve in the benchmark design. However, it is a segment of a circle primitive with large radius value under our parameterization framework, and this is the best approximation to the benchmark design. If we want to reduce the residual, more details are needed there to fully approximate the benchmark design. For example, we could have more linear segments in that region.

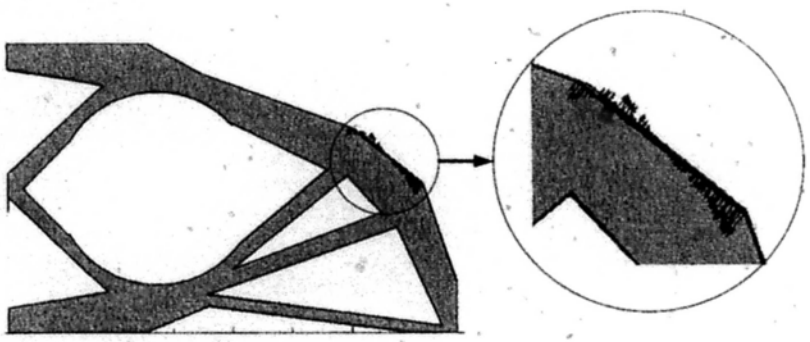


Figure 5.38: Shape gradient residuals for the short cantilever beam

5.3 Extension to Simple 3D Examples

In fact, we do not need to know the exact boolean operators that define a shape. As long as we can use the local operations to update the B-Rep data structure to form the new solid model, the optimization process could proceed till it reaches the optimum. In this section, our proposed SDSA formulations for a set of 3D parametric plane primitives are implemented with the ACIS kernel. Two simple examples are shown here, both of them are beam design problems as shown in Figure (5.39). One need to note that the examples given in this section only involves planar primitives and they are not genuine 3D examples. We only want to these simple examples to show the capability of the proposed formulations and the computational framework.

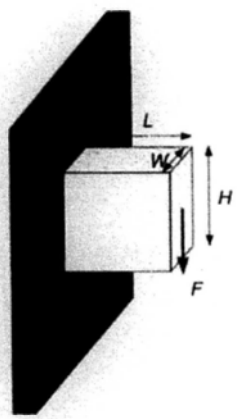
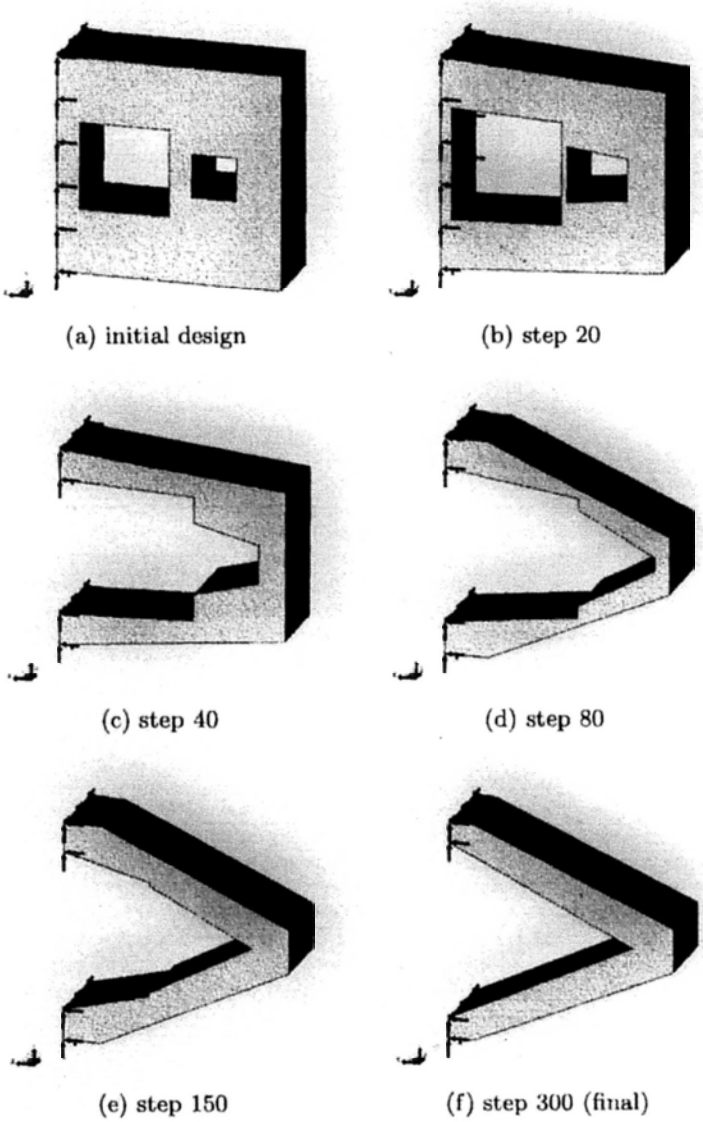


