A Finite Element Based Level Set Method for Structural Topology Optimization

XING, Xianghua

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

February 2009

Thesis/Assessment Committee

Professor W. H. Liao (Chairman)

Professor Michael Y. Wang (Supervisor)

Professor Charlie C. L. Wang (Committee member)

Professor Y. H. Chen (External examiner)

摘要

本文使用一种基于有限单元法的水平集方法来解决结构拓扑优化问题。

使用水平集方法进行结构拓扑优化，是一种新兴的并已得到广泛关注和研究的方法。该方法通常分为两个阶段：应力分析阶段和基于水平集方法的边界演化阶段。前者通常由有限元法求解；后者通常由有限差分法求解。本文的目的是将两个阶段统一在有限单元法的体系下。此外，基于有限单元法的水平集方法更适合于求解不规则设计域上的结构优化问题。

由于水平集方程是一个双曲型偏微分方程，传统的伽辽金型有限元法会产生振荡的数值解。本文采用 SDFEM (streamline diffusion finite element method) 求解水平集方程，从而得到稳定的数值解。该方法的优势在于：方程组的系数矩阵对称正定，且为常矩阵，因此非常便于求解。此外，该系数矩阵类似于结构动力学中的质量矩阵，因此“集中系数矩阵”的方法被借鉴过来，从而使得方程组完全解耦，求解及存储的成本都极大地降低了。而且数值结果显示，使用集中矩阵使得方法更加稳定。本文将基于有限元法的水平集方法和传统的基于有限差分法的水平集方法进行了对比，并对它们的精度以及精度对结构优化问题的影响进行了讨论。

重新初始化方程同样采用 SDFEM 求解，但是由于边界附近稳定效果不足，需要在方程中增加额外的耗散项。本文讨论了如何选择及调整耗散项的系数。由于数值误差及耗散项的影响，结构边界在重新初始化的过程中会发生漂移。为了限制结构边界的移动，本文采用了施加本质边界条件的方法。在有限元的体系下，该方法可以很自然地实现。

本文还讨论了两种速度扩展方法：自然扩展法及基于偏微分方程的扩展法。其中第二种扩展法同样采用 SDFEM 求解。同时也讨论了和速度扩展有关的“弱材料”法，应力磨平等话题。

本文以最小柔度问题为例，检验了本文所提出的方法在结构拓扑优化中的表现。数值算例既包括规则设计域上的优化问题，又包括非规则域上的优化问题。数值结果表明了本文所提出的方法的可靠性，并且展示了该方法在求解非规则域上的结构优化问题的优势。

Abstract of thesis entitled:

A Finite Element Based Level Set Method for Structural Topology Optimization

Submitted by XING, Xianghua

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in December 2008

A finite element (FE) based level set method is proposed for structural topology optimization problems in this thesis. The level set method has become a popular tool for structural topology optimization in recent years because of its ability to describe smooth structure boundaries and handle topological changes. There are commonly two stages in the optimization process: the stress analysis stage and the boundary evolution stage. The first stage is usually performed with the finite element method (FEM) while the second is often realized by solving the level set equation with the finite difference method (FDM). The first motivation for developing the proposed method is the desire to unify the techniques of both stages within a uniform framework. In addition, there are many problems involving irregular design domains in practice, the FEM is more powerful than the FDM in dealing with these problems. This is the second motivation for this study.

Solving the level set equation with the standard Galerkin FEM might produce unstable results because of the hyperbolic character-

istic of this equation. Therefore, the streamline diffusion finite element method (SDFEM), a stabilized method, is employed to solve the level set equation. In addition to the advantage of simplicity, this method generates a system of equations with a constant, symmetric, and positive defined coefficient matrix. Furthermore, this matrix can be diagonalized by virtue of the lumping technique in structural dynamics. This makes the cost in solving and storing quite low. It is more important that the lumped coefficient matrix may help to improve the stability under some circumstance.

The reinitialization equation is also solved with the SDFEM and an extra diffusion term is added to improve the stability near the boundary. We propose a criterion to select the factor of the diffusion term. Due to numerical errors and the diffusion term, boundary will drift during the process of reinitialization. To constrain the boundary from moving, a Dirichlet boundary condition is enforced. Within the framework of FEM, this enforcement can be conveniently preformed with the Lagrangian multiplier method or the penalty method.

Velocity extension is discussed in this thesis. A natural extension method and a partial differential equation (PDE)-based extension method are introduced. Some related topics, such as the "ersatz" material approach and the recovery of stresses, are discussed as well.

The accuracy of the finite element based level set method (FELSM) is compared with that of the finite difference based level set method (FDLSM). The FELSM is a first-order accurate algorithm but we prove that its accuracy is enough for structural optimization problems considered in this study. Even higher-order accurate FDLSM schemes are used, the numerical results are still the same as those obtained by

FELSM. It is also shown that if the Courant-Friedreichs-Lewy (CFL) number is large, the FELSM is more robust and accurate than FDLSM.

Numerical examples are involved in this thesis to illustrate the reliability of the proposed method. Problems on both regular and irregular design domains are considered and different meshes are tested and compared.

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 and a half 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 have been 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. LUO Zhen, Dr. LIANG Sen, Dr. XIA Hongjian, Dr. WANG Nianfeng, Mr. HO Hon Shan, Mr. LUO Junzhao, Mr. ZHANG Jiwei, Mr. MA Jie, Mr. POON Ming Ki, and Ms. LUI Fung Yee.

This work is dedicated to my wife and my family,
for their support and patience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Structural shape and topology optimization has become an effective design tool for obtaining more efficient structures. The topology optimization, as a conceptual design tool, has been regarded as a powerful structural optimization method because of its ability in producing structures with the highest performance. On the other hand, structural topology optimization has been identified as one of the most challenging tasks in structural design since it is not easy to handle the topological changes in structures. Since the birth of the finite element (FE) based topology optimization at the end of 1980s, various techniques and approaches have been developed during the past decade.

One main approach to structural design for variable topologies is the method of homogenization introduced in [1] (see also [2,3]), in which a material model with micro-scale voids is introduced and the topology optimization problem is solved by seeking the optimal porosity of such a porous medium using one of the optimality criteria. By transforming

the difficult topology design problem into a relatively easier "sizing" problem, the homogenization technique is capable of producing internal holes without prior knowledge of their existence. However, the homogenization method may not yield the intended results for some objectives in the mathematical modeling of structural design. It often produces designs with infinitesimal pores in the materials that make the structure not manufacturable. Furthermore, numerical instabilities may introduce "non-physical" artifacts in the results and make the designs sensitive to variations in the loading.

A number of variations of the homogenization method have been investigated to deal with these issues by penalization of intermediate densities, especially the "solid isotropic material with penalization" (SIMP) approach for its conceptual simplicity [4, 5]. Material properties are assumed constant within each element used to discretize the design domain and the design variables are the element densities. 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, numerical instability and unsmooth structure boundary remain to be the major difficulties for realistic requirements.

A simple method for shape and layout optimization, called "evolutionary structural optimization" (ESO), has been proposed by Xie and co-workers [6, 7], which is based on the concept of gradually removing material 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). This approach is essentially based on an evolutionary strategy focusing on local consequences but not on the global optimum. It is typically computationally expensive. A similar approach called "reverse adaptivity" was proposed by Reynolds et al. [8], in which a fixed percentage of relatively under-stressed material is removed to find approximately fully stressed structures. Essentially, both ESO and reverse adaptivity are homotopy methods based on material hard kills. In reverse adaptivity, finite element meshes near the boundary during the design procedure are refined to reduce computational cost or increase resolution.

Another related approach is called "bubble method" which is proposed by Eschenauer and co-workers [9, 10]. In this method, 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, thus modifying the structural topology in a prescribed manner. In such case, the design for a given topology is settled before its further changes.

All the methods mentioned above focus on material and take material properties as design variables. Differently, the "level set method" is based on boundary variations. In this method, the design variable is actually the exterior and interior boundaries of the structure. Bound-

aries are represented by the level set function and are propagated by the level set equation. Since the level set method can handle the merging and separating of interfaces naturally and flexibly, it offers a tool for simultaneous shape and topology optimization. A short introduction will be given in the next section.

1.2 Level Set Methods for Structural Optimization

The level set method is introduced into the structural optimization field first by Sethian and Wiegmann [11]. In their method, the boundaries are allowed to move according to the stresses on the boundaries. A level set method 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 [12] propose a level set method for frequencies optimization problems. They use functional gradients to calculate the velocity of the level set and deal with optimization problems with geometrical constraints.

In a series of papers [13–16], the theories and algorithms of level set based structural optimization method are developed gradually and the technique is implemented into more general problems. As stated in [15], a boundary-based method with the capability of handling topology 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.

There are some variations of the level set based optimization method, which usually focus on the solution method for the level set equation. In [17,18], radial basis functions (RBFs) are used to discretize the level set function. By means of the method of lines, the authors separate the dependence of the level set function on time and space and transform the partial differential equation (PDE) into a system of ordinary differential equations (ODEs).

RBFs are also employed in [19–22] but in a different manner. Instead of solving the level set equation, this kind of methods parameterize the level set method. The level set equation, which has been discretized using RBF, is substituted into the shape derivative formulation. By virtue of the chain rule, design sensitivities with respect to parameters are derived, and the level set function can be updated by varying parameters according to sensitivities. This method differs from other level set methods in that it needs boundary velocity only and requires boundary integration.

A piecewise constant level set (PCLS) method is implemented to solve the structural optimization problems in [21, 23, 24]. 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.

1.3 Finite Element Based Level Set Methods

The level set equation is a hyperbolic PDE. If the standard Galerkin finite elements are used to solve it, numerical instabilities may arise. There are usually two categories of methods to overcome this difficulty. The first category is to use some stabilized finite element methods which are suitable for hyperbolic or advection dominated equations. The second category changes the level set equation to what can be solved by using the standard Galerkin finite element method. Most of the finite element based level set methods fall in the first category, which are introduced below. For details of the stabilized finite element methods, the reader is referred to [25] and references therein.

Barth and Sethian are the first to discretize the level set equation on unstructured triangular meshes using finite element techniques [26]. They use the stabilized Petrov-Galerkin method to approximate the Hamilton-Jacobi equation. To remove small oscillations sometimes presenting near slope discontinuities, a discontinuity capturing operator [27] is employed. This method is subsequently applied in [28] to treat the growth of cracks. Petrov-Galerkin method is also used, with different formulations, to solve the level set problem in some special applications. For example, the incompressible multiphase flow is simulated in [29,30] and the geodesic contours problem in image processing is handled in [31]. In [31], the Eikonal equation is combined with the level set equation, and both equations are solved simultaneously. No reinitialization is needed in this case. To improve the efficiency, the authors use a banded algorithm which restrict computation to the vicinity of the zero set of the level set function. It is worthwhile to note

that there are some different names of the Petrov-Galerkin method for advection dominated equations in the literature, such as the streamline upwind/Petrov-Galerkin (SUPG) [32], and the streamline diffusion method [33].

The Galerkin least squares (GLS) finite element methods [34], which coincide with SUPG, have also been implemented to solve level set equations [35–37]. In [35], both the level set equation and the velocity extension equation are discretized with GLS and the shock capturing operator [27] is added to prevent numerical oscillations at sharp corners in the interface. Formulations for reinitialization are not proposed in this paper. In [36], the same method is used to solve the level set equation for modeling thermal oxidation of Silicon. The reinitialization is not discussed either.

The discontinuous Galerkin (DG) method is originally developed to provide an approximation exhibiting a better behavior in the presence of discontinuous solutions. A complete review on the method can be found in [38]. This method has also been implemented for level set problems in [39, 40].

The least squares finite element method (LSFEM) [41, 42] is another stable FEM for hyperbolic problems. It is applied in [43, 44] to solve the level set equation on curvilinear coordinates. This FE-based level set technique is subsequently used to optimize the shell structure.

In another kind of stabilized FEMs, temporal discretization precedes the spatial one. The unknown variable is often expanded by Taylor series in time and then the time derivatives are replaced by using the advection equation. This procedure introduces into the equation some additional terms that add the stabilizing diffusion in the streamline di-

rection. The characteristic-Galerkin method and the Taylor-Galerkin method belong to this kind. Although these two methods are developed differently, they are very similar to each other [25]. The former is applied in [45–47] to solve the level set equation; the latter is used in [48, 49].

Next, we introduce the second category in which the hyperbolic level set equation is modified first. The new equation can be solved with the standard Galerkin finite element method. The first method in this category is to add a diffusion term to the level set equation. Then the hyperbolic equation becomes a advection-diffusion equation, which can be solved with the standard Galerkin FEM because the stability is guaranteed by the artificial dissipation. This idea is realized in [50] and the authors couple the level set method with structural topology optimization via the FEMLAB package [51]. It is pointed in [33] that, however, adding artificial diffusion term is not a good choice since this terms usually causes too much dissipations in the crosswind direction.

The second method in the second category is to assume that the level set function is always a signed distance function. Consequently, the advection term disappear and the level set equation becomes a ordinary differential equation. This is method is called the assumed-gradient method in [52] and the same idea is used in [37, 53, 54] too. In this method, one obtains a very simple level set equation at the expense of the task to maintain a signed distance function strictly. Ordinarily, the level set function used in structural optimization is not sensitive to its slope. In most cases, we reinitialize the level set function after several steps. Moreover, it is not necessary to obtain a strict signed distance function. However, in the assumed-gradient method, the level

set function needs to be fully reinitialized before each step. In [52], the reinitialization requires locating the closest-point projection of each node onto the interface. In [53,54], a geometric reinitialization scheme is proposed, which also needs to calculate the closest distance from each node to the interface. It is interesting that, in [37], the reinitialization is realized by solving the Eikonal equation with GLS and Newton-Raphson iteration.

In [43,44,50,53,54], the finite element based level set methods have been applied to structural optimization.

1.4 Contributions and Organization of this Dissertation

The advantages of the level set method for structural optimization have been extensively discussed in literature. In most of the applications, the level set method is implemented with the finite difference method (FDM). This method works well on a structured grid, but difficulties happen if the problem involves complex geometries and boundaries, where spatial discretization with the structured grid is impossible. However, the finite element method (FEM) handles these problems flexibly. This is one of our motivations for implementing the level set method with the FEM. The second motivation is related to the procedure of the structural optimization. There are generally two stages in a level set based structural optimization procedure: the stress analysis stage and the boundary evolution stage involving level set methods. The first one is typically carried out with FEMs as often in industrial applications. Therefore, our aim is to unify the techniques of both

stages within a uniform framework.

In this dissertation, a finite element based level set method is introduced for structural topology optimization. The streamline diffusion finite element method (SDFEM) is used to solve the level set equation and the reinitialization equation, and this SDFEM-based level set technique is combined with the structural optimization in the first time. The reason that we employ SDFEM for level set methods may be summarized as follows:

1. This method is relatively simple compared with other stabilized FEMs.

2. In this method, the coefficient matrix of the discretized level set equation is symmetric and positive definite. Moreover, we point out that this matrix is similar to the mass matrix in structural dynamics. Therefore, the mass lumping technique is borrowed. Numerical results show that using the lumped coefficient matrix improves efficiency significantly.

We have also discussed the accuracy of the proposed method and compared it with the finite difference method (FDM) commonly used in conventional level set methods. The presented method possesses the same order of accuracy as ENO1, the first-order accurate upwinding FDM. Although there are some higher-order schemes in FDM, we show that, in this study, the accuracy of the presented method is enough for structural optimization problems.

While the reinitialization equation is solved, numerical errors or added diffusion term will cause the boundary to move. We use the Lagrangian multiplier method or the penalty method to fix the bound-

ary. It turns out that this is a natural procedure within the FEM framework.

Since the performance of the level set method depends highly on the velocity field, the velocity extension aiming at structural optimization problems is discussed. Some related issues, such as the influence of stresses singularities and stresses smoothing, are also discussed.

This dissertation is organized as follows. In Chapter 2, the background knowledge of the structural optimization and the level set method are introduced. Some algorithms for level set based structural optimization are discussed. In what follows we present the finite element based level set method in Chapter 3. Formulations are derived in details and parameters are defined explicitly. Some test cases demonstrate the performance of the proposed method. The velocity extension is discussed in Chapter 4, where two methods are introduced and compared. Chapter 5 exhibits numerical examples including problems in regular and irregular domain, with structured and free mesh. Results illustrate the feasibility of the presented method. Conclusions and future work are discussed in the last chapter.

□ **End of chapter.**

Chapter 2

Level Set Methods For Structural Optimization

2.1 Structural Optimization Problems

There are several kinds of structural optimization problems, such as the minimum mean compliance problem, the maximum natural frequency problem, and the minimum stress problem. In this chapter, the first one is taken for example. In general, the minimum mean compliance problem can be specified as:

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

with

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

where Ω is the domain occupied by a structure and $\mathbf{u}$ is the displacement of the structure under some loads and boundary conditions, $\boldsymbol{\varepsilon}$ the strains and $\mathbf{D}$ the elasticity matrix. J is called the objective function and the integrand $F(\mathbf{u})$ 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 $\mathbf{u}$ which is the solution of the following state equations defined on Ω:

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

where $\boldsymbol{\sigma}$ is the stresses, $\mathbf{f}$ the body force, $\mathbf{g}$ the specified traction acting on Neumann boundary Γ_N. The boundary of structure is denoted by $\partial\Omega$ and following relationships are satisfied:

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

where Γ_D is called the Dirichlet boundary and Γ_f means the traction free boundary.

Eq. (2.3) can be expressed as the following weak form: find $\mathbf{u} \in U$ such that

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

with

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

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

where U is the space of kinematically admissible functions.

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

2.1.1 Shape Design Sensitivity Analysis

In the shape and topology optimization based on boundary variation, the shape of the domain Ω is treated as the design variable. The material derivative idea of continuum mechanics and the adjoint variable method of design sensitivity analysis are applied to obtain a computable shape derivative.

We introduce a mapping T which is defined only by one parameter t, then

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

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

$$V = \frac{dx}{dt}. \tag{2.9}$$

Suppose that z is a component of the solution to Eq. (2.3) defined in Ω. Then the material derivative of z is defined as

$$\dot{z} = z' + \nabla z \cdot \mathbf{V}, \tag{2.10}$$

where z' is the partial derivative of z with respect to t and $\mathbf{V}$ is the velocity vector:

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

or in three-dimensional form,

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

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.,

$$\left(\frac{\partial z}{\partial x}\right)' = \left(\frac{\partial z'}{\partial x}\right) \tag{2.11}$$

Next, two Lemmas are given to show how to compute the material derivative of integrals in which both the integrands and the domains of integration depend on the parameter t.

Lemma 1 *Let I_1 be a domain integral over Ω*

$$I_1 = \int_\Omega f(\mathbf{u}) d\Omega$$

where f is a regular function defined on Ω. The material derivative of I_1 is

$$\dot{I}_1 = \int_\Omega f'(\mathbf{u}) d\Omega + \int_{\partial\Omega} f(\mathbf{u}) V_n d\Gamma$$

where V_n is the normal velocity on boundary $\partial\Omega$.