Figure 5.39: Problem definition for 3D beam

5.3.1 A Short Beam

For the mean compliance problem of a 3D short beam, we set $L = 100$, $H = 100$ and $W = 40$. The FEM scheme is density-based FEM and the mesh density is $20 \times 20 \times 8$. Figure (5.40) shows several intermediate results. The optimal result is just a two-bar structure.



(a) initial design (b) step 20

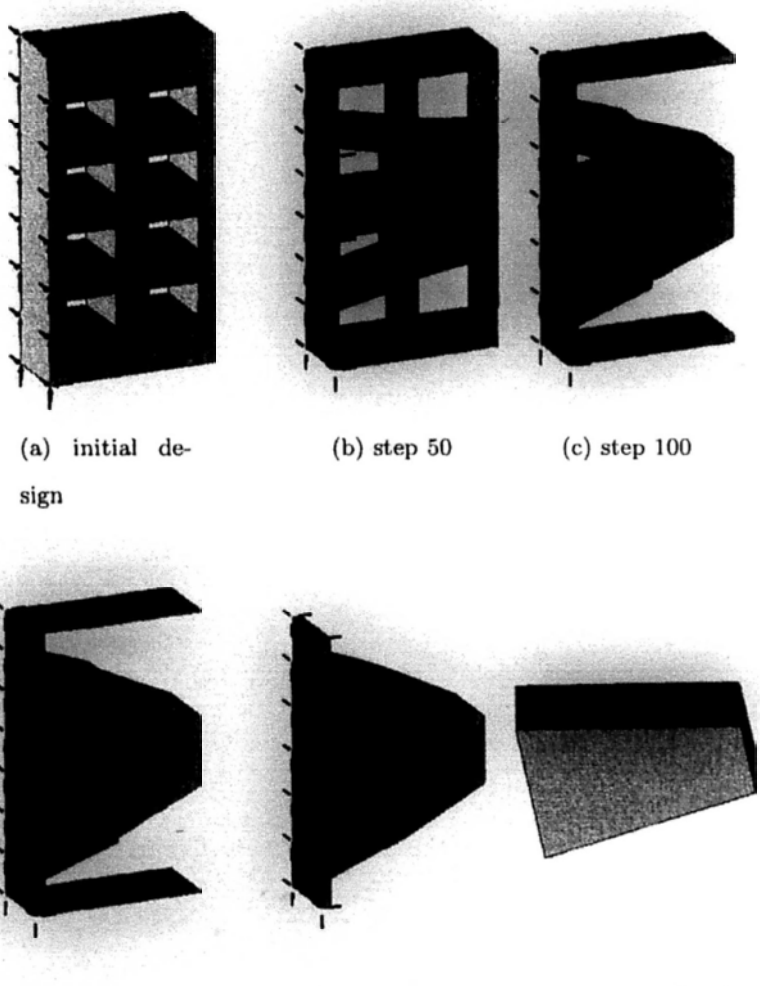
(c) step 40 (d) step 80

(e) step 150 (f) step 300 (final)

Figure 5.40: Intermediate results of the 3D short beam design

5.3.2 A High Beam

For the mean compliance problem of a 3D high beam, we set $L = 100$, $H = 200$ and $W = 40$. The FEM scheme is density-based FEM with the mesh density of $40 \times 20 \times 8$. The intermediate results are shown in Figure (5.41).