Lemma 2 *Consider a boundary integral over Γ,*

$$I_2 = \int_\Gamma g(\mathbf{u}) d\Gamma$$

where g is a regular function defined on Γ. The material derivative of I_2 is

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

where $\mathbf{n}$ 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 function J:

$$\dot{J} = \int_\Omega F'(\mathbf{u}) d\Omega + \int_{\partial\Omega} F(\mathbf{u}) V_n d\Gamma \tag{2.12}$$

where

$$F'(\mathbf{u}) = \frac{\partial F}{\partial \mathbf{u}} \cdot \mathbf{u}' \tag{2.13}$$

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

We first take the material derivative of both sides of Eq. (2.4) using Lemma 1 and Lemma 2,

$$\dot{a}(\mathbf{u},\mathbf{v}) = \int_\Omega c'(\mathbf{u},\mathbf{v})d\Omega + \int_{\partial\Omega} c(\mathbf{u},\mathbf{v})V_n d\Gamma \tag{2.14}$$

$$\begin{aligned}\dot{l}(\mathbf{v}) = &\int_\Omega \mathbf{f}\cdot\mathbf{v}'d\Omega + \int_{\partial\Omega}\mathbf{f}\cdot\mathbf{v}V_n d\Gamma + \int_{\Gamma_N}\mathbf{g}\cdot\mathbf{v}'d\Gamma \\ &+ \int_{\Gamma_N}\big(\nabla(\mathbf{g}\cdot\mathbf{v})\cdot\mathbf{n} + \kappa\mathbf{g}\cdot\mathbf{v}\big)V_n d\Gamma\end{aligned} \tag{2.15}$$

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

Consider the integrand of the first integral on the right hand side of Eq. (2.14):

$$\begin{aligned}c'(\mathbf{u},\mathbf{v}) &= \big(\boldsymbol{\varepsilon}(\mathbf{u})^T\mathbf{D}\boldsymbol{\varepsilon}(\mathbf{v})\big)' \\ &= \big(D_{ijkl}\varepsilon_{ij}(\mathbf{u})\varepsilon_{kl}(\mathbf{v})\big)' \\ &= D_{ijkl}\big(\varepsilon'_{ij}(\mathbf{u})\varepsilon_{kl}(\mathbf{v}) + \varepsilon_{ij}(\mathbf{u})\varepsilon'_{kl}(\mathbf{v})\big)\end{aligned} \tag{2.16}$$

According to Eq. (2.11), the shape derivative of ε can be transformed as follows:

$$\begin{aligned}\varepsilon'_{ij}(\mathbf{u}) &= \frac{1}{2}\left(\left(\frac{\partial u_i}{\partial x_j}\right)' + \left(\frac{\partial u_j}{\partial x_i}\right)'\right) \\ &= \frac{1}{2}\left(\left(\frac{\partial u'_i}{\partial x_j}\right) + \left(\frac{\partial u'_j}{\partial x_i}\right)\right) \\ &= \varepsilon_{ij}(\mathbf{u}')\end{aligned} \tag{2.17}$$

and similarly

$$\varepsilon'_{kl}(\mathbf{v}) = \varepsilon_{kl}(\mathbf{v}') \tag{2.18}$$

Substituting Eq. (2.17) and (2.18) into Eq. (2.16) produces

$$\begin{aligned} c'(\mathbf{u},\mathbf{v}) &= D_{ijkl}\big(\varepsilon_{ij}(\mathbf{u}')\varepsilon_{kl}(\mathbf{v}) + \varepsilon_{ij}(\mathbf{u})\varepsilon_{kl}(\mathbf{v}')\big) \\ &= c(\mathbf{u}',\mathbf{v}) + c(\mathbf{u},\mathbf{v}'), \end{aligned} \tag{2.19}$$

so the material derivative of the bilinear form becomes

$$\dot{a}(\mathbf{u},\mathbf{v}) = \int_\Omega c(\mathbf{u}',\mathbf{v})d\Omega + \int_\Omega c(\mathbf{u},\mathbf{v}')d\Omega + \int_{\partial\Omega} c(\mathbf{u},\mathbf{v})V_n d\Gamma\,. \tag{2.20}$$

From Eq. (2.20) and Eq. (2.15), we have

$$\begin{aligned} &\int_\Omega c(\mathbf{u}',\mathbf{v})d\Omega + \int_\Omega c(\mathbf{u},\mathbf{v}')d\Omega + \int_{\partial\Omega} c(\mathbf{u},\mathbf{v})V_n d\Gamma \\ &= \int_\Omega \mathbf{f}\cdot\mathbf{v}'d\Omega + \int_{\partial\Omega} \mathbf{f}\cdot\mathbf{v}V_n d\Gamma \\ &\quad + \int_{\Gamma_N} \mathbf{g}\cdot\mathbf{v}'d\Gamma + \int_{\Gamma_N} \big(\nabla(\mathbf{g}\cdot\mathbf{v})\cdot\mathbf{n} + \kappa\mathbf{g}\cdot\mathbf{v}\big)V_n d\Gamma \end{aligned} \tag{2.21}$$

Note that

$$\int_\Omega c(\mathbf{u},\mathbf{v}')d\Omega = \int_\Omega \mathbf{f}\cdot\mathbf{v}'d\Omega + \int_{\Gamma_N} \mathbf{g}\cdot\mathbf{v}'d\Gamma \tag{2.22}$$

which is the variational identity. Hence these terms can be canceled out from Eq. (2.21) and the following equation is obtained:

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

Next, we construct the adjoint equation:

$$a(\mathbf{u}_a,\mathbf{v}_a) = \int_\Omega \frac{\partial F}{\partial \mathbf{u}}\cdot\mathbf{v}_a d\Omega, \quad \forall\,\mathbf{v}_a \in U\,, \tag{2.24}$$

where $a(\cdot,\cdot)$ is the bilinear form defined in Eq. (2.5) and the subscript a means "adjoint" variables. Since the test function $\mathbf{v}_a$ can be selected arbitrarily as long as it belongs to U, we can replace it with $\mathbf{u}'$ ($\mathbf{u}' \in U$) and change Eq. (2.24) into

$$a(\mathbf{u}_a, \mathbf{u}') = \int_\Omega \frac{\partial F}{\partial \mathbf{u}} \cdot \mathbf{u}' d\Omega \,. \tag{2.25}$$

Similarly, we can replace $\mathbf{v}$ in Eq. (2.23) with $\mathbf{u}_a$:

$$\begin{aligned} a(\mathbf{u}', \mathbf{u}_a) = &\int_{\partial\Omega} \big(\mathbf{f} \cdot \mathbf{u}_a - c(\mathbf{u}, \mathbf{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\mathbf{g} \cdot \mathbf{u}_a) \cdot \mathbf{n} + \kappa \mathbf{g} \cdot \mathbf{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.26}$$

Comparing Eq. (2.26), (2.25), and (2.12) and noting that the bilinear functional is symmetric, we can eliminate all the terms related to $\mathbf{u}'$ from Eq. (2.12):

$$\begin{aligned} \dot{J} = &\int_{\partial\Omega} \big(F(\mathbf{u}) + \mathbf{f} \cdot \mathbf{u}_a - c(\mathbf{u}, \mathbf{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\mathbf{g} \cdot \mathbf{u}_a) \cdot \mathbf{n} + \kappa \mathbf{g} \cdot \mathbf{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.27}$$

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

$$\begin{aligned} \dot{J} = &\int_{\Gamma_N \cup \Gamma_f} \big(F(\mathbf{u}) + \mathbf{f} \cdot \mathbf{u}_a - c(\mathbf{u}, \mathbf{u}_a)\big) V_n d\Gamma \\ &+ \int_{\Gamma_N} \big(\nabla(\mathbf{g} \cdot \mathbf{u}_a) \cdot \mathbf{n} + \kappa \mathbf{g} \cdot \mathbf{u}_a\big) V_n d\Gamma \,. \end{aligned} \tag{2.28}$$

Substituting Eq. (2.2) into the adjoint equation Eq. (2.24), we obtain

$$a(\mathbf{u}_a, \mathbf{v}_a) = a(\mathbf{u}, \mathbf{v}_a), \quad \forall\, \mathbf{v}_a \in U, \tag{2.29}$$

which means that the adjoint variable $\mathbf{u}_a$ is same as the state variable $\mathbf{u}$. Therefore, the minimum mean compliance problem is a self-adjoint

problem and the shape derivative reduces to

$$\begin{aligned}\dot{J} = &\int_{\Gamma_N \cup \Gamma_f} \left(\mathbf{f} \cdot \mathbf{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u})\right) V_n d\Gamma \\ &+ \int_{\Gamma_N} \left(\nabla(\mathbf{g} \cdot \mathbf{u}) \cdot \mathbf{n} + \kappa \mathbf{g} \cdot \mathbf{u}\right) V_n d\Gamma \,.\end{aligned} \tag{2.30}$$

2.1.2 Volume Constraint

In practical problems, a volume constraint is always applied:

$$\int_\Omega d\Omega \leq vol, \tag{2.31}$$

which describes an upper limit on the amount of material in terms of the maximum admissible volume *vol* of the structure. One can combine it with the objective function using the augmented Lagrangian method [55] to construct an augmented objective function:

$$\bar{J} = J + \lambda\Big(\int_\Omega d\Omega - vol\Big) + \frac{r}{2}\Big(\int_\Omega d\Omega - vol\Big)^2, \tag{2.32}$$

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

Similar as in section 2.1.1, the material derivative of $\bar{J}$ is:

$$\dot{\bar{J}} = \int_{\Gamma_f} \Big(\mathbf{f} \cdot \mathbf{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}) + \bar{\lambda}\Big) V_n \, d\Gamma \tag{2.33}$$

where

$$\bar{\lambda} = \max\Big\{0,\ \lambda + r\Big(\int_\Omega d\Omega - vol\Big)\Big\}. \tag{2.34}$$

It has been assumed that Γ_N is fixed while Eq. (2.33) is being derived, so the integral over Γ_N vanishes. This assumption will be used in this study, unless otherwise specified.

It is proved in [56] that along with the design variable converges 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.

2.1.3 Optimization Algorithm

To guarantee the reduction in $\bar{J}$, we require the boundary to move under the velocity V_n that satisfies the descent property:

$$\dot{\bar{J}} < 0. \tag{2.35}$$

A simple descent method is the steepest descent method in which

$$V_n = -\Big(\mathbf{f} \cdot \mathbf{u} - \frac{1}{2}\boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}) + \bar{\lambda}\Big). \tag{2.36}$$

Substituting Eq. (2.36) into (2.33), we can see that the descent property is satisfied.

The boundary velocity is not only the important result of the shape design sensitivity analysis but also the critical link between the structural optimization and the level set method, which will be introduced in the next section.

It is worthwhile to note that in practice, minimization of volume (weight) subject to compliance constraint is more popular than minimization of compliance under volume constraint. As has been proved in [57], these two problems are equivalent to each other. If one can solve one of them, the other one is just the dual problem. We consider the later problem in this thesis because enforcing the volume constraint is much easier than enforcing the compliance one.

2.2 Level Set Methods

The level set method, first introduced in [58], has become a powerful tool for computing and analyzing the motion of an interface in two or three dimensions. It has been applied in many fields, such as image processing, solids modeling, fluid mechanics, and combustion [59, 60]. The fundamental concept of level set methods is described here to provide necessary background for later parts.

2.2.1 Implicit Interface Representations

In the level set framework, an interface Γ (curve or surface) is represented implicitly through a level set function $\phi(\mathbf{x})$, which is Lipschitz-continuous, and the interface itself is the zero isocontour or the zero level set. Mathematically, it can be stated as

$$\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0, \mathbf{x} \in D\} \tag{2.37}$$

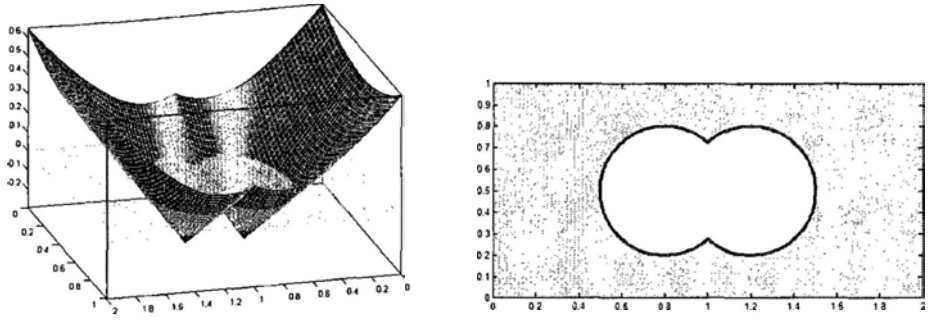
where D is a domain that contains Γ completely. To describe a structure Ω, we give the following definition:

$$\begin{cases} \phi(\mathbf{x}) > 0 & \mathbf{x} \in \Omega \\ \phi(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \\ \phi(\mathbf{x}) < 0 & \mathbf{x} \in D \backslash \bar{\Omega} \end{cases} \tag{2.38}$$

where $\bar{\Omega} = \Omega \cup \partial\Omega$. Figure (2.1) shows a implicit representation through a level set function.

Implicit representations include some very powerful geometric tools. For example, the unit outward normal on the interface can be defined as:

$$\mathbf{n} = -\frac{\nabla\phi}{|\nabla\phi|} \tag{2.39}$$



(a) (b)

Figure 2.1: Implicit interface: (a) level set function ϕ and (b) interface and partition of domain

Note that the minus sign guarantees that $\mathbf{n}$ points from the interior to the exterior. From now on, I shall use *outward normal* and *normal* interchangeably unless otherwise noted. The mean curvature of the interface is defined as the divergence of the normal $\mathbf{n}$:

$$\kappa = \nabla \cdot \left(-\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{2.40}$$

Actually, since the level set function defines the interface in a domain of one higher-dimension, instead of defining $\mathbf{n}$ and κ on the interface only, we can use Eq. (2.39) and (2.40) to define them everywhere on the domain.

2.2.2 Level Set Equation

Level set methods add dynamics to implicit interfaces. The implicit function ϕ is used both to represent the interface and to evolve the interface. If we take derivative of the following equation:

$$\phi(\mathbf{x}, t) = 0 \qquad \mathbf{x} \in \partial\Omega \tag{2.41}$$

we get a advection (or convection) equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0 \tag{2.42}$$

which is referred to as the level set equation. If the motion of the interface is only in the normal direction, as in the structural shape optimization, the velocity becomes $\mathbf{V} = V_n \mathbf{n}$. Substituting it into Eq. (2.42) and using Eq. (2.39), we obtain the level set equation for motion in the normal direction:

$$\frac{\partial \phi}{\partial t} - V_n |\nabla \phi| = 0 \tag{2.43}$$

This partial differential equation defines the motion of the interface where $\phi(\mathbf{x}) = 0$ 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. And in addition, with implicit representation boolean operations on the boundaries are easy to implement.

It should be noted that in some literature, the definition of interior and exterior of the structure in Eq. (2.38) is reversed:

$$\begin{cases} \phi(\mathbf{x}) < 0 & \mathbf{x} \in \Omega \\ \phi(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \\ \phi(\mathbf{x}) > 0 & \mathbf{x} \in D \backslash \bar{\Omega} \end{cases} \tag{2.44}$$

In this case, the normal $\mathbf{n}$ becomes

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$$

and then Eq. (2.43) reads

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

2.2.3 Reinitialization

The level set function of a given interface is not unique—if ϕ is the level set function of Γ, then $\alpha\phi$ ($\alpha \neq 0$) defines the same interface Γ. Therefore, one can choose the best level set function according to some criteria. In general, the criteria concern the stability and accuracy of solving the level set equation and the accuracy of extracting the interface from the level set function. In most applications, analytical solution of the level set equation does not exist and the level set function ϕ has to be replaced by an approximated function ϕ^h and the PDE must be solved numerically. In this case, in addition to smoothness of ϕ^h, one may desire that the level set function is neither too steep nor too flat, especially near the interface. According to these, the signed distance function should be a good choice [60].

A signed distance function is an implicit function ϕ with

$$\phi(\mathbf{x}) = \begin{cases} \mathbf{d} & \mathbf{x} \in \Omega \\ 0 & \mathbf{x} \in \partial\Omega \\ -\mathbf{d} & \mathbf{x} \in D \backslash \bar{\Omega} \end{cases} \tag{2.45}$$

where

$$\mathbf{d} = |\mathbf{x} - \mathbf{x}_C| \tag{2.46}$$

with $\mathbf{x}_C$ the closest point on the boundary to $\mathbf{x}$.

As the interface evolves, ϕ will generally drift away from the feature as a signed distance function. In other words, it will becomes too steep

or flat. Thus, one always needs to re-construct the signed distance function during the evolution, which is called "reinitialization".

There are various techniques for reinitialization [59,60]. In this section, the method of solving the reinitialization equation is introduced. The reinitialization equation [61] is

$$\frac{\partial \phi}{\partial t} = S(\phi_0)(1 - |\nabla \phi|) \tag{2.47}$$

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

$$\frac{\partial \phi}{\partial t} = 0, \tag{2.48}$$

the signed distance function is rebuilt since

$$|\nabla \phi| = 1. \tag{2.49}$$

Equation (2.47) can be written in the following form

$$\frac{\partial \phi}{\partial t} + \mathbf{w} \cdot \nabla \phi = S(\phi_0), \tag{2.50}$$

where

$$\mathbf{w} = S(\phi_0)(\nabla \phi / |\nabla \phi|). \tag{2.51}$$

For numerical purposes it is useful to smooth the sign function. In [61], S is smoothed as

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \alpha^2}} \tag{2.52}$$

where α is a small constant, which can be specified as the mesh size. Peng et al. [62] suggest that

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \alpha^2 |\nabla \phi|^2}} \tag{2.53}$$

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.

2.2.4 Numerical Discretization

A simple first-order accurate method for the time discretization of Eq. (2.43) is the forward Euler method:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} - V_n^n |\nabla \phi^n| = 0 \tag{2.54}$$

where V_n^n is the given velocity at time t^n and $\nabla \phi^n$ is the gradient of ϕ at time t^n.

The upwind differencing or upwinding is used to calculate the spatial derivatives of ϕ. We describe it in one dimension.

$$\frac{\partial \phi}{\partial x} = \begin{cases} \phi_x^- = \dfrac{\phi_i - \phi_{i-1}}{\triangle x} & \text{if } V_n > 0 \\ \phi_x^+ = \dfrac{\phi_{i+1} - \phi_i}{\triangle x} & \text{if } V_n < 0 \end{cases} \tag{2.55}$$

This is a first-order accurate discretization of the spatial operator. To improve the accuracy, the idea of essentially non-oscillatory (ENO) introduced in [63] has been used [61, 64, 65]. In ENO, the velocity is still used to decide whether ϕ_x^- or ϕ_x^+ is used, but the approximation for ϕ_x^- or ϕ_x^+ 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] improve the WENO method to obtain the optimal fifth-order accuracy in smooth regions of the flow. In [68], Jiang and Peng extend WENO to the Hamilton-Jacobi framework.

2.3 Structural Optimization With Level Set Methods

As has been mentioned, the boundary velocity is the link between the structural shape and topology optimization and the level set method. Once the stress analysis has been performed and the velocity has been calculated based on stresses and strains, one can evolve the boundary by solving the level set equation. Then a new boundary and therefore a new 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 [13–16].

In addition, there are some level set structural optimization methods based on the radial basis function (RBF) techniques. These methods can be divided into two categories. In the first, the procedure is similar to what we have discussed above except that the level set function is approximated with RBF and the Hamilton-Jacobi PDE is transformed into a system of ODEs. This category is called the RBF level set optimization method and is introduced in (2.3.1). In the second category, the velocity is eliminated and the level set equation is embedded into the shape derivative formulation. Then the sensitivities of the objective function with respect to the parameters are derived and the parameters are updated with a proper optimization algorithm. This category is called the parametric level set optimization method and is described in (2.3.2).

2.3.1 RBF Level Set Optimization Method

In this method, the level set function is approximated with radial basis functions:

$$\phi(\mathbf{x},t)=\sum_{i=1}^{N}\varphi_i(\mathbf{x})\alpha_i(t)=\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\alpha}(t) \tag{2.56}$$

where $\varphi_i(\mathbf{x})$ is the well defined radial basis function on knot i and $\alpha_i(t)$ is the related unknown coefficient, which is usually called *expansion coefficient*. Substituting Eq. (2.56) into Eq. (2.43), we get

$$\boldsymbol{\varphi}^{\mathrm{T}}\frac{d\boldsymbol{\alpha}}{dt}-V_n|\nabla\boldsymbol{\varphi}^{\mathrm{T}}\boldsymbol{\alpha}|=0 \tag{2.57}$$

and the PDE problem becomes a initial value problem. To determine N unknown coefficients, one can use the collocation method and obtain a system of ODEs as follows:

$$\mathbf{H}\frac{d\boldsymbol{\alpha}}{dt}=\mathbf{B}(\boldsymbol{\alpha}) \tag{2.58}$$

where

$$\mathbf{H}=\begin{bmatrix}\varphi_1(\mathbf{x}_1) & \cdots & \varphi_N(\mathbf{x}_1)\\ \vdots & \ddots & \vdots\\ \varphi_1(\mathbf{x}_N) & \cdots & \varphi_N(\mathbf{x}_N)\end{bmatrix} \tag{2.59}$$

$$\mathbf{B}(\boldsymbol{\alpha})=\begin{pmatrix}V_n(\mathbf{x}_1)|\nabla\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_1)\boldsymbol{\alpha}|\\ \vdots\\ V_n(\mathbf{x}_N)|\nabla\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_N)\boldsymbol{\alpha}|\end{pmatrix} \tag{2.60}$$

Using forward Euler method, an approximate solution to Eq. (2.58) can be given by

$$\boldsymbol{\alpha}(t_{n+1})=\boldsymbol{\alpha}(t_n)+\Delta t\mathbf{H}^{-1}\mathbf{B}(\boldsymbol{\alpha}(t_n)) \tag{2.61}$$

For details, please refer to [17,18].

2.3.2 Parametric Level Set Optimization Method

In this method, the level set function is approximated in the same manner as in Eq. (2.56). From Eq. (2.57), we obtain

$$V_n = \frac{1}{|\nabla \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{\alpha}|} \boldsymbol{\varphi}^{\mathrm{T}} \frac{d\boldsymbol{\alpha}}{dt} = \frac{1}{|\nabla \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{\alpha}|} \sum_{i=1}^{N} \varphi_i(\mathbf{x}) \dot{\alpha}_i \tag{2.62}$$

Substituting Eq. (2.62) into the shape derivative formulation (2.33), we can eliminate the velocity and get

$$\dot{\bar{J}} = \int_{\Gamma_f} (G + \bar{\lambda}) \frac{1}{|\nabla \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{\alpha}|} \sum_{i=1}^{N} \varphi_i(\mathbf{x}) \dot{\alpha}_i \, d\Gamma \tag{2.63}$$

with

$$G = \mathbf{f} \cdot \mathbf{u} - \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}) \tag{2.64}$$

Note that the summation and integration can be interchanged and $\dot{\alpha}_i$ can be taken out from the integral since it is independent of the spatial coordinates. Then we get

$$\dot{\bar{J}} = \sum_{i=1}^{N} \dot{\alpha}_i \int_{\Gamma_f} \left(G + \bar{\lambda} \right) \frac{\varphi_i(\mathbf{x})}{|\nabla \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{\alpha}|} \, d\Gamma \tag{2.65}$$

Because $\bar{J}$ depends on ϕ, i.e.

$$\bar{J} = \bar{J}(\phi) = \bar{J}\Big(\sum_{i=1}^{N} \varphi_i(\mathbf{x}) \alpha_i(t) \Big), \tag{2.66}$$

we can take the derivative by the chain rule

$$\dot{\bar{J}} = \sum_{i=1}^{N} \frac{\partial \bar{J}}{\partial \alpha_i} \dot{\alpha}_i \tag{2.67}$$

and define

$$\bar{J}_{,\alpha_i} = \frac{\partial \bar{J}}{\partial \alpha_i} \tag{2.68}$$

as the sensitivity of $\bar{J}$ with respect to parameter α_i.

Comparing Eq. (2.63) and (2.67) and noting that all α_i are independent, we get

$$\bar{J}_{,\alpha_i} = \int_{\Gamma_f} \left(G + \bar{\lambda}\right) \frac{\varphi_i(\mathbf{x})}{|\nabla \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{\alpha}|} \, d\Gamma \qquad i = 1, \cdots, N \tag{2.69}$$

Therefore, the sensitivities are evaluated as a boundary integral.

An advantage of the parametric level set optimization method is that one can use not only the simplest steepest descent method but also many other advanced optimization algorithms, e.g. mathematical programming, optimal criteria (OC), and method of moving asympote (MMA). For details, the reader is referred to [19–22].

□ **End of chapter.**

Chapter 3

Finite Element Based Level Set Method

3.1 Discretization of the Level Set Equation

3.1.1 Formulation

The level set equation (2.42) has the following weak formulation: to find $\phi(\mathbf{x}, t) \in H^1$ such that

$$\left(\frac{\partial \phi}{\partial t},\ v\right) + (\mathbf{V} \cdot \nabla \phi,\ v) = 0 \quad \forall v \in H^1 \tag{3.1}$$

where

$$(f_1,\ f_2) = \int_D f_1 f_2 d\Omega$$

However, this standard Galerkin formulation gives rise to central difference type approximations of differential operator, and therefore produces spurious oscillations when advection dominated equations are considered. In order to get stable numerical solutions, the streamline diffusion (SD) method [33] is used, with which Eq. (3.1) becomes the

follows: Find $\phi(\mathbf{x},t) \in H^1 \times T_n$, $T_n = [t_n, t_{n+1}]$, such that

$$\int_{T_n} \big(L(\phi),\ v + \beta L(v)\big)dt + \big([\phi^n],\ v^{n+}\big) = 0 \tag{3.2}$$

where

$$L(\phi) = \frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi \tag{3.3}$$

$$L(v) = \frac{\partial v}{\partial t} + \mathbf{V} \cdot \nabla v \tag{3.4}$$

$$[\phi^n] = \phi^{n+} - \phi^{n-} \tag{3.5}$$

$$\phi^{n+} = \lim_{s \to 0} \phi(\mathbf{x}, t_n + s) \tag{3.6}$$

$$\phi^{n-} = \lim_{s \to 0} \phi(\mathbf{x}, t_n - s) \tag{3.7}$$

and β is a parameter which will be discussed later. Here, the trial function ϕ and the test function v are piecewise constant in time and $[\phi^n]$ is the jump of ϕ at time t_n. Therefore, we can state that

$$\frac{\partial \phi}{\partial t} = 0 \quad \text{and} \quad \frac{\partial v}{\partial t} = 0 \qquad \text{on } T_n \tag{3.8}$$

and write Eq. (3.2) as

$$\Delta t\big(\mathbf{V} \cdot \nabla \phi,\ v + \beta \mathbf{V} \cdot \nabla v\big) + \big([\phi^n],\ v^{n+}\big) = 0 \tag{3.9}$$

where $\Delta t = t_{n+1} - t_n$ is the length of T_n. Substituting Eq. (3.5) into Eq. (3.9) and designating ϕ^{n-} and ϕ^{n+} as ϕ^n and ϕ^{n+1}, we obtain the final weak formulation: Given $\phi^n \in H^1$, find $\phi^{n+1} \in H^1$ such that

$$(\phi^{n+1} - \phi^n,\ v) + \Delta t\big(\mathbf{V} \cdot \nabla \phi,\ v + \beta \mathbf{V} \cdot \nabla v\big) = 0 \quad \forall v \in H^1 \tag{3.10}$$

Similarly, for Eq. (2.43) describing the motion in normal direction, the weak formulation reads: Assume that $\phi^n = \phi(\mathbf{x}, t_n)$ and V_n are given, find $\phi^{n+1} \in H^1$, such that

$$(\phi^{n+1} - \phi^n, v) = \Delta t(V_n |\nabla \phi^n|, \bar{v}) \qquad \forall v \in H^1 \tag{3.11}$$

where

$$\bar{v} = v + \beta_1 \mathbf{V} \cdot \nabla v \tag{3.12}$$

and $\mathbf{V}$ is $V_n\mathbf{n}$. If we let $\delta\phi = \phi^{n+1} - \phi^n$, Eq. (3.11) can be written as: find $\delta\phi \in H^1$, such that

$$(\delta\phi, v) = \Delta t(V_n|\nabla\phi^n|, \bar{v}) \qquad \forall v \in H^1 \tag{3.13}$$

The parameter β_1 is chosen as [30]

$$\beta_1 = \frac{1}{2\sqrt{(\Delta t)^{-2} + |\mathbf{J}^{-1}\mathbf{V}|^2}} \tag{3.14}$$

where $\mathbf{J}$ is the Jacobian matrix for the transformation from the global coordinates to the local coordinates. In two dimensions, it is defined as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{3.15}$$

with (x, y) meaning the global coordinates and (ξ, η) the local coordinates.

The whole design domain is discretized with a finite element mesh and the level set function ϕ is approximated by

$$\phi(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\phi_i \tag{3.16}$$

where n is the number of nodes, $N_i(\mathbf{x})$ are shape functions, and ϕ_i are nodal value of the level set function. Equation (3.16) can also be written in the vector form:

$$\phi(\mathbf{x}) = \mathbf{N}\mathbf{\Phi} \tag{3.17}$$

with

$$\mathbf{N} = [N_1(\mathbf{x}),\ N_2(\mathbf{x}),\ \cdots,\ N_n(\mathbf{x})] \tag{3.18}$$

$$\mathbf{\Phi} = [\phi_1,\ \phi_2,\ \cdots,\ \phi_n]^{\mathrm{T}} \tag{3.19}$$

Similarly, the increment $\delta\phi$ and the test function v are approximated as

$$\delta\phi(\mathbf{x}) = \mathbf{N}\mathbf{d} \tag{3.20}$$

$$v(\mathbf{x}) = \mathbf{N}\mathbf{v} \tag{3.21}$$

with

$$\mathbf{d} = [\delta\phi_1,\ \delta\phi_2,\ \cdots,\ \delta\phi_n]^{\mathrm{T}}$$
$$\mathbf{v} = [v_1,\ v_2,\ \cdots,\ v_n]^{\mathrm{T}}.$$

After substituting Eq. (3.17), (3.20), and (3.21) into (3.13), we obtain the following finite element equations:

$$\mathbf{A}\mathbf{d} = \mathbf{b} \tag{3.22}$$

where the coefficient matrix is

$$\mathbf{A} = \int_D \mathbf{N}^{\mathrm{T}}\mathbf{N}\, d\Omega \tag{3.23}$$

and the right hand side vector is

$$\mathbf{b} = \Delta t \int_D \mathbf{W}^{\mathrm{T}} V_n |\nabla\phi^n|\, d\Omega \tag{3.24}$$

with

$$\mathbf{W} = \mathbf{N} + \beta_1 \mathbf{V}^{\mathrm{T}}\nabla\mathbf{N} \tag{3.25}$$

$$\nabla\mathbf{N} = [\nabla N_1(\mathbf{x}),\ \nabla N_2(\mathbf{x}),\ \cdots,\ \nabla N_n(\mathbf{x})] \tag{3.26}$$

Once the unknown vector $\mathbf{d}$ is obtained, we can update the nodal values of the level set function using:

$$\mathbf{\Phi}^{n+1} = \mathbf{\Phi}^n + \mathbf{d} \tag{3.27}$$

The temporal discretization is the forward Euler scheme, so the time step size Δt is restricted by the CFL condition:

$$\Delta t = \alpha \frac{h}{\max\{V_n\}}, \quad \alpha \in (0, 1) \tag{3.28}$$

where h is the size of element. In this study, $\alpha = 0.5$ is used.

3.1.2 Discussions And Comparison With Upwind Differencing

The streamline diffusion method uses a modified weighting function for the advection term. The modified weighting function is constructed by adding a perturbation term, which acts only in the streamline direction (or, in other words, the direction of velocity), to the standard Galerkin weighting function. In this way, the the streamline diffusion finite element method introduces diffusion effect in the streamline direction, and hence is more stable than the Galerkin finite element method which is actually quite *under-diffused* for hyperbolic equations [69].

Unlike other classical artificial diffusion methods, the streamline diffusion method does not introduce diffusion in the direction perpendicular to the streamlines. Therefore, it circumvents the problem of excessive diffusion in crosswind direction and is more accurate than artificial diffusion methods.

Furthermore, the SDFEM and upwind differencing have some characters in common. To show this, we consider a simple one-dimensional case, where the advection term in level set equation (2.42) reduces to a linear, first-order partial derivative term $V\partial\phi/\partial x$, and the SDFEM formulation of this term is

$$V(\frac{\partial \phi}{\partial x}, v + \beta V \frac{\partial v}{\partial x}) \tag{3.29}$$

If the one-dimensional domain is meshed with elements, ϕ is approximated as

$$\phi(x) = \sum_{i=1}^{n} N_i(x)\phi_i = \mathbf{N}\mathbf{\Phi}, \tag{3.30}$$

and the advection term becomes $V\mathbf{K}\mathbf{\Phi}$, where $\mathbf{K}$ is a global matrix assembled from element matrices:

$$\mathbf{K}_e = \int_0^h (\mathbf{N}_e + \beta V \nabla \mathbf{N}_e)^{\mathrm{T}} \nabla \mathbf{N}_e dx \tag{3.31}$$

Considering the linear elements as shown in Figure (3.1), we have

$$\mathbf{N}_e = [N_e,\ N_{e+1}], \tag{3.32}$$

$$\nabla \mathbf{N}_e = \left[\frac{\partial N_e}{\partial x}, \frac{\partial N_{e+1}}{\partial x}\right]. \tag{3.33}$$

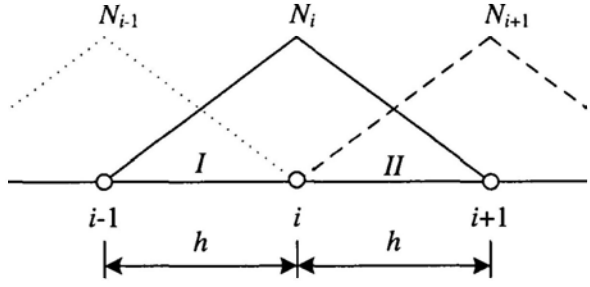


Figure 3.1: Linear shape functions for a one-dimensional problem.

The parameter β defined in Eq. (3.14) becomes the following form, if we ignore the contribution of Δt and note that $\mathbf{J}^{-1}$ is $1/h$ because the local integration domain is $[0, 1]$:

$$\beta = \frac{h}{2|V|}. \tag{3.34}$$

Now, we consider the equation related to node i, in which element matrices of element I and II are involved:

$$\mathbf{K}_I = \mathbf{K}_{II} = \frac{1}{2h}\begin{bmatrix} -1+s & 1-s \\ -1-s & 1+s \end{bmatrix} \tag{3.35}$$

with

$$s = V/|V|,$$

or

$$s = \begin{cases} 1, & \text{if } V > 0 \\ -1, & \text{if } V < 0 \end{cases}$$

After assembly, we have

$$\frac{1}{2h}\begin{bmatrix} -1+s & 1-s & 0 \\ -1-s & 2s & 1-s \\ 0 & -1-s & 1+s \end{bmatrix}\begin{pmatrix} \phi_{i-1} \\ \phi_i \\ \phi_{i+1} \end{pmatrix}. \tag{3.36}$$

So the approximated advection term at node i becomes

$$D\phi_i = \frac{1}{2h}\big((-1-s)\phi_{i-1} + 2s\phi_i + (1-s)\phi_{i+1}\big) \tag{3.37}$$

or

$$D\phi_i = \begin{cases} \dfrac{\phi_i - \phi_{i-1}}{h} & \text{if } V > 0 \\ \dfrac{\phi_{i+1} - \phi_i}{h} & \text{if } V < 0 \end{cases} \tag{3.38}$$

which is just the upwinding scheme as defined in Eq. (2.55).

For multi-dimensional formulations, it is not easy to find the similar relationships between the SDFEM and upwind differencing. In time-dependent problems, β defined in Eq. (3.14) is recommended because it is stated that in addition to depending on the spatial discretization, the weighting function must also depend on the temporal discretization [70]. Numerical results also demonstrate that adding Δt into the paramter can improve the performance of the method.

3.1.3 Numerical Schemes

In this study, to implement the proposed method, we use the four-node quadrilateral element, which is also called Q4 element for simplicity. If

the element is square or rectangle, it is called the regular Q4 element; otherwise, the irregular Q4 element. See Figure (3.2) for the examples. In this section, some numerical schemes related to Q4 element are discussed.

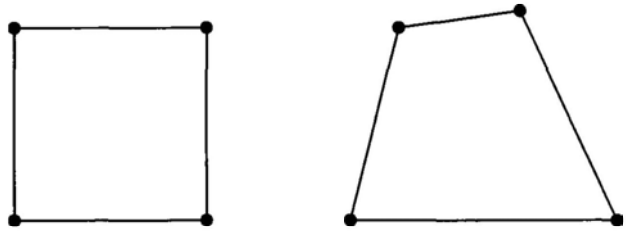


Figure 3.2: Q4 element: regular (left) and irregular (right).