(a) initial design (b) step 50 (c) step 100

(d) step 120 (e) step 180 (f) step 500 (final)

Figure 5.41: Intermediate results of the 3D high beam design

□ **End of chapter.**

Chapter 6

Conclusions and Future Work

6.1 Conclusions

With the conventional discrete level set based structural optimization, postprocessing work is always a tedious job to turn the optimal design into a CAD model. In this thesis, the SDSA computation framework for both ALS model with implicit primitives and the the B-Rep model with parametric primitives are proposed for structural shape and topology optimization. The results delivered within our proposed frameworks have direct and intuitive geometric meanings.

Starting from Halmilton-Jacobi equation of the LSM, we come up with the concept of ALS, which is a natural extension of the LSM. The ALS model uses the half-space concept from CSG by considering all the coefficients as time dependent. A set of 2D and 3D primitives are selected to construct complex shapes with boolean operators, and the selected primitives will evolve over time. With the selective set of basic

primitives, we get the SDSA formulations for the all design parameters assuming that no geometric constraints are imposed on these primitives. Since the shape derivative formulation we utilize only involves boundary integration, we only need the differentiable property along the boundary. Therefore, as long as we can identify all independent design parameters in each independent primitive and calculate their normal design velocity, we can compute the corresponding design sensitivity. In the similar manner, the corresponding SDSA formulations are extended to parametrically represented primitives. We only need a small number of primitives to construct a solid, so the number of the design variables is quite small compared with other parameterization methods such as the RBFs level set, and this will make the SDSA computation very efficient.

For the ALS model with implicitly represented primitives, the geometry updating scheme is to re-perform the predefined boolean operations acting on all relevant primitives after the design parameters are updated. In fact, we do not need to know in advance about the the specific boolean operators that define a solid. As long as we know such a constructive expression exists, we can calculate the relevant SDSA. For the B-Rep model with parametric primitives in CAD system, as long as we can update the geometry with rules offered by local operations, we can update the geometry each step in a similar way with simultaneous shape and topological changes.

The SDSA formulations are the basis for the gradient-based optimization algorithms. Since line search method for best time step size is too costly in practice, we choose the fixed time step size for our proposed optimization algorithms. The parametric ST algorithm is

adopted as the first algorithm in current research. The dimensional problem is then discussed with the same fixed time step size for different dimensional variables. We thereafter proposed the LSQ based optimization algorithm which can guarantee the objective L moves towards the descent direction, meanwhile, can reconcile the existing dimensional issues using the $\boldsymbol{H}$ matrix. Since the scale of $\boldsymbol{H}$ is always very small, it will not affect the overall optimization efficiency.

Both 2D and 3D numerical examples are given in this thesis, several benchmark examples are used for comparison purpose. It is shown that with the parametric ST algorithm, the convergence process is quite slow due to the dimensional problems mentioned above, whereas the proposed LSQ based algorithm is quite efficient. As we have observed, no matter how a solid is represented (implicitly or parametrically), this limited parameterization of basic primitives might get themselves trapped at a local minimum. Also the initial design has critical influence on the production of the final design. By observing the shape gradient residuals on the boundary of several optimal designs, we know that more geometric details are needed for many cases.

With our computational framework, the final design is compatible with CAD data structure and detail design can be conducted directly thereafter.

There are several limitations of the current research. Firstly, since we only derived the SDSA formulations for basic primitives, they are easy to be stuck at many of the local minima during optimization. Secondly, in the 2D ALS framework, the blending of the individual primitives using boolean operators are conducted in a sequential way for simplicity. However, to make the approach more practical, in the

future, we should blend the primitives in a tree structure. In the 3D B-Rep based framework, the examples are limited by the topology changing capability of geometry kernel and further improvements need to be explored for more sophisticated results. Thirdly, the convergence criteria in current research is determined either by the maximum number of optimization steps or by a pre-defined small numerical value, and the maximum number of optimization steps is usually determined on a case by case basic. Therefore a more mature and unified criteria should be further studied.

6.2 Future Work

There are a lot of possibilities for future work of the presented optimization framework. The numerical examples clearly demonstrate the feasibility of the proposed methods in solving the minimum mean compliance problems. However, the performance of these methods in frequency optimization and stress optimization problems still need to be investigated.

The current research includes a set of basic primitives and treats all primitives independently. However, as we can observe, even though the different type of primitives can blend well (such as line primitive and circle primitive), some of the critical features (such as fillet) may not be properly preserved during the evolution process. We expect to use a unique representation to represent first-order and second-order primitives while preserving the important features during optimization in the future study.

Even though some 3D examples are given, only the planar primi-

tives are involved. The presented methods should be extended to more general three dimensional problems with multiple types of primitives. Although there is no theoretical difficulty, some numerical issues need to be considered.

In this study, only the parametric ST and the LSQ based optimization algorithms are proposed and verified, while other optimization algorithms are not yet tested. We hope to get a better convergence speed with other advanced mathematical programming techniques in the near future.

For the initial-design-dependent issue, if the initial design is poor, the shape may converge to a unsatisfactory design, and the reparameterizations based on the analysis of shape gradient density residuals should be discussed in the future. An expectation for the reparameterizations is to make the result closer to the optimal designs in the benchmark results.

□ **End of chapter.**

Bibliography

[1] R. J. Yang, D. L. Dewhirst, J. E.Allison, and A.Lee. Shape optimization of connecting rod pin end using a generic model. *Finite Elem. Anal. Des.*, 11:257–264, 1992.

[2] K. H. Chang and K. K. Choi. A geometry-based parameterization method for shape design of elastic solids. *Mech. Struct. & Mach.*, 20(2):215–252, 1992.

[3] D. A. Tortorelli, J. A. Tomasko, T. E. Morthland, and J. A. Dantzig. Optimal design of nonlinear parabolic systems - part ii: Variable spatial domain with applications to casting optimization. *Comput. Methods Appl. Mech. Engrg.*, 113(1-2):157–172, 1994.

[4] D. A. Tortorelli. A geometric representation scheme suitable for shape optimization. *Mech. Struct. & Mach.*, 21(1):95–121, 1993.

[5] A. Longo, J. Unzueta, E. Schaeidt, A. Alvarez, and J. J. Anza. A general related variational approach to shape optimum design. *Advances in Engineering Software*, 16:135–142, 1993.

[6] W. H. Zhang, P. Beckers, and C. Fleury. A unified parametric design approach to structural shape optimization. *Int. J. Numer. Methods Eng.*, 38:2283–2292, 1995.

[7] S. Chen and D. A. Tortorelli. Three-dimensional shape optimization with variational geometry. *Structural Optimization*, 13:81–94, 1997.

[8] I. E. Sutherland. Sketch pad a man-machine graphical communication system. In *Proc. of Spring Joint Computer Conference*, 1964.

[9] R. C. Hillyard and L. C. Braid. Characterization of Non-ideal Shapes in Terms of Dimensions and Tolerances. *Computer Graphics*, 12:234–238, 1978.

[10] V. C. Lin, D. C. Gossard and R. A. Light. Variational Geometry in Computer-Aided Design. *Computer Graphics*, 15(3):171-177, 1981.

[11] R. A. Light. *Symbolic Dimensioning in Computer-Aided Design.* M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, February 1980.

[12] V. C. Lin. *Three-dimensional variational geometry in computer-aided design.* M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1981.

[13] R. A. Light and D. C. Gossard. Modification of Geometric Models through Variational Geometry. *Computer Aided Design*, 14:209–214, 1982.

[14] R. A. Light and D. C. Gossard. Variational Geometry: A New Method for Modifying Part Geometry for Finite Element Analysis. *Comp. & Struct.*, 17:903–909, 1983.

[15] C. Hoffmann. Summary of Basic 2D Constraint Solving. *Intl. J. Prod. Lifecycle Mgmt.*, 1:143–149, 2006.

[16] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Engrg.*, 71:197–224, 1988.

[17] B. Hassani and E. Hinton. *Homogenization and Structural Topology Optimization: Theory, Practice and Software.* Springer, London, 1999.

[18] G. Allaire. *Shape Optimization by the Homogenization Method.* Springer, New York, 2001.

[19] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications.* Springer-Verlag, Berlin, 2003.

[20] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Struct. Multidisc. Optim.*, 1:193–202, 1989.

[21] Y. M. Xie and G. P. Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49(5):885–896, 1993.

[22] Y. M. Xie and G. P. Steven. *Evolutionary Structural Optimization.* Springer-Verlag London Limited, UK, 1997.