Consistent and Lumped Coefficient Matrix

The integral in Eq. (3.23) is calculated with numerical integration. For the Q4 element, two-point Gaussian quadrature is needed to get the accurate result. The matrix $\mathbf{A}$ is independent on time so it is a constant matrix. This means that although we need to solve the system of equations (3.22) in each step, matrix decomposition is needed only in the first time step and after that only substitution is used. In addition to the property of constant, $\mathbf{A}$ is symmetric and positive definite and therefore is easily to be decomposed.

The coefficient matrix $\mathbf{A}$ is something like the *consistent mass matrix* in structural dynamics problems. Therefore, we call $\mathbf{A}$ the *consistent coefficient matrix* analogously. In structural dynamics problems, the consistent mass matrix is frequently replaced by the so called *lumped mass matrix*. An important advantage of using a lumped mass matrix is that the matrix is diagonal, and the numerical operations for

the solution of the dynamic equations of equilibrium are in some cases reduced very significantly. Similarly, If the coefficient matrix $\mathbf{A}$ is diagonalized to a *lumped coefficient matrix* $\tilde{\mathbf{A}}$, the system of equations in Eq. (3.22) can be solved without decomposing a matrix. After the right-hand-side vector is constructed, the unknown vector are obtained using

$$\delta\phi_i = b_i / \tilde{A}_{ii} \tag{3.39}$$

where b_i is the ith component of the vector $\mathbf{b}$ and $\tilde{A}_{ii}$ is the ith diagonal element of the lumped coefficient matrix. Therefore, using the lumped coefficient matrix is more convenient and economical.

There are various methods for matrix lumping [71,72]. In this study, the row sum method in which

$$\tilde{A}_{ii} = \sum_j A_{ij} \tag{3.40}$$

is used. This method works well for four-node quadrilateral element (Q4). However, it should be noted that the row sum method sometimes produces negative masses for some other types of elements, e.g., the eight-node serendipity element.

Integration Schemes for Right-Hand-Side Vector

To compute the integral in Eq. (3.24), the standard numerical integration scheme is the two-point Gaussian quadrature. However, for regular Q4 element (square or rectangle), one-point scheme is accurate enough, which can be demonstrated by the result of test case (see 3.1.4).

3.1.4 Test Case

In this test case, we consider a circle which shrinks with constant normal velocity $V_n = 1$, as shown in Figure (3.3). The computational domain is a two by two square. The origin radius of the circle is 0.5. Ten steps are computed with 0.01 as the time step size and the final radius of the circle reduces to 0.4.

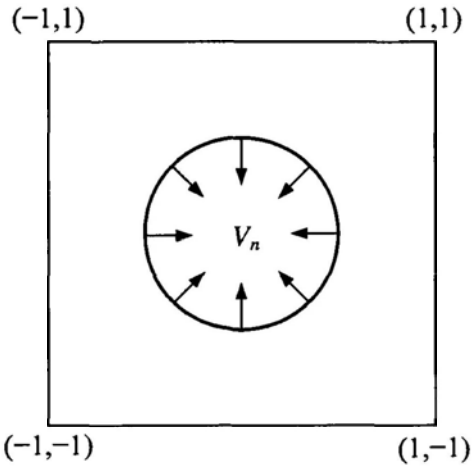


Figure 3.3: A circle shrinks with constant normal velocity.

To measure the accuracy, three error indicators are defined:

1. Error of the interface.

   Assume there are n_I intersections of the exact interface and element edges, and let the coordinates of intersections be denoted by

   $$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_I}.$$

   Then, the error of the interface is defined as

   $$E_I = \left[\sum_{I=1}^{n_I} \left(\phi^h(\mathbf{x}_I)\right)^2\right]^{1/2} \tag{3.41}$$

   where $\phi^h(\mathbf{x})$ means the approximated level set function

2. Error of the area.

   Let Ω^h and Ω^e denote the numerical and exact results of the shrinking circle. Let A^h and A^e denote the area of Ω^h and Ω^e respectively. Then the error of area is defined as

$$E_A = \frac{|A^h - A^e|}{A^e} \times 100\% \tag{3.42}$$

3. Error of the length (perimeter).

   Let L^h and L^e denote the perimeter of Ω^h and Ω^e. Then the error of length is defined as

$$E_L = \frac{|L^h - L^e|}{L^e} \times 100\% \tag{3.43}$$

Uniform four-node quadrilateral elements

First, the computational domain is uniformly meshed with 100 by 100 regular Q4 elements. By uniform, we mean that all elements have the same shapes and sizes. For the proposed finite element based level set method (FELSM), four schemes are tested: NG1lump, NG2lump, NG1cons, and NG2cons. The number following "NG" means the number of Gaussian quadrature points in each dimension. The strings "lump" and "cons" mean the "lumped matrix" and "consistent matrix", respectively. The finite difference based level set method (FDLSM) is also implemented on the same grid for comparison. Four schemes are considered: ENO1, ENO2, ENO3, and WENO, which are first-, second-, third-, and fifth-order accurate, respectively.

Table (3.1) shows the CPU time and errors of virous schemes. All CPU time are normalized with respect to the CPU time of ENO1, which is 0.375 second.

Table 3.1: CPU time and errors of uniform Q4 elements.

Program	Scheme	CPU time	Error		
		(0.375 s)	E_I	$E_A(\%)$	$E_L(\%)$
FDLSM	ENO1	1	1.384e-2	0.4744	0.2360
	ENO2	1.21	7.677e-4	0.0240	0.0120
	ENO3	1.46	4.922e-5	0.0015	7.6e-4
	WENO	3.96	2.545e-7	8.9e-6	4.4e-6
FELSM	NG1lump	1	9.686e-3	0.3927	0.1965
	NG2lump	2.29	9.676e-3	0.3932	0.1967
	NG1cons	9.37	9.684e-3	0.3926	0.1964
	NG2cons	10.7	9.674e-3	0.3931	0.1967

The difference of accuracy among different schemes of FDLSM is obvious. The higher the order, the smaller the errors. However, the accuracies of four schemes of FELSM are almost the same. If we compare the CPU time, we find that the lumped scheme is faster than the consistent scheme. This fact states that using lumped coefficient matrix improves the efficiency significantly while loses nearly no accuracy.

Now, we compare the performances of FDLSM and FELSM. The accuracy of FEMLS is a little higher than ENO1 while much lower than other higher-order FDLSM schemes. The speed of NG1lump, which is the fastest one in FELSM, is similar to ENO1. However, CPU time of other FELSM schemes are larger than FDLSM. It should be noted that the code can be optimized to speed up for uniform mesh. If the non-uniform mesh mentioned later is used, the CPU time for FELSM will be even larger.

It seems that higher-order finite difference schemes are superior to

FELSM because of the high order of accuracy. However, it is noteworthy that the velocity field is simple and can be calculated analytically in this test case. Therefore, the accuracy of the results depends only on the accuracy of discretization of the level set equation. In the structural optimization problems, the velocity can not be obtained analytically. If the velocity is of lower-order accuracy, higher-order results can not bee obtained even higher-order level set method is used. This will be shown in numerical examples.

Irregular four-node quadrilateral elements

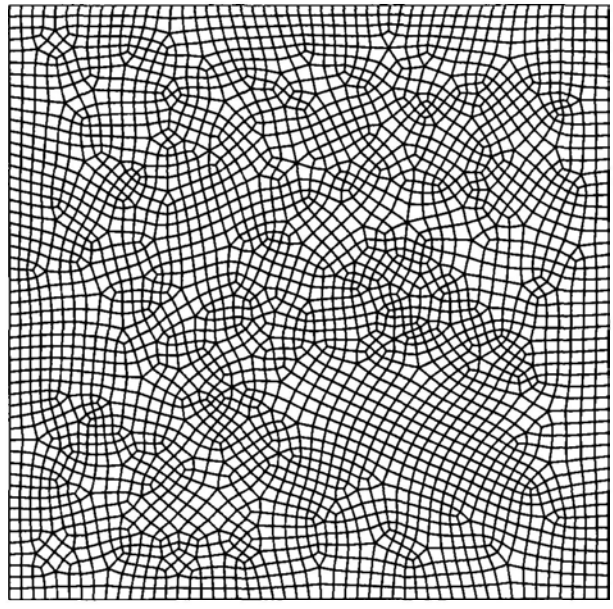


Figure 3.4: A mesh of irregular four-node quadrilateral elements.

Next a mesh containing 2920 irregular Q4 elements and 3023 nodes, as shown in Figure (3.4) is considered. This mesh is automatically generated by ALGOR [73]. Mesh generation and the influence of meshes to the performance of FELSM beyond the scope of this study. Hence, we

just generate a mesh and use it. Table (3.2) shows erros of four schemes of FELSM. It is shown that errors of two-point Gaussian quadrature are lower than one-point quadrature. This is coincident with the principle of quadrature in finite element technique, which states that the reduced integration is often insufficient for accuracy when irregular Q4 element is considered.

Table 3.2: Errors of irregular Q4 elements.

Scheme	E_I	$E_A(\%)$	$E_L(\%)$
NG1lump	0.02139	0.3968	0.1557
NG2lump	0.00919	0.4622	0.2291
NG1cons	0.02757	0.3935	0.3687
NG2cons	0.00912	0.4528	0.2251

Three-node triangular elements

At last the computational domain is divided into 3632 three-node triangular (T3) elements and 1897 nodes as shown in Figure (3.5). For triangular elements, area coordinates are used [71] and one-point and three-point integration are tested. Errors are listed in Table (3.3).

Table 3.3: Errors of triangular elements.

Scheme	E_I	$E_A(\%)$	$E_L(\%)$
NG1lump	0.01085	0.5160	0.2585
NG3lump	0.01085	0.5160	0.2585
NG1cons	0.01076	0.5108	0.2560
NG3cons	0.01076	0.5108	0.2560

The accuracy of the consistent matrix is a little higher than the

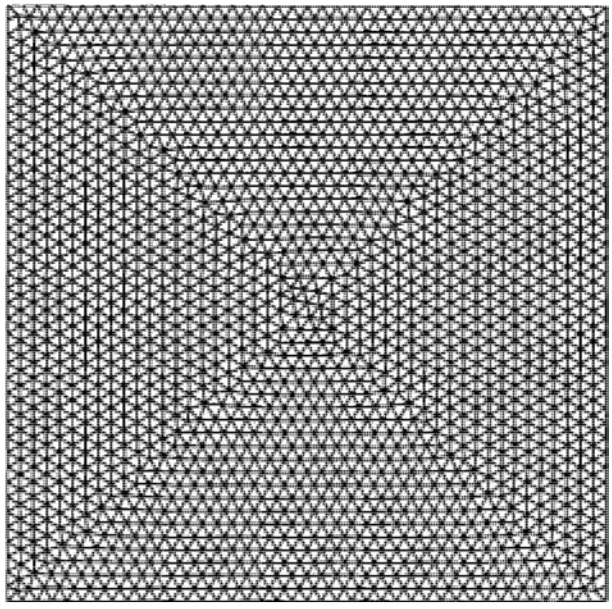


Figure 3.5: A mesh of three-node triangular elements.

lumped one. For either types of matrices, the accuracy of three-point quadrature is the same as one-point quadrature. To explain this results, let's observe Eq. (3.24). On T3 elements, shape functions are linear and their derivatives are constant. The velocity is constant (equals 1 everywhere) in this test case. Consequently, the integrand in Eq. (3.24) is linear and one-point Gaussian quadrature is enough to obtain accurate retults.

However, in practice, V_n is not always constant. For example, in structural optimization problems considered in this study, velocity is sometimes described by shape functions (see Chapter 4). In this case, the integrand becomes quadratic polynomials, for which one-point Gaussian quadrature is inadequate.

Actually, T3 elements are not recommended because of its lower accuracy. It is well known that T3 element is strain constant element

for structural analysis. As a result, the velocity field which is calculated based on strains and stresses is inaccurate and this inaccuracy will lead to rough boundary. If one has to use T3 elements in some cases, such as meshing a very irregular domain or the transition zone between two types of meshes, he should pay attention to the issue of accuracy.

3.1.5 Discussion

It seems that higher-order finite difference schemes (ENO2, ENO3, and WENO) are more accurate than the finite element based schemes. This is true in the test case and some applications while it is not the case in the structural optimization problems. In this test case, the velocity is simple and can be expressed analytically. Errors of the results come from the errors of approximating the derivatives. Hence, higher-order schemes produce higher-order results. In structural optimization problems, however, the velocity can not be described analytically and it can only be approximated in a low-order manner. For instance, if Q4 elements and bilinear interpolations are used, the final velocity field is only first-order accurate. Under the circumstances, even fifth-order accurate WENO scheme can only give first-order accurate results because of the velocity bottleneck.

CPU times listed in Table (3.1) are used to give us an intuitive understanding of the efficiency of the FELSM. In this case, NG1lump is similar to ENO1 in CPU time; but if the scale of the problem increase, cost for FELSM might go up more rapidly than that for FDLSM. Further analysis of the time complexity is necessary, if one want to know the rigorous algorithm efficiency. It is worthwhile to remark that one level set step is much cheaper than the solution of the state equation.

For examples, in this study, time for one level set step is always less than one fifth of the time for one optimization step. Hence, increase in the time for the level set method will not affect the optimization time dramatically.

3.2 Discretization of the Reinitialization Equation

3.2.1 Formulation

Similar as in 3.1, the weak formulation of Eq. (2.50) is expressed as: Let $\psi^0 = \phi^n$ and suppose ψ^i is known, find $\psi^{i+1} \in H^1$ such that

$$\begin{aligned} &(\psi^{i+1} - \psi^i, v) + \Delta\tau(\epsilon\nabla\psi^{i+1}, \nabla v) \\ =& -\Delta\tau(\mathbf{w}^i \cdot \nabla\psi^i, \tilde{v}) + \Delta\tau(S, \tilde{v}) \qquad \forall v \in H^1 \end{aligned} \tag{3.44}$$

where S is the sign function defined in Eq. (2.53) and

$$\tilde{v} = v + \beta_2 \mathbf{w}^i \cdot \nabla v \tag{3.45}$$

where $\mathbf{w}$ is defined as in Eq. (2.51). The parameter β_2 is

$$\beta_2 = \frac{1}{2\sqrt{(\Delta\tau)^{-2} + |\mathbf{J}^{-1}\mathbf{w}^i|^2}}, \tag{3.46}$$

and the time step size is selected according to the CFL condition:

$$\Delta\tau = \alpha\frac{h}{\max\{|\mathbf{w}|\}} \qquad \alpha \in (0,\ 1) \tag{3.47}$$

with $\alpha = 0.5$ in this study. From Eq. (2.51) we know that $\max\{|\mathbf{w}|\} = 1$, hence $\Delta\tau$ is actually $h/2$.

Eq. (3.44) should be solved iteratively until the steady state to obtain the signed distance function. With numerical algorithms, however, achieving the steady state is impractical. Once the solution has converged according to some criteria, one can think of the result as a steady

solution. An indicator is usually defined to determine the convergence:

$$I_c = \Big[\frac{1}{n_p}\sum_{i=1}^{n_p}(|\nabla\psi_i| - 1)^2\Big]^{1/2}, \tag{3.48}$$

where n_p is the number of points at which the gradient is evaluated. If $I_c \leq \omega$ (ω is a small number specified in advance), the solution is regarded as convergence and the iteration stops.

It should be noted that an extra diffusion term is added into Eq. (3.44), since the streamline diffusion modification gives an insufficient diffusion effect around the boundary, where S is small and $\mathbf{w}$ is therefore small in magnitude [30]. The factor ϵ of the diffusion term is decided according to the following criterion

$$\epsilon = c\frac{h^2}{\Delta\tau} \tag{3.49}$$

where c is a constant coefficient to adjust the diffusion effect. This coefficient should be selected properly. On one hand, it provides enough diffusion effect to stable the numerical results, especially results near the boundary; on the other hand, it should not be so large that extra numerical errors are introduced. We suggest that c is selected in $[0.1, 1]$. One can use firstly a relatively small value, say 0.1, and if he find that the iterations of the reinitialization equation are unstable, for example, the value of I_c increases, he can increase c until the iterations become stable. In this study, $c = 0.1$ is used and it is enough for stable results.

Equation (3.44) can be written as

$$\begin{aligned}&(\psi^{i+1}, v) + \Delta\tau(\epsilon\nabla\psi^{i+1}, \nabla v)\\ =&\Delta\tau(S - \mathbf{w}^i\cdot\nabla\psi^i, \tilde{v}) + (\psi^i, v) \qquad \forall v \in H^1\end{aligned} \tag{3.50}$$

After discretization, the system of finite element equations is

$$\mathbf{C}\boldsymbol{\Psi}^{i+1} = \mathbf{R} \tag{3.51}$$

with

$$\mathbf{C} = \mathbf{A} + \Delta\tau\epsilon\mathbf{B} \tag{3.52}$$

$$\mathbf{R} = \Delta\tau(\mathbf{R}_1 - \mathbf{R}_2) + \mathbf{R}_3 \tag{3.53}$$

The matrix $\mathbf{A}$ is same as Eq. (3.23) and $\mathbf{B}$ is

$$\mathbf{B} = \int_D (\nabla\mathbf{N})^{\mathrm{T}}\nabla\mathbf{N}\, d\Omega \tag{3.54}$$

Similar as $\mathbf{A}$, the matrix $\mathbf{B}$ is also a constant matrix. Therefore the coefficient matrix in Eq. (3.50) is constant since both $\Delta\tau$ and ϵ are constant. Only one time of matrix factorization is needed. The right-hand-size vectors are

$$\mathbf{R}_1 = \int_D \widetilde{\mathbf{W}}^{\mathrm{T}} S\, d\Omega \tag{3.55}$$

$$\mathbf{R}_2 = \int_D \widetilde{\mathbf{W}}^{\mathrm{T}}(\mathbf{w}^{\mathrm{T}}\nabla\psi^i)\, d\Omega \tag{3.56}$$

$$\mathbf{R}_3 = \int_D \mathbf{N}^{\mathrm{T}}\psi^i\, d\Omega \tag{3.57}$$

with

$$\widetilde{\mathbf{W}} = \mathbf{N} + \beta_2\mathbf{w}^{\mathrm{T}}\nabla\mathbf{N} \tag{3.58}$$

3.2.2 Enforce Dirichlet Boundary Condition

In theory, the boundary remains stationary during the reinitialization procedure because the velocity $\mathbf{w}$ is zero on the interface. But in practice, there are two factors that will tend to move the boundary. The first factor is the numerical errors introduced by discretization and the second one is the diffusion term in Eq. (3.44). To fix the boundary during reinitialization, the Dirichlet boundary condition

$$\psi(\mathbf{x}) = 0 \qquad \mathbf{x} \in \partial\Omega \tag{3.59}$$

should be enforced. The procedure of enforcing this boundary condition is explained with a Q4 element. The method for other types of elements is similar. Consider a Q4 element intersected by the boundary as shown in Figure (3.6). The Dirichlet boundary condition require

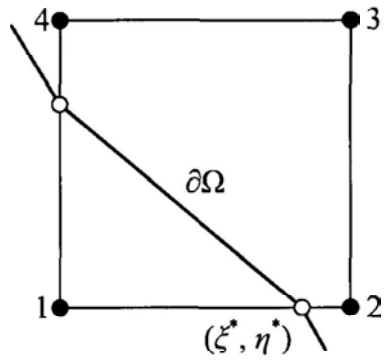


Figure 3.6: A Q4 element intersected by the boundary.

$$\sum_{i=1}^{4} N_i(\xi^*, \eta^*)\psi_i = 0 \tag{3.60}$$

Hence, for each intersection, there is a submatrix

$$\mathbf{G}_I = \begin{bmatrix} N_1(\xi^*, \eta^*) \\ N_2(\xi^*, \eta^*) \\ N_3(\xi^*, \eta^*) \\ N_4(\xi^*, \eta^*) \end{bmatrix}^{\mathrm{T}} \tag{3.61}$$

Actually, shape functions N_3 and N_4 vanish on the considered intersection, so the submatrix reduces to

$$\mathbf{G}_I = [N_1(\xi^*, \eta^*),\ N_2(\xi^*, \eta^*)] \tag{3.62}$$

A global matrix $\mathbf{G}$ can be assembled

$$\mathbf{G}(I, loc) = \mathbf{G}_I \tag{3.63}$$

where I is the index of intersection and loc is the locations of nodes in the global nodes sequence. We can now describe the Dirichlet boundary condition as a series of constraint equations:

$$\mathbf{G\Psi} = \mathbf{0} \tag{3.64}$$

This constraint can be enforced with the Lagrange multiplier method, in which the system of equations is augmented to

$$\begin{bmatrix} \mathbf{C} & \mathbf{G}^{\mathrm{T}} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{\Psi} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} \tag{3.65}$$

or with the penalty method

$$(\mathbf{C} + \rho \mathbf{G}^{\mathrm{T}}\mathbf{G})\mathbf{\Psi} = \mathbf{R} \tag{3.66}$$

where ρ is a large number.

Numerical results show that the Dirichlet boundary condition is enforced satisfactorily and drift of the boundary during reinitialization is tiny.

3.2.3 Test Case

In this test case, level set functions of the interface shown in Figure (3.3) are reinitialized with the proposed method. For the circle in Figure (3.3), the signed distance function is

$$\phi_s = R - \sqrt{x^2 + y^2} \tag{3.67}$$

where $R = 0.5$ is the radius. Given two level set functions with respect to the circle, $\phi_1 = 2\phi_s$ and $\phi_2 = 0.5\phi_s$, we now rebuild them to the sign distance function.

First, a structured mesh with 50 by 50 uniform Q4 elements are used. The time step size $\Delta\tau$ is 0.025 and 20 iterative steps are run. In theory, the reinitialization equation propagate information at speed 1 in the direction normal to the interface and the information will be moved to 0.5 away from the interface after 20 iterations. In other words, in this test case, the information will occupy a unit circle after 20 iterations, whose center lies in the origin. The level set function on this unit circle should be reset to the signed distance function.

However, because of the smoothness of the sign function S and numerical diffusion, the speed is always less than 1 and so the reinitialization is slower than theory. Figure (3.7) shows the results of reinitializing ϕ_1. Some contours of the final level set function are plotted. These contours correspond to $\phi = [-0.5 : 0.1 : 0.5]$ and the bold line is the interface where $\phi = 0$. Dash grid lines help to judge whether these contours have reached their theoretical positions. One can see that there are six contours that have almost reached the right positions. They are contours of $|\phi| \leq 0.3$. The others have not been completely.

Next, ϕ_2 is reinitialized in the same mesh. Figure (3.8) shows those contours, whose values are same as those in Figure (3.7). Once again, one can find out that numerical speed is slower than theoretical speed.

To measure the difference between the numerical results and the exact signed distance function, an error is defined as

$$Err_n = \frac{|\mathbf{\Phi}_s(I_n) - \mathbf{\Phi}(I_n)|}{|\mathbf{\Phi}_s(I_n)|} \times 100 \tag{3.68}$$

where $\mathbf{\Phi}_s$ is the vector of nodal values of the signed distance function,

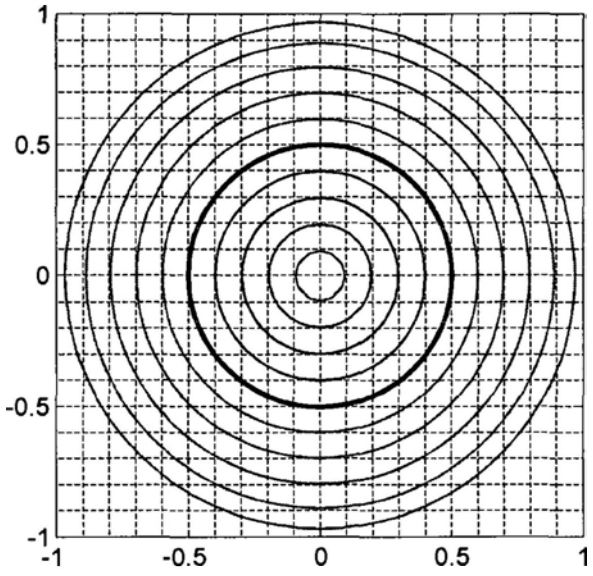


Figure 3.7: Reinitializing ϕ_1

$\mathbf{\Phi}$ is the nodal values of the reinitialized function, and I_n means the set of nodal indices,

$$I_n = \{i : |\mathbf{\Phi}_s(i)| \leq n \times d\} \tag{3.69}$$

with $d = 0.1$. Figure (3.9) shows the errors of NG1cons and NG2cons schemes, with n from 1 to 5. It is apparent that errors increase rapidly from $n = 3$, which coincides with the conclusions obtained by observing the contours.

Next, we use the free mesh shown in Figure (3.4) to reinitialize ϕ_1. Errors are exhibited in Figure (3.10). Contrary to the case in solving the level set equation, the one-point quadrature scheme on irregular Q4 element works well when solving the reinitialization equation.

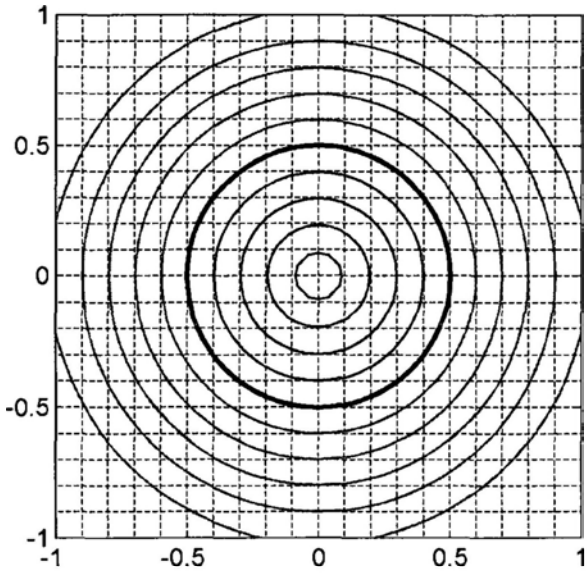


Figure 3.8: Reinitializing ϕ_2

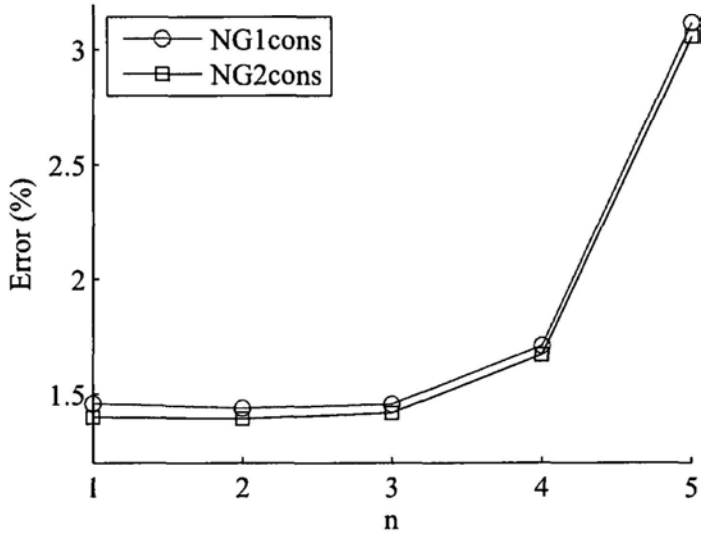


Figure 3.9: Error on uniform Q4

3.2.4 Discussion

For a particular application, one has to decide how sensitive the relevant techniques are to ϕ's approximation of a signed distance function. If

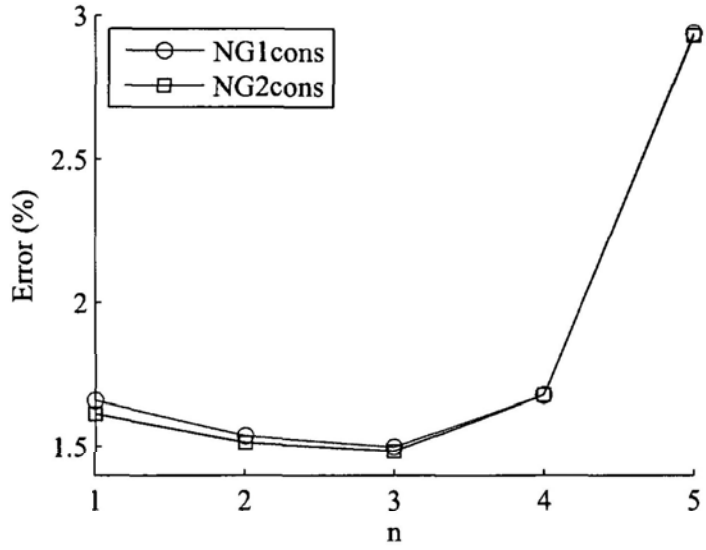


Figure 3.10: Error on free Q4

they are very sensitive, ϕ needs to be reinitialized to signed distance both accurately and often. If they are not sensitive, one can reinitialize with a lower-order accurate method on an occasional basis.

Fortunately, the proposed method, when employed in structural optimization problems, is not sensitive to whether the level set function is a strict signed distance function or not. Therefore, we can perform one reinitialization every three or five steps. Furthermore, in each reinitialization, there is no need to solve the reinitialization equation to steady state. Several iterations are enough.

□ **End of chapter.**

Chapter 4

Velocity Extension

To solve the level set equation, the normal velocity V_n must be defined on the whole computational domain, or at least on Gaussian quadrature points. However, we can get only the velocity on the boundary from the shape derivative. Therefore, an extension from the boundary to the entire design domain should be preformed and the new velocity field is known as the "extension velocity".

Recall that the level set technique relies on the embedding of the moving interfaces as the zero level set of higher dimensional time-dependent function. We can consider the extension as the embedding of interface's velocity to this higher dimensional level set function. This means that the velocity is now defined for all level sets, not just the zero level set.

In practice, we have considerable freedom to construct the extension velocity as soon as the following constraint is satisfied,

$$V_{ext} = V_n \quad \text{on } \phi = 0. \tag{4.1}$$

In the level set field, several extension methods have been intro-

duced, such as the fast marching method [74] and the PDE-based method [62]. Although all these methods can be employed in the structural optimization problems, we prefer the PDE-based method because we are using FEM. Actually, there is also a natural and physically meaningful extension method in structural optimization problems, which is called the "natural extension".

From Eq. (2.36), we know that velocity involves the computation of stresses and strains. So the accuracy of velocities depends on the accuracy of the stress analysis. Before introducing the extension methods, we shall discuss the issue of computing velocity.

4.1 Velocity Computation

4.1.1 Stress Analysis

A challenge to structural topology optimization is the fact that if the conventional FEM is used for the stress analysis, the finite element mesh will become distorted after the shape and topology change. Under these circumstances, the structure domain must be remeshed. However, remeshing is a complicated and time consuming task, and will bring down the efficiency of optimization.

In this study, we use the so-called "ersatz material" approach [14] which has been widely used in stress analysis of the compliance optimization problem. 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

$$E_0 = c \cdot E \tag{4.2}$$

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 intersected by the boundary, Young's modulus is calculated according to the fraction of solid material. For example, in one element, if the volume of solid takes one half of the volume of the element, then Young's modulus of this element is set to $0.5E$.

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

$$\begin{aligned} -\text{div}\,\boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{f} \quad \text{in} \quad D, \\ \mathbf{u} &= 0 \quad \text{on} \quad \Gamma_D, \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{g} \quad \text{on} \quad \Gamma_N. \end{aligned} \tag{4.3}$$

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. If more accurate stresses are required, such as in minimum stress optimization, some advanced techniques should be adopted, e.g., the extended finite element method (XFEM). About the implementation of XFEM in topology optimization, the reader is referred to [75] and references herein.

In this study, once a mesh is generated, it will be used for stress analysis, velocity extension, and level set evolution.

4.1.2 Recovery of Stresses

With bilinear Q4 element, the stress is piecewise continuous. On edges of elements, the stress is discontinuous. So we can not get a smooth stress field. This can be resolved by using the recovery of stresses.

The recovery of stresses is also known as the "stress smoothing". In this study, the superconvergent patch recovery (SPR) method [71, 76] is adopted. In this method, the stresses are first evaluated on the superconvergent sampling points because the values of stresses on these points are more accurate than those on other points. Typically, the superconvergent sampling points are Gaussian quadrature points. If we take the Q4 element as an example, the superconvergent sampling point is the point where one-point Gaussian quadrature is executed. Subsequently, nodal values of stresses are recovered from stresses on superconvergent sampling points. Stresses on any point can be obtained by interpolation in the same manner as the displacements. This recovered stress field is smooth and has the same order of accuracy as the displacement field. For details. the reader is referred to the cited references.

4.2 Natural Velocity Extension

Equation (2.36) tells us that the boundary normal velocity depends on the displacements, stresses, and strains. All these quantities have physical meanings on the whole design domain by virtue of the ersatz material approach. So the extension velocity can be naturally obtained by evaluating these quantities everywhere and using Eq. (2.36):

$$V_{ext} = G - \lambda \tag{4.4}$$

where

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

Although this method is simple, a problem will arise if there are stresses concentrations or singularities. When these phenomena happen, stresses and hence velocity on the relevant regions become enormous. If we consider the CFL condition:

$$\Delta t = \alpha \frac{h}{\max\{V_{ext}\}}, \quad \alpha \in (0,1), \tag{4.5}$$

the time step size has to be very small to satisfy it. As a result, the propagation of the boundary is slow and the optimization takes a large of steps to converge.

If the regions of stresses concentrations and singularities are what we care about, we have to resort to some techniques to relieve the concentrations or singularities. However, in the minimum mean compliance problems—at least in problems in this study—stresses concentrations and singularities occur always on the regions which are either far away from the boundary or within the non-designed domain. There is no need to consider these extreme stresses within the context of compliance optimization.

Therefore, instead of eliminating stresses concentrations and singularities, we use a relatively simpler "modified natural extension method" to circumvent the difficulty. This method reads: Define B as a narrow band around the boundary, then the extension velocities are calculated with the following formulation,

$$V_{ext} = \begin{cases} G - \lambda & \text{on } B \\ \bar{G} - \lambda & \text{on } D\backslash B \end{cases} \tag{4.6}$$

where

$$\bar{G} = \min\{G, G^B_{max}\}$$
$$G^B_{max} = \max\{G\} \quad \text{on } B$$

The modified natural extension method pays attention to the velocity near the boundary, which influence the boundary propagation directly. On the region outside the narrow band B, extreme values of velocity are cut off.

4.3 PDE Based Extension

The idea of the PDE-based extension method is first proposed and analyzed in the appendix of [77], but the authors do not try to implement it. Later, this method is employed in [78] and extensively discussed in [62]. Suppose we have a quantity q defined on the interface Γ. The most natural way to extend q off Γ is to let q be a constant along the characteristic line normal to Γ. This suggests the following hyperbolic PDE,

$$q_t + S(\phi)\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla q = 0, \tag{4.7}$$

where $S(\phi)$ is the sign function of ϕ and is approximated by

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \alpha^2}} \tag{4.8}$$

where α is a small smoothing parameter which can be taken as Δx.

In [28], Eq. (4.7) is solved with the method proposed by Barth et al. [26]. Here, we solve it with the SDFEM. If we define

$$\mathbf{w} = S(\phi)\frac{\nabla\phi}{|\nabla\phi|}, \tag{4.9}$$

Eq. (4.7) can be written as

$$q_t + \mathbf{w} \cdot \nabla q = 0. \tag{4.10}$$

Obviously, this is a first-order hyperbolic PDE and can be discretized with SDFEM as follows: Suppose $q^n \in H^1$ is given, find $q^{n+1} \in H^1$, such that

$$\left(\frac{q^{n+1} - q^n}{\Delta t}, v\right) + \epsilon\left(\nabla q^{n+1}, \nabla v\right) + \left(\mathbf{w}^n \cdot \nabla q^n, \bar{v}\right) = 0, \quad \forall v \in H^1 \tag{4.11}$$

where

$$\bar{v} = v + \beta \mathbf{w}^n \cdot \nabla v, \tag{4.12}$$

$$\beta = \frac{1}{2\sqrt{(\Delta t)^{-2} + |\mathbf{J}^{-1}\mathbf{w}^n|^2}}. \tag{4.13}$$

Similar to SDFEM discretization of the reinitialization equation, an extra diffusion term is added into Eq. (4.11), because the velocity $\mathbf{w}$ is small near the boundary and consequently the streamline diffusion effect is too small to stablize the numerical results. The coefficient ϵ of the extra diffusion term can be selected as in Eq. (3.49).

The system of finite element equations is

$$\left(\mathbf{A} + \Delta t \epsilon \mathbf{B}\right)\mathbf{Q} = \mathbf{R} \tag{4.14}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined as in Eq. (3.23) and Eq. (3.54) respectively, and

$$\mathbf{R} = \mathbf{R}_1 - \Delta t \mathbf{R}_2 \tag{4.15}$$

$$\mathbf{R}_1 = \int_D \mathbf{N}^{\mathrm{T}} q^n d\Omega \tag{4.16}$$

$$\mathbf{R}_2 = \int_D (\mathbf{N} + \beta \mathbf{w}^{\mathrm{T}} \nabla \mathbf{N})^{\mathrm{T}} \nabla q^n d\Omega \tag{4.17}$$

The advantage of the PDE-based extension method is that the obtained velocity field has a tendency to preserve the signed distance

function. As discussed in [77], if ϕ_0 is a signed distance function, updating it with the extension velocity obtained from Eq. (4.7) produces a new level set function ϕ_{new}, which is also a signed distance function. In this sense, the reinitialization stage can be omitted.

Nevertheless, there are some disadvantages of the PDE-based method. First, this method involves solving PDE which makes it more time-consuming than the natural extension method. In addition, to get the property of preserving signed distance functions, the PDE must be solved to steady state. This asks for many iterations if the design domain is large, because the information can only be extended forward a distance of Δt in each iteration, similar as in reinitialization process. The second disadvantage is due to the numerical errors. Although the signed distance function should be retained in theory, numerical errors always weaken this ability and the signed distance function degenerates gradually. So we have to resort to the reinitialization again.

As stated in [74], if one's goal is to extend the velocity in a narrow band several cells around the interface, one might try the iterative technique. Because in the narrow band scheme, the number of freedom in Eq. (4.14) is small and solving the system of equations is fast. Furthermore, in this case, the computational domain is the narrow band and several iterations are enough to extend the velocity to the whole band. On the contrary, if the whole domain scheme is considered, the natural extension method is appreciate.

In this study, only the whole domain scheme is considered, hence the natural extension method is used. The narrow band scheme will be investigated in the future.

□ **End of chapter.**

Chapter 5

Numerical Examples

In this chapter, the proposed method is implemented to solve some compliance optimization problems in two dimensions. Unless stated otherwise, all 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 $\nu = 0.3$. The convergence criterion for the optimization is defined as follows: If

$$\frac{|J^n - J^{n-1}|}{J^{n-1}} \leq 1e-6,$$

the optimization process terminates and the design of step n is the final design. However, we have found that this 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 as 200. For all the numerical examples tested in this study, this number is enough and no obvious change in the designs and the objective functions is found even if more steps are used.

5.1 A Cantilever Beam

The minimum compliance design problem of a cantilever beam is shown in Figure (5.1), which is a well known benchmark of the structural topology optimization problem. The design domain is a rectangle with $L = 2$ and $H = 1$. A vertical concentrated load $P = 1$ acts on the middle of the right edge and the left edge of the cantilever beam is fixed. As in the current problem, the obejctive 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. The maximum allowable volume fraction is 0.5. An initial design is shown in Figure (5.2). With current level set algorithm, no hole can nucleate inside a material region in two dimensions. The number of holes always decreases through merging of initial holes. Therefore, in this chapter, all initial designs have more or less interior and exterior holes. The implicit level set function is initially defined as a signed distance function with respect to the initial design. It is pointed out in [14] that, in 2D, the best results are obtained if the number of holes of the initial design is sufficiently large. In this study, we don't investigate the effect of different initial designs. Only one initial design is used for each numerical example. It is also shown in [14] that, in general 3D case, new holes can easily be created and then the initial topology is less important.

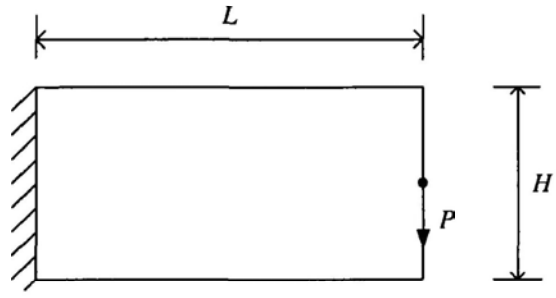


Figure 5.1: A cantilever beam

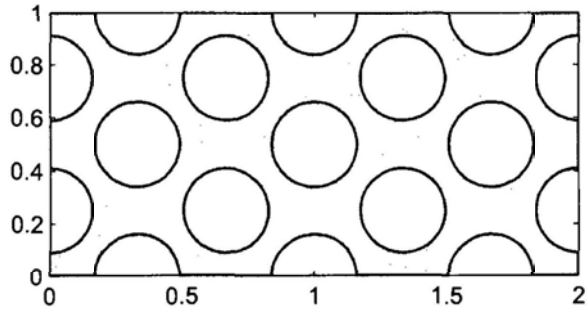
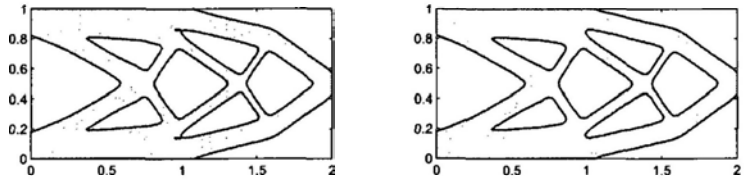


Figure 5.2: The initial design of the cantilever beam

5.1.1 Structured Mesh

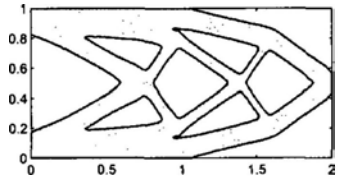
Firstly, a structured mesh consisting of 100 by 50 uniform Q4 elements is used. Each element is a 0.02 by 0.02 square. Four schemes of FELSM and four schemes of FDLSM are tested. Final designs are presented in Figure (5.3). All results have the same topology, shape, and even the details except that in Figure (5.3(a)) and (5.3(c)), there are two little gaps at the corners of interior holes. On the one hand, these are not good designs because these gaps may cause stress concentration and cracks will propagate from the tips. On the other hand, designs obtained by NG1lump and NG1cons possess less compliance than designs obtained by their two-point Gaussian quadrature counterparts, as listed in Table (5.1). The appearance of small gaps has been found

for a MBB beam problem in [18] and the authors explain that these gaps mean the hinge-like or pin connections.

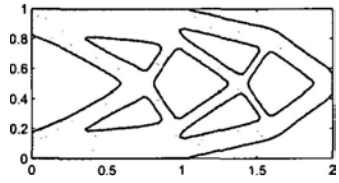


(a) NG1lump

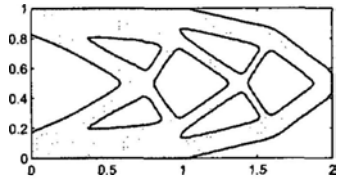
(b) NG2lump



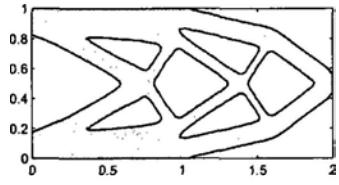
(c) NG1cons



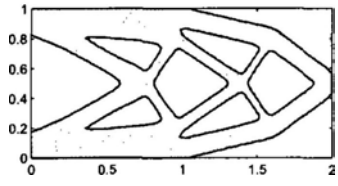
(d) NG2cons



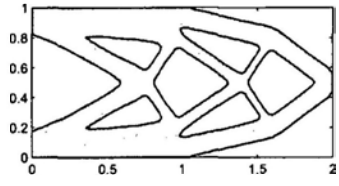
(e) ENO1



(f) ENO2



(g) ENO3



(h) WENO

Figure 5.3: Final design of the cantilever beam

It seems that higher-order ENO schemes and WENO scheme have no apparent advantage in accuracy in this example. The same fact is also stated in [14] that there is no clear difference on the objective

Table 5.1: Mean compliance of the final cantilever beam

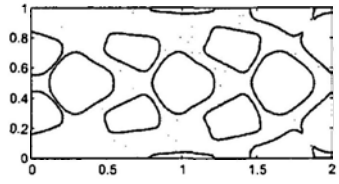
FELSM	NG1lump	NG2lump	NG1cons	NG2cons
	29.8421	29.8549	29.8153	29.8327
FDLSM	ENO1	ENO2	ENO3	WENO
	29.8528	29.8435	29.8424	29.8409

function between a first-order or second-order scheme for the Hamilton–Jacobi equations (both the level set equation and the reinitialization equation). The optimal shapes for first or second order schemes are slightly different, mainly near the boundary of the design domain D.

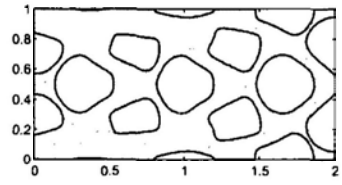
Figure (5.4) displays some intermediate designs produced by NG2cons and WENO. The optimal topology is achieved after about 60 steps, and the subsequent steps are mainly performed to achieve the optimal shape. It is shown that the evolution speed of the boundary is slower when WENO is used. This fact can also be observed in Figure (5.5), where the compliance obtained by NG2cons tends towards stability faster than that obtained by WENO.

5.1.2 Free Quadrilateral Mesh

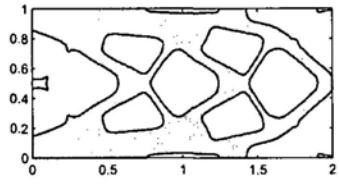
To illustrate the performance of the proposed method on unstructured mesh, we implement the method on a free mesh that contains 4864 bilinear Q4 elements and 5015 nodes as shown in Figure (5.6). Figure (5.7) displays the final level set function obtained by NG2lump scheme and the black line means the boundary of the final design. The final level set function is quite regular with current frequency of reinitialization. Figure (5.8) shows the convergence history of the objective function and the volume fraction. Along with the convergence of the



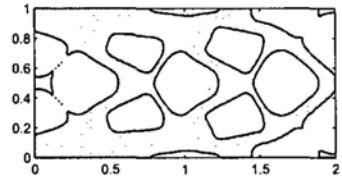
(a) NG2cons, step 32



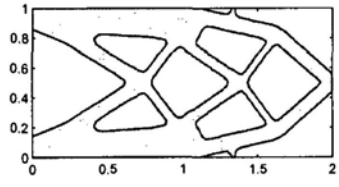
(b) WENO, step 32



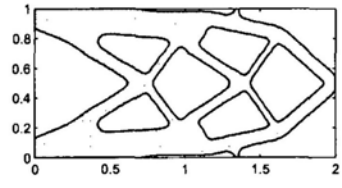
(c) NG2cons, step 37



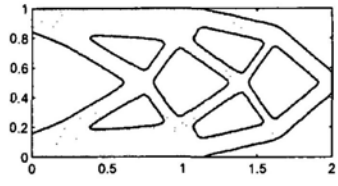
(d) WENO, step 37



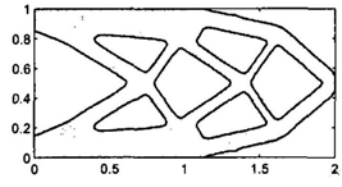
(e) NG2cons, step 55



(f) WENO, step 55



(g) NG2cons, step 65



(h) WENO, step 65

Figure 5.4: Intermediate designs of the cantilever beam

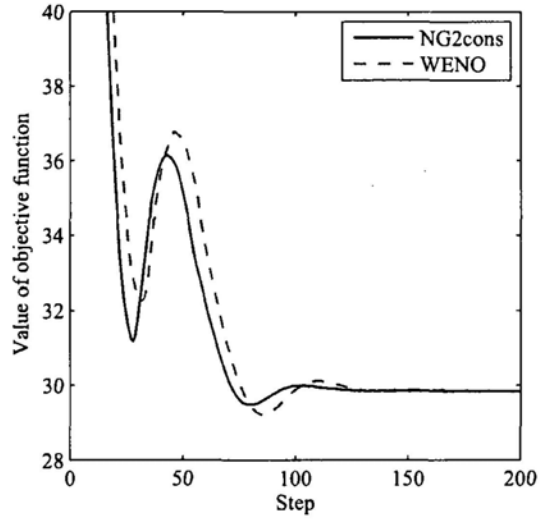


Figure 5.5: Convergence history of the cantilever beam on the structured mesh.

mean compliance, the constraint on volume is gradually satisfied by using the augmented Lagrangian method.

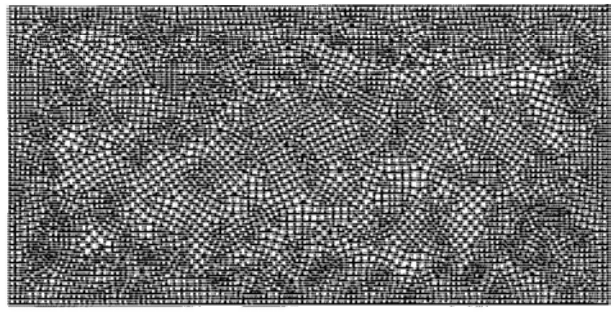


Figure 5.6: A free quadrilateral mesh for the cantilever beam.

5.1.3 Free Triangular Mesh

A free triangular mesh containing 5550 elements and 2896 nodes as shown in Figure (5.9) is also considered for this example. Three-point integration is used to calculate the right-hand-side vector of the sys-

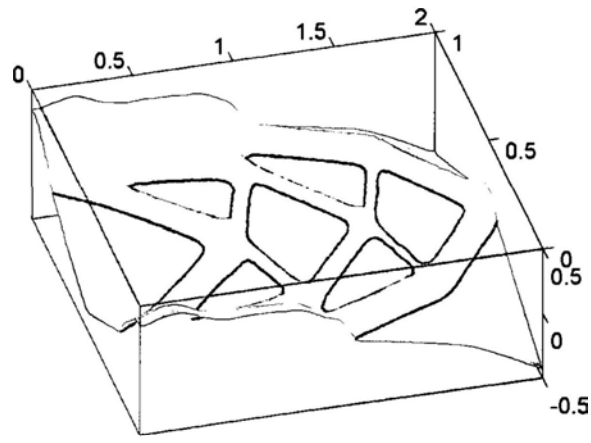
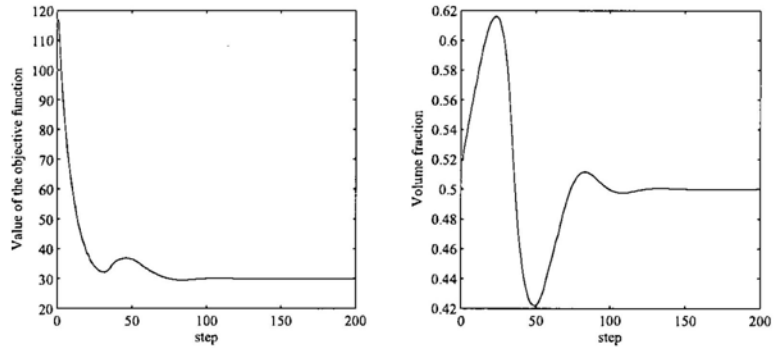


Figure 5.7: Final level set function of the cantilever beam on free Q4 mesh



(a) objective function (b) volume fraction

Figure 5.8: Convergence history of the cantilever beam on free Q4 mesh

tem of level set equations and lumped coefficient matrix is used, so the scheme on the triangular mesh is "NG3lump". The optimization process terminates after 114 steps since the specified convergence criterion is satisfied. Figure (5.10) shows the final design. To compare the results on the structured mesh, the free quadrilateral mesh, and the free triangular mesh, two regions in the dashed and solid rectangles are enlarged and exhibited in Figure (5.11). The left and right columns show the regions in the dashed and solid rectangles, respectively. The

boundaries are generated as follows: For each element that is crossed by the boundary, we calculate the coordinates of intersection points of the boundary and element edges according to nodal level set value and in each element, the boundary is assumed as a straight line connecting two intersection points. One can see the swings of the boundaries when the problem is solved on the free triangular mesh (Figure 5.11(e) and 5.11(f)). Boundaries on quadrilateral meshes are smoother and the uniform Q4 mesh gives the best results.

As we have mentioned in Chapter 3, the triangular mesh is not recommended for structural optimization because it produce lower-order accurate velocity fields. If one has to use triangular elements, he should be careful about this issue.

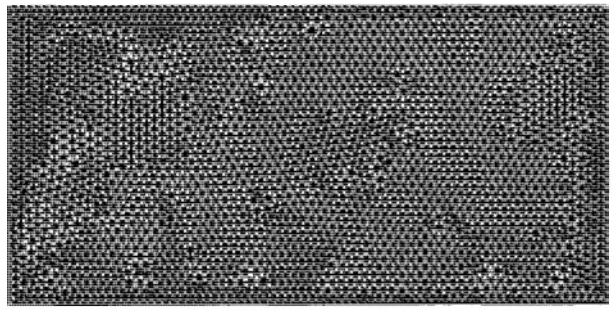


Figure 5.9: A free triangular mesh for the cantilever beam.

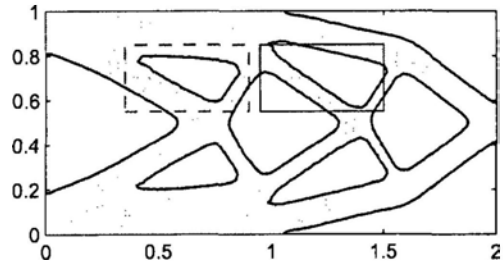
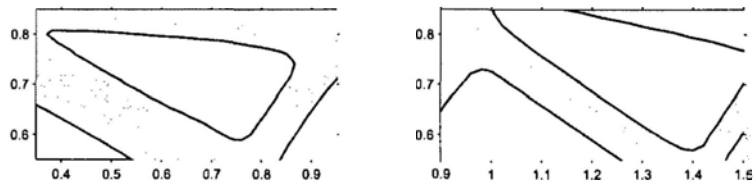
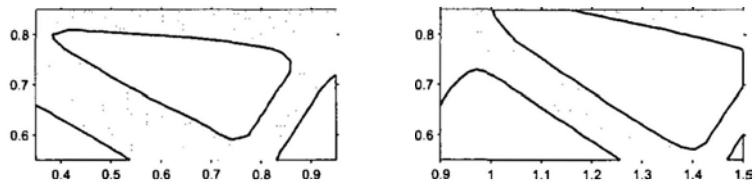


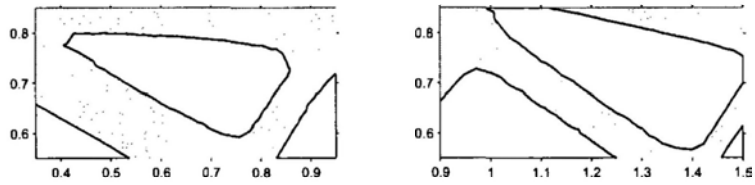
Figure 5.10: Final design of the cantilever beam on free T3 mesh



(a) uniform Q4 (b) uniform Q4



(c) free Q4 (d) free Q4



(e) free T3 (f) free T3

Figure 5.11: Enlarged regions in the dashed rectangle (the left column) and in the solid rectangle (the right column) in Figure (5.10)

5.1.4 Influence of the CFL Number

It has been stated in Chapter 3 that the CFL number α is specified as 0.5 in this study. To investigate the influence of the CFL number, we now increase it to the critical value 1 and the time step size becomes

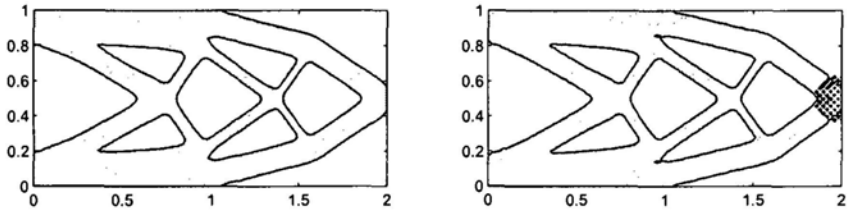
$$\Delta t = h/\max\{V_{ext}\}.$$

Two FELSM schemes, NG2lump and NG2cons, and four FDLSM schemes are tested with this CFL number. Figure (5.12) shows final results on the structured mesh. The result obtained by NG2lump is

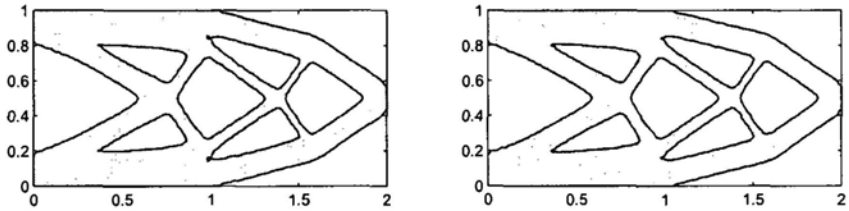
very different with that by NG2cons (compare Figure (5.12(a)) with Figure (5.12(b))). The former is good and very similar to the result with $\alpha = 0.5$ while the latter has many fragments at the free end. The failure of NG2cons results from the instability when the critical CFL number is used. The scheme with lumped coefficient matrix turns out to be more robust than the scheme with consistent one. In this sense, it is more reliable to use the NG2lump scheme especially when the CFL number is near the critical value or there is a possibility that the time step size can not be accurately estimated according to the CFL condition. It is interesting that a similar statement about the lumped technique is given in [71], which reads that it has occasionally been found that lumped matrix can improve the accuracy of some problems because the lumping process introduces additional dissipation and this can help in canceling out numerical oscillation.

Figures (5.12(c)) to (5.12(f)) show the results by the FDLSM. We can see that only ENO3 gives a similar result as what is obtained when small CFL number is used, the others create little cracks near corners of interior holes and lead to swings on the boundaries.

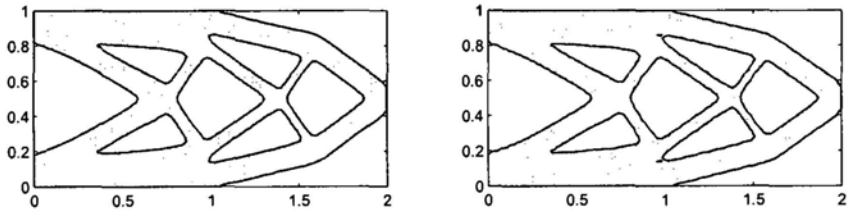
Figure (5.13) shows the convergence history of the objective function, when NG2lump is used with different CFL numbers. From Figure (5.13(a)), one can see that with $\alpha = 1$ the compliance converges faster since the time step size is larger. However, Figure (5.13(b)) shows that when α is large, the oscillation in the objective function is more drastic than that with the smaller CFL number. In addition to the risk of instability, an optimization process with a large time step size might miss some useful intermediate topologies. Therefore, in this work, a conservative choice of α (0.5) is used.



(a) NG2lump (b) NG2cons

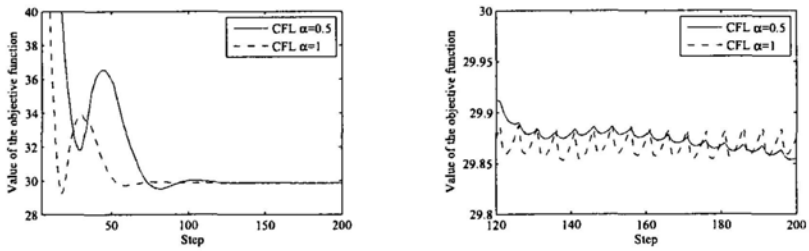


(c) ENO1 (d) ENO2



(e) ENO3 (f) WENO

Figure 5.12: Final designs of the cantilever beam when the CFL number is 1.



(a) Step 5 to 200 (b) Step 120 to 200

Figure 5.13: Convergence history of the cantilever beam with different CFL numbers (NG2lump).

5.2 A Michell-Type Structure

5.2.1 Single Load

The classical optimization problem of the Michell-type structure is considered. The model is shown as in Figure (5.14). The design domain is a rectangle with $L = 1$ and $H = 1$. Two bottom corners are constrained with pin supports. A concentrated load $P = 2$ is applied in the middle of the bottom. The limit of the volume fraction is 0.3. Because of the symmetry, only the right half part is analyzed, so the design domain is actually a unit square. The initial design for this problem is shown in Figure (5.15).

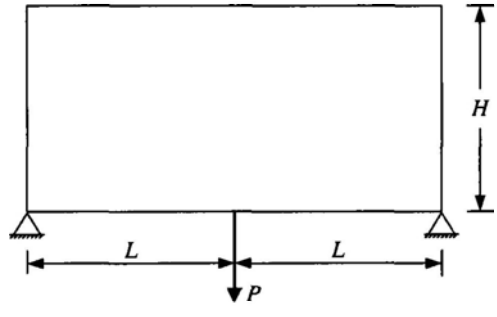


Figure 5.14: A Michell-type structure

First, a structured mesh with 50 by 50 uniform quadrilateral elements is used. Three schemes are implemented. They are WENO, NG2lump, and NG2cons. Figure (5.16) shows the final designs obtained with these schemes. A whole optimal structure is constructed based on the results by NG2lump and is shown in Figure (5.17). The final structure consists of two 45° arms extending from the supports towards an approximately 90° central fan section which extends upwards from the point of application of the force. This design is quite similar

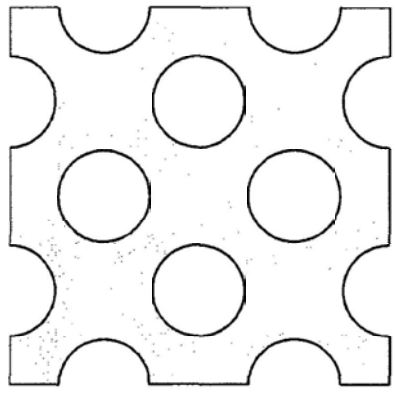
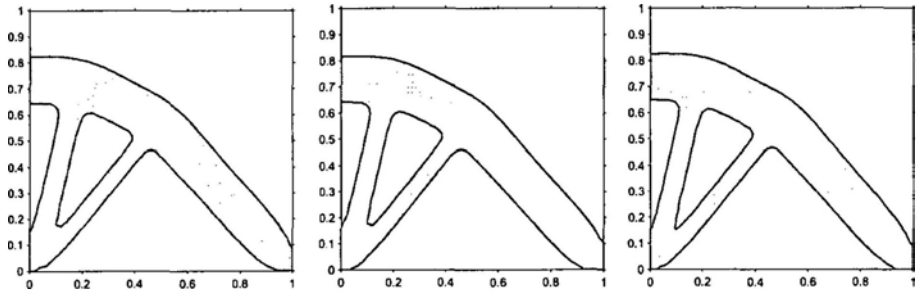


Figure 5.15: The initial design of the Michell-type structure.

to the theoretical optimum structure. The convergence history of the objective function and the volume fraction are shown in Figure(5.18). Although the objective function increases in the early steps due to the decline in the volume to satisfy the volume constraint, as shown in Figure (5.18(a)), it finally converges in a smooth and stable way.



(a) WENO (b) NG2lump (c) NG2cons

Figure 5.16: Final design of the Michell-type structure on uniform mesh.

Next, a free mesh as shown in Figure (5.19) is used. This mesh contains 2490 Q4 elements and 2591 nodes. The whole final design obtained by NG2lump after 153 steps on this mesh is shown in Figure (5.20), which is almost the same as the design in Figure (5.17). Ap-

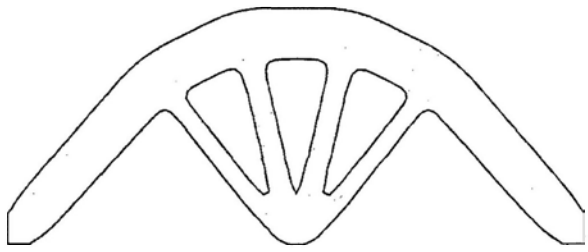
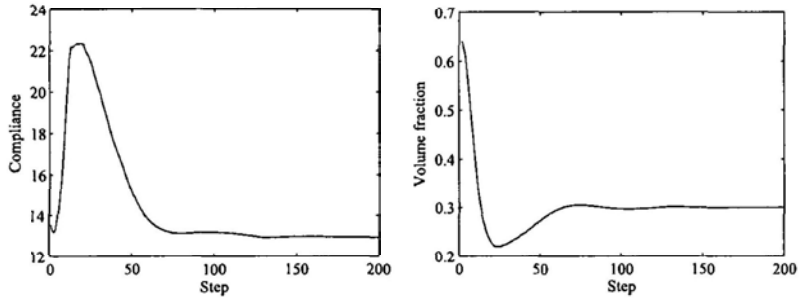


Figure 5.17: The whole final design of the Michell-type structure on the uniform mesh (by NG2lump).



(a) objective function (b) volume fraction

Figure 5.18: Convergence history of the Michell-type structure by NG2lump on the uniform mesh

plying the presented FELSM on unstructured meshes has no additional difficulties and this feature facilitates the optimization on irregular design domains as will be shown in following sections. The convergence history of the objective function and the volume fraction is shown in Figure (5.21).

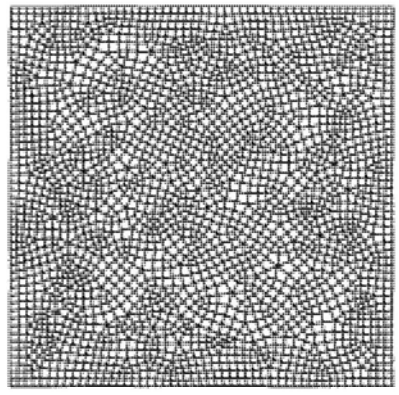


Figure 5.19: A free quadrilateral mesh for the Michell-type structure.

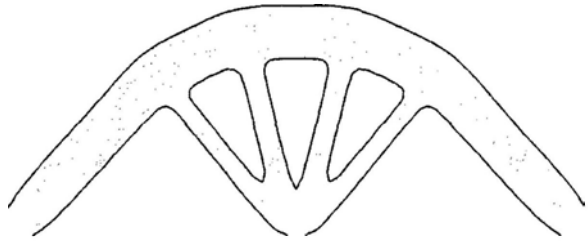
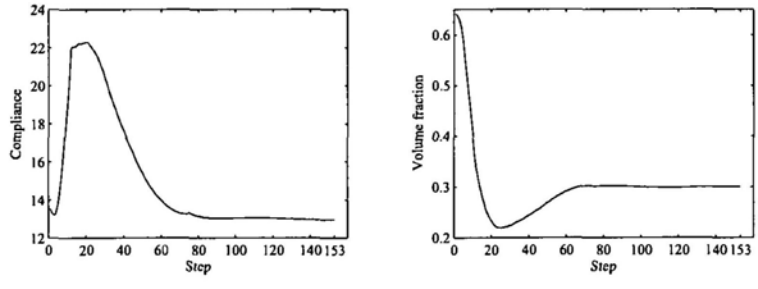


Figure 5.20: The whole final design of the Michell-type structure on the free mesh (by NG2lump).



(a) objective function (b) volume fraction

Figure 5.21: Convergence history of the Michell-type structure by NG2lump on the free mesh

5.2.2 Multiple Loads

The same structure is optimized where three loads are applied on the bottom side. Figure (5.22) shows the positions of these loads, whose magnitudes are $P = 2$ and $Q = 1$. WENO, NG2lump, and NG2cons are used to optimize this model on the uniform mesh used in single load case. The final results shown in Figure (5.23) are almost the same. A whole structure is recovered from the result by NG2lump and is displayed in Figure (5.24).

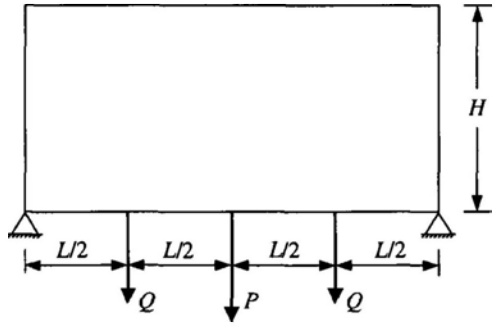
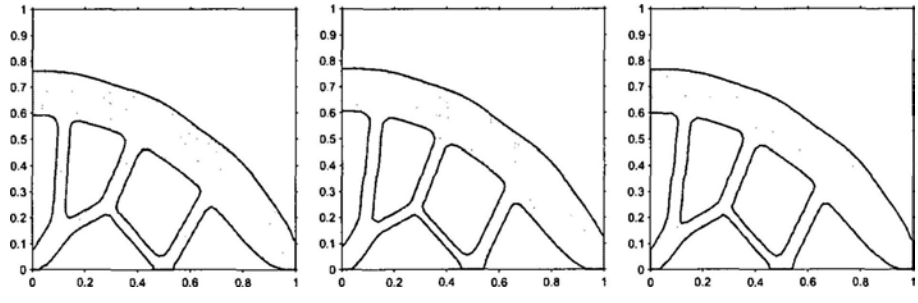


Figure 5.22: A Michell-type structure with multiple loads.



(a) WENO (b) NG2lump (c) NG2cons

Figure 5.23: Final design of the Michell-type structure under multiple loads.

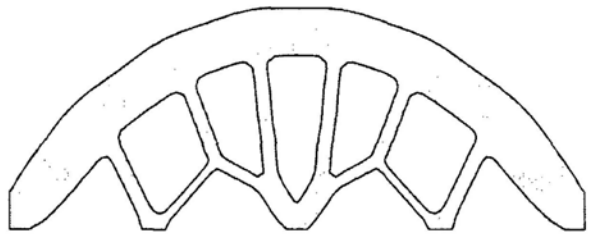


Figure 5.24: The whole final design of the Michell-type structure under multiple loads (by NG2lump).

5.3 A Cantilever Beam with a Fixed Hole

A optimization problem with an irregular design domain is considered in this example. This problem has been studies in [79, 80]. As shown in Figure (5.25), the dimensions of the cantilever beam are: $L = 9$ and $H = 6$. The radius of the hole is 2 and its center is determined by $W = 3$ and $D = 3$. A vertical concentrated load $P = 1$ is applied at the bottom right corner. The maximum allowed volume is half of the volume of the design domain which is $54 - 4\pi$.

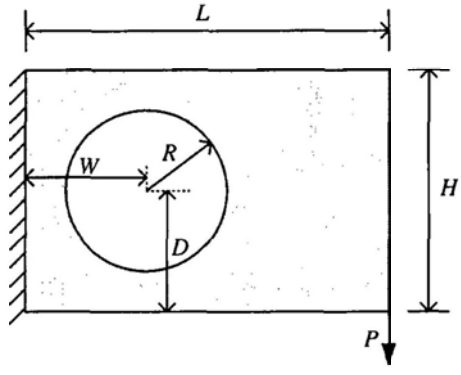


Figure 5.25: A cantilever beam with a fixed hole

In this problem, no design is allowed inside the hole. In practical structures, this hole might be created beforehand for operation or in-

stallation. Figure (5.26) shows the initial design of this problem. Only the NG2lump scheme is used in this example.

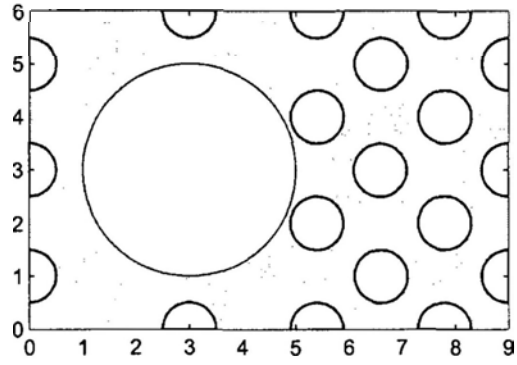


Figure 5.26: The initial design of the cantilever beam with a fixed hole.

First, a mapped mesh is considered. This mesh consists of 5400 quadrilateral elements and 5670 nodes. Because the right part of the structure is a regular domain, this part is actually meshed with uniform elements as shown in Figure (5.27). The optimization process converges after 110 steps. Figure (5.28) shows some intermediate designs and the final design. The topology and shape of the final design is very similar to what is reported in [79, 80]. Figure (5.29) shows the convergence history of this problem on the mapped mesh.

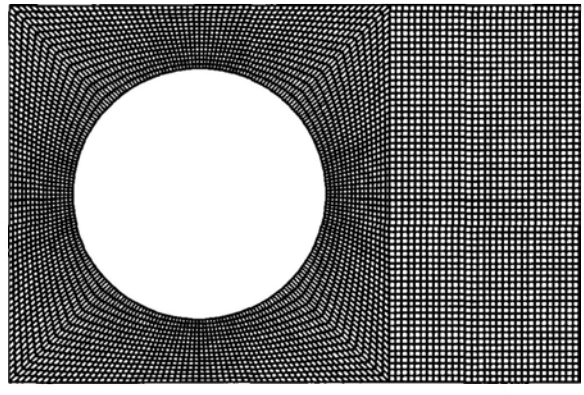
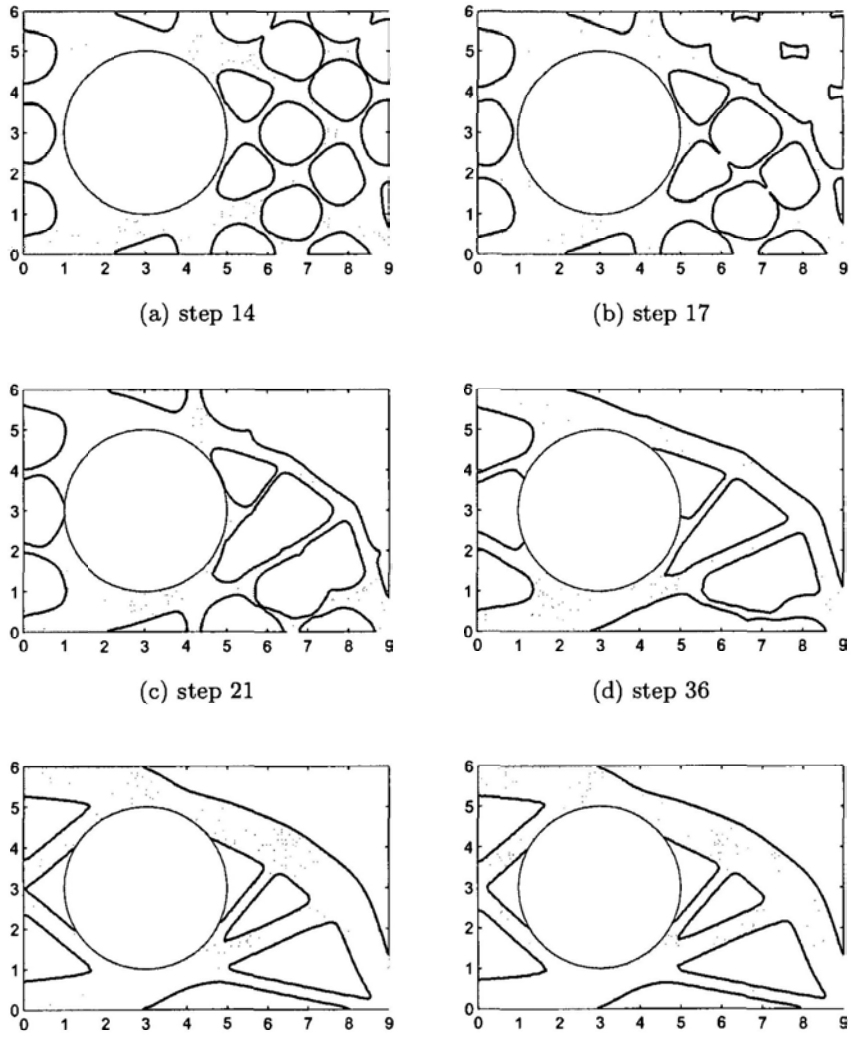


Figure 5.27: A mapped mesh for the cantilever beam with a fixed hole.



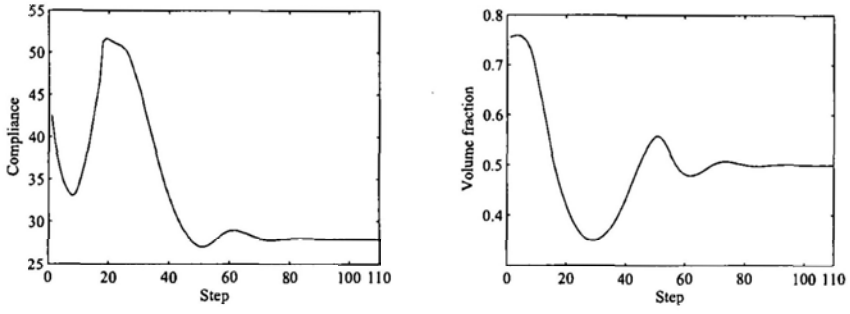
(a) step 14 (b) step 17

(c) step 21 (d) step 36

(e) step 89 (f) step 110 (final)

Figure 5.28: Intermediate designs of the cantilever beam with a fixed hole on the mapped mesh.

In Figure (5.30), we plot the zero isocontour of the extension velocity field as well as the boundary of the final design. It is shown that the zero isocontour almost coincides with the boundary except at some sharp corners of interior holes. This phenomena agrees with the theoretical prediction that normal velocity on the boundary of the optimal



(a) objective function (b) volume fraction

Figure 5.29: Convergence history of the cantilever beam with a fixed hole on the mapped mesh.

structure is zero. It is noteworthy that there are two segments of zero isocontour at the top and bottom of the fixed circle, which means that there should be two segments of boundary. However, boundaries can not emerge in this region because no hole is pre-collocated here in the initial design shown in Figure (5.26) and current method has no nucleate mechanism. Now, two holes are punched on the final design shown in Figure (5.28(f)) and the obtained structure as shown in Figure (5.31) is used as a new initial design for a new optimization process. After 15 steps, a new final design is obtained and the boundary as well as the zero isocontour of velocity are shown in Figure (5.32). The newly added holes evolve to two segments of boundary which coincide with zero isocontour. The compliance of the design in Figure (5.28(f)) is 27.83 and that in Figure (5.32) is 27.77. The decrease in compliance is not significant, but the new design is much closer to a truss structure than the design in Figure (5.28(f)).

Next, the design domain is discretized with a free mesh containing 3913 irregular quadrilateral elements and 4127 nodes. This mesh is

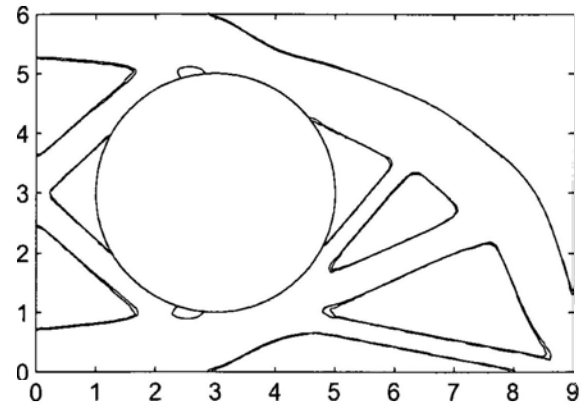


Figure 5.30: Zero isocontour of the velocity field (red line) and the boundary (black line).

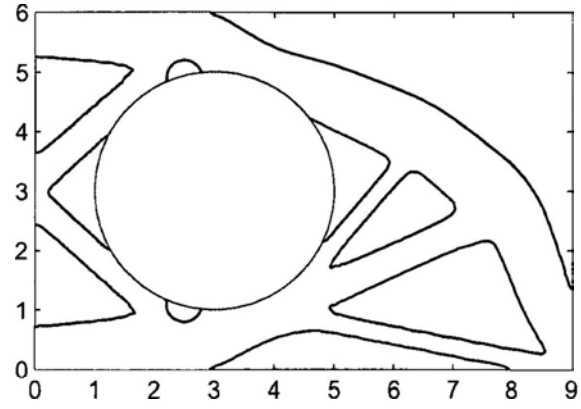


Figure 5.31: New initial design by adding holes to the final design.

illustrated in Figure (5.33). Intermediate designs on the same steps as in Figure (5.28(a)) to (5.28(e)) are shown in Figure (5.34) and the convergence history are shown in Figure (5.35). On this free quadrilateral mesh, the optimization process runs up to 200 steps. In this and the next cases, we don't plot the zero isocontour of the velocity field of the final design as in Figure (5.30). However, the same phenomena exist if we perform that analysis.

At last, a free triangular mesh as shown in Figure (5.36) is used.

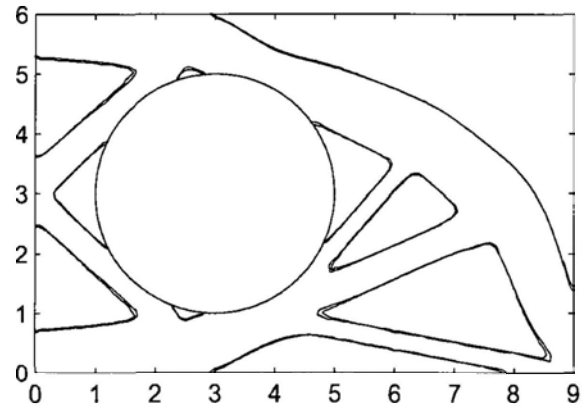


Figure 5.32: New final design: zero isocontour of the velocity field (red line) and the boundary (black line).

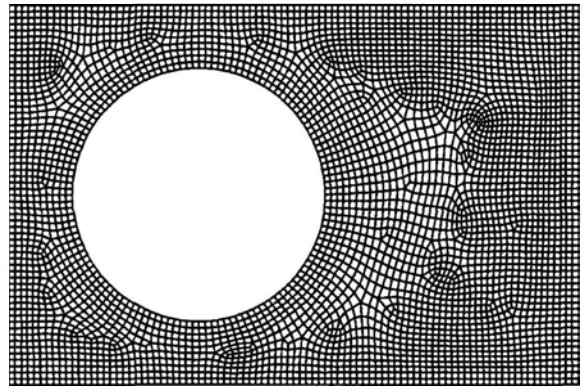
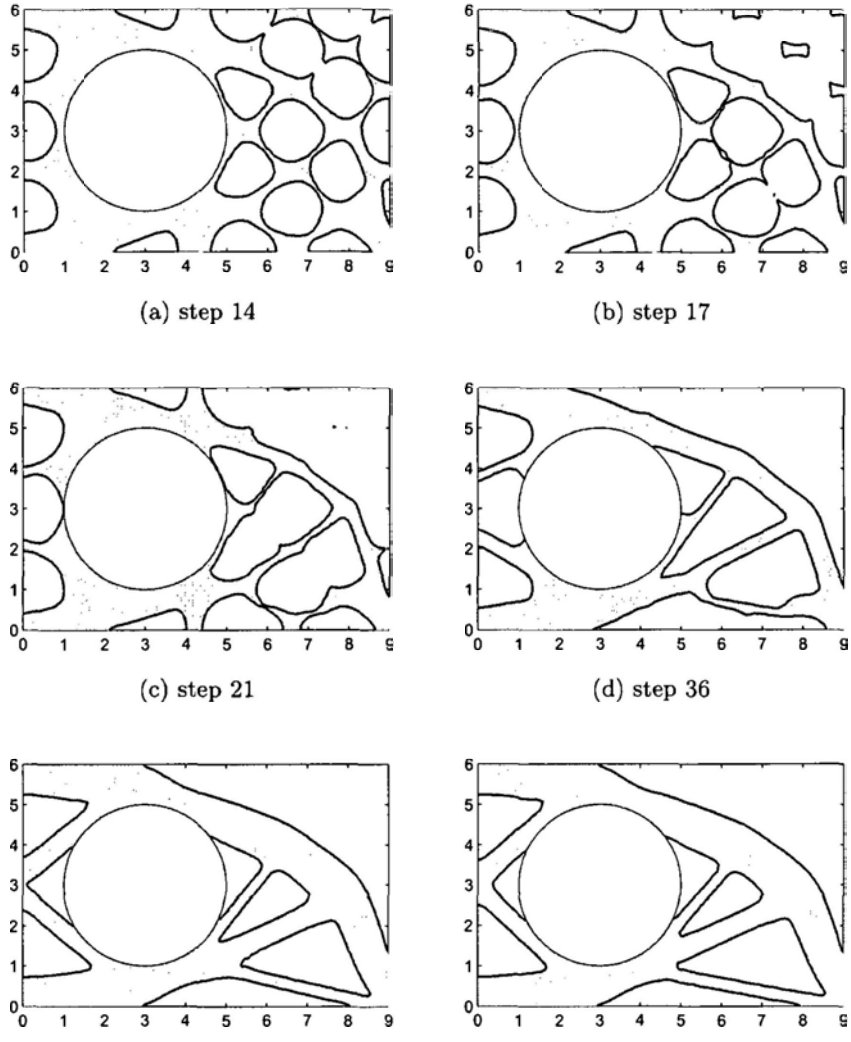


Figure 5.33: A free Q4 mesh for the cantilever beam with a fixed hole.

There are 4030 three-node triangular elements and 2157 nodes. Some intermediate designs and the final design after 200 steps are exhibited in Figure (5.37). Similar to previous results on triangular meshes, there are oscillations on the boundary, which shows the shortcoming of this type of elements. The convergence history of the objective function and the volume fraction is shown in Figure (5.38).



(a) step 14

(b) step 17

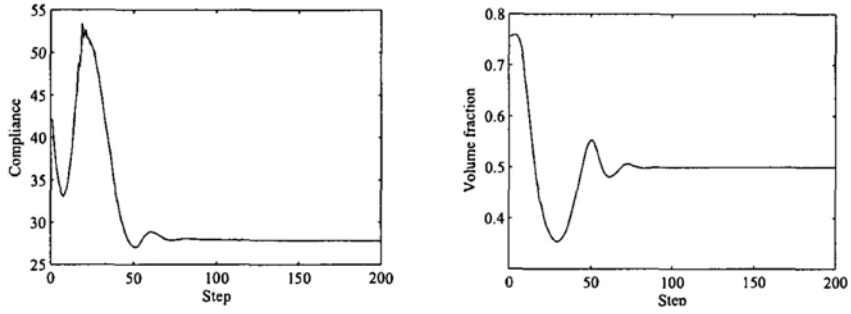
(c) step 21

(d) step 36

(e) step 89

(f) step 200 (final)

Figure 5.34: Intermediate designs of the cantilever beam with a fixed hole on the free quadrilateral mesh.



(a) objective function (b) volume fraction

Figure 5.35: Convergence history of the cantilever beam with a fixed hole on the free quadrilateral mesh.

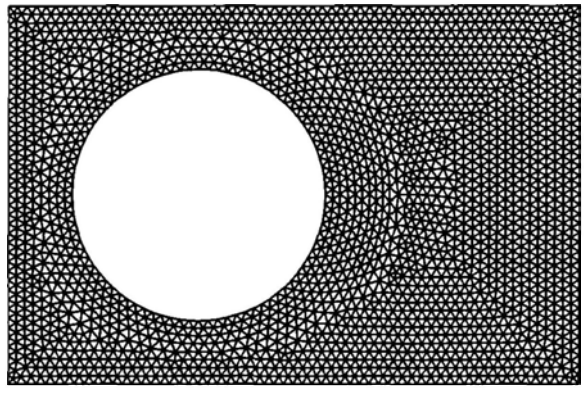
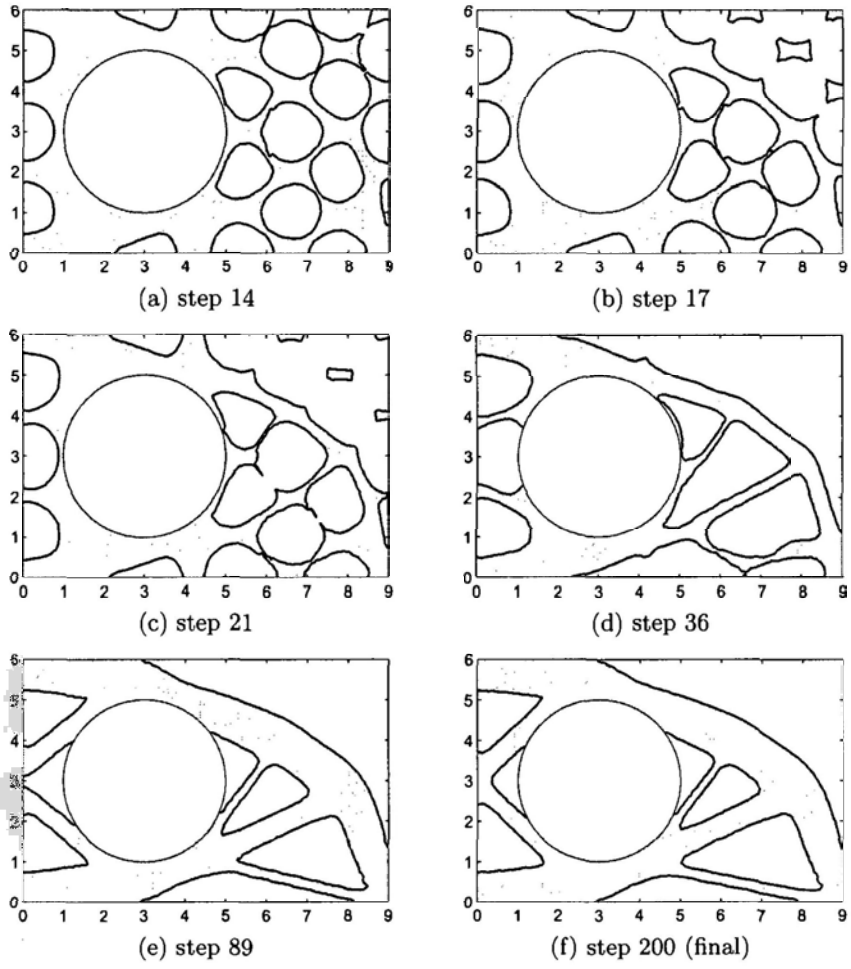


Figure 5.36: A free triangular mesh for the cantilever beam with a fixed hole.

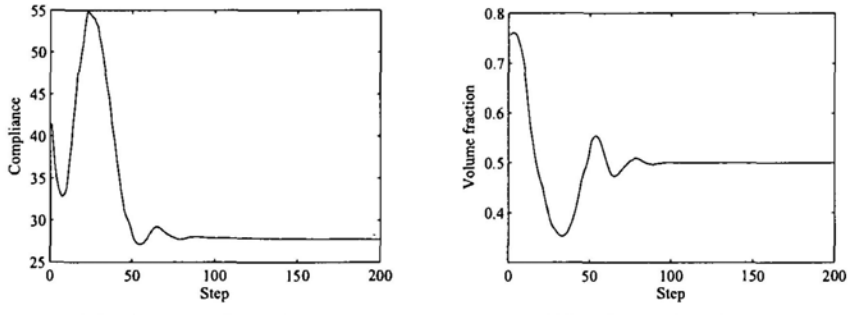


(a) step 14 (b) step 17

(c) step 21 (d) step 36

(e) step 89 (f) step 200 (final)

Figure 5.37: Intermediate designs of the cantilever beam with a fixed hole on the free triangular mesh.



(a) objective function (b) volume fraction

Figure 5.38: Convergence history of the cantilever beam with a fixed hole on the free triangular mesh.

5.4 A Michell Structure with a Semicircular Support

This example is related to a Michell-type structure as shown in Figure (5.39). The structure is fixed on a semicircular support (the bold curve) and a concentrated load $P = 1$ acts on the middle of the right edge. Geometric sizes of the design domain are: $L = 4$, $H = 1$, and $R = 1$. This problem is previously studied in [2,81]. The initial design is shown in Figure (5.40).

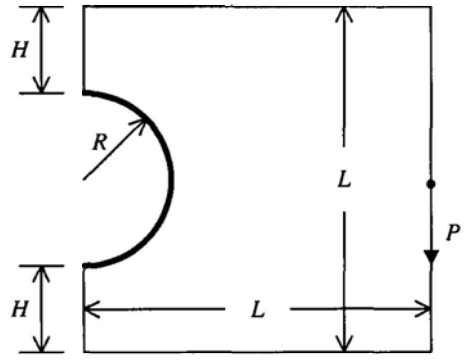


Figure 5.39: A Michell structure with a semicircular support.

5.4.1 Lower volume fraction

First, the maximum volume fraction is specified as 0.2. Two mapped meshes are used in this example. Mesh 1 is a coarse one that contains 1750 quadrilateral meshes and 1856 nodes as shown in Figure (5.41(a)) and mesh 2 is a fine one with 4480 elements and 4649 nodes as shown in Figure (5.41(b)). Final designs by NG2lump on both meshes are shown in Figure (5.42). They are very similar to each other.

Some intermediate designs are displayed in Figure (5.43). It can be seen that the evolution speed on mesh 1 is faster than that on mesh 2.

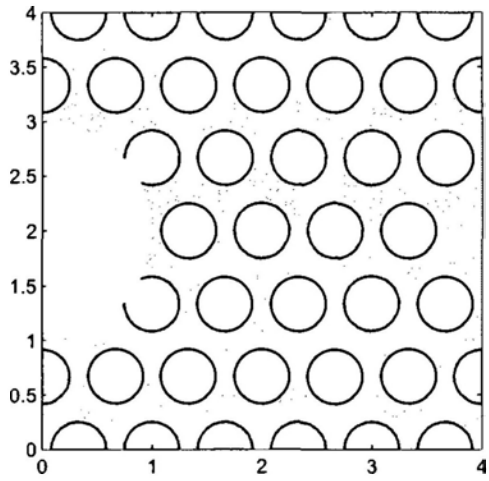