[23] D. Reynolds, J. P. McConnachie, W. Bettess, C. Christie, and J. W. Bull. Reverse adaptivity - a new evolutionary tool for structural optimization. *Int. J. Numer. Methods Eng.*, 45:529–552, 1999.

[24] X. Huang and Y. M. Xie. *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications.* John Wiley & Sons, Ltd, 2010.

[25] H. A. Eschenauer, H. A. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Struct. Multidisc. Optim.*, 8:42–51, 1994.

[26] H. A. Eschenauer and A. Schumacher. Topology and shape optimization procedures using hole positioning criteria - theory and applications. In G. I. N. Rozvany, editor, *Topology Optimization in Structural Mechanics*, volume 374 of *CISM International Centre for Mechanical Sciences. Courses and Lectures*, pages 135–196. Springer, Wien, 1997.

[27] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *J. Comput. Phys.*, 163(2):489–528, 2000.

[28] S. Osher and J. A. Sethian. Front propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 78:12–49, 1988.

[29] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, UK, 2nd edition, 1999.

[30] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer-Verlag, New York, 2002.

[31] S. J. Osher and F. Santosa. Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum. *J. Comput. Phys.*, 171(1):272–288, July 2001.

[32] G. Allaire, F. Jouve, and A. M. Toader. A level-set method for shape optimization. *C. R. Acad. Sci. Paris, Ser. I*, 334:1125–1130, 2002.

[33] G. Allaire, F. Jouve, and A. M. Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, 194:363–393, 2004.

[34] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Engrg.*, 192:227–246, 2003.

[35] M. Y Wang and X. Wang. PDE-driven level sets, shape sensitivity and curvature flow for structural topology optimization. *Computer Modeling in Engineering & Sciences*, 6(4):373–395, 2004.

[36] S. Y. Wang and M. Y. Wang. Structural shape and topology optimization using an implicit free boundary parameterization method. *Computer Modeling in Engineering & Sciences*, 13(2):119–147, 2006.

[37] S. Y. Wang and M. Y. Wang. Radial basis functions and level set method for structural topology optimization. *Int. J. Numer. Methods Eng.*, 65(12):2060–2090, 2006.

[38] M. Y. Wang and S. Y. Wang. *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, chapter Parametric Shape and Topology Optimization with Radial Basis Functions, pages 13–22. Springer Netherlands, 2006.

[39] P. Wei and M. Y. Wang. Parametric structural shape and topology optimization method with radial basis functions and level-set method. In *Proc. of ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 32nd Design Automation Conference*, 2006.

[40] P. Wei. *Level Set Methods for Shape and Topology Optimization of Structures*. PhD thesis, The Chinese University of Hong Kong, June 2007.

[41] Z. Luo, M. Y. Wang, S. Y. Wang, and P. Wei. A level set-based parameterization method for structural shape and topology optimization. *Int. J. Numer. Methods Eng.*, 76(1):1–26, 2008.

[42] F. Y. Lui, M. Y. Wang, and Q. Xia. Parametric shape and topology optimization via radial basis functions, partition of unity and level set method. In *Proc. of 5th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, 2008.

[43] H. S. Ho, and M. Y. Wang. Parametric shape and topology optimization with radial basis functions and partition of unity method. In *Proc. of Second International Symposium on Computational Mechanics (ISCM II) in conjunction with Twelfth International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science (EPMESC XII)*, 2009.

[44] P. Wei and M. Y. Wang. A piecewise density function method for structural shape and topology optimization. In *proceeding of 7th World Congress on Structural and Multidisciplinary Optimization(WCSMO-7)*, pages 1959–1968, 2007.

[45] P. Wei and M. Y. Wang. Piecewise constant level set method for structural topology optimization. *Int. J. Numer. Methods Engng.*, 78(4):379–402, 2009.

[46] X. H. Xing, M. Y. Wang. Structural topology optimization using finite element based level set methods. In *Proc. of NEFEMS World Congress*, 2009.

[47] X. H. Xing. *A Finite Element Based Level Set Method for Structural Shape and Topology Optimization.* PhD thesis, The Chinese University of Hong Kong, February 2009.

[48] J. Chen, V. Shapiro, K. Suresh, and I. Tsukanov. Shape Optimization with Topological Changes and Parametric Control. *Int. J. Numer. Methods Engng.*, 71:313–346, 2007.

[49] J. Q. Chen. *Shape Optimization Using Constructive Representations.* PhD thesis, University of Wisconsin-Madison, April 2007.

[50] T. T. Robinson, C. G. Armstrong, H. S. Chua, C. Othmer, T. Grahs. Sensitivity-based optimization of parameterised CAD geometries. In *Proc. of 8th World Congress on Structural and Multidisciplinary Optimization*, 2009.

[51] H. Voelcker and A. Requicha. Geometrical modeling of mechanical parts and processes. *IEEE Computer*, 12:48–57, 1977.

[52] I. Braid. The sysnthesis of solids bounded by many faces. *Comm. ACM*, 18:209–216, 1975.

[53] E. J. Haug, K. K. Choi and V. Komkov. *Design Sensitivity Analysis of Structural Systems*. Academic Press, 1986.

[54] J. Sokolowski and J. P. Zolesio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer, 1992.

[55] K. K. Choi and N. H. Kim. *Structural Sensitivity Analysis and Optimization I*. Springer, 2005.

[56] J. Chen, M. Freytag, and V. Shapiro. Shape Sensitivity of Constructive Representations. In *Proc. of the 2007 ACM Symposium on Solid and Physical Modeling*, 2007.

[57] J. Chen, M. Freytag, and V. Shapiro. Shape sensitivity of constructively represented geometric models. *Comput. Aided Geom. D.*, 25(7):470–488, 2008.

[58] Y. L. Mei, X. M. Wang and G. D. Cheng. A feature-based topological optimization for structure design. *Adv. Eng. Software*, 39:71–87, 2008.

[59] L. V. Miegroet. 3D Shape Optimization with X-FEM and a Level Set Constructive Geometry Approach. In *Proc. of the 8th World Congress on Structural and Multidisciplinary Optimization*, 2009.

[60] K. K. Choi and N. H. Kim. *Structural Sensitivity Analysis and Optimization II*. Springer, 2005.

[61] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-Based Fast Local Level Set Method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[62] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 71(74):347–377, 1987.

[63] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capture schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[64] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.*, 83:32–78, 1989.

[65] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[66] X. D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, November 1994.

[67] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[68] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

[69] R. Courant, K. Friedrichs and H. Lewy. Ueber die partiellen differenzgleichungen der mathematische physi. *Math Ann.*, 100:32-74, 1928.

[70] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann Pub., 1989.

[71] M. E. Mortenson. *Geometric Modeling.* 3nd ed., Industrial Press, NY, 2006.

[72] M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, Md, 1988.

[73] M. E. Mortenson. *Computer Graphics Handbook: Geometry and Mathematics.* Industrial Press Inc, 2006.

[74] http://www.spatial.com/products/3d-acis-modeling

[75] T. J. Barth and J. A. Sethian. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *J. Comput. Phys.*, 145(1):1–40, September 1998.

[76] W. Achtziger. Topology optimization of discrete structures – an introduction in view of computational and nonsmooth aspects. In G.I.N. Rozvany, editor, *Topology Optimization in Structural Mechanics*, pages 57–100. Springer, 1997.

[77] A. D. Belegundu and T. R. Chandrupatla. *Optimization Concepts And Applications In Engineering.* Prentice Hall, 1 edition, April 1999.

[78] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, 1999.

[79] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[80] M. Y. Wang and P. Wei. Topology optimization with level set method incorporating topological derivatives. In *Proc. of 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO6)*, Rio de Janeiro, Brazil, May 2005.

[81] G. Allaire and F. Jouve. Coupling the Level Set Method and the Topological Gradient in Structural Optimization. *Solid Mechanics and Its Applications*, 137:3–12, 2006.

[82] P. Wei, M. Y. Wang and X. Xing. A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design*, 42(8):708-719, 2010.

[83] Q. Xia. *Boundary and Material in Structural Optimization*. PhD thesis, The Chinese University of Hong Kong, August 2007.

[84] O. Sigmund. A 99 line topology optimization code written in MATLAB. *Struct. Multidisc. Optim.*, 21(2):120–127, 2001.

[85] T. Belytschko, S. P. Xiao, and C. Parimi. Topology optimization with implicit functions and regularization. *Int. J. Numer. Methods Eng.*, 57(8):1177–1196, 2003.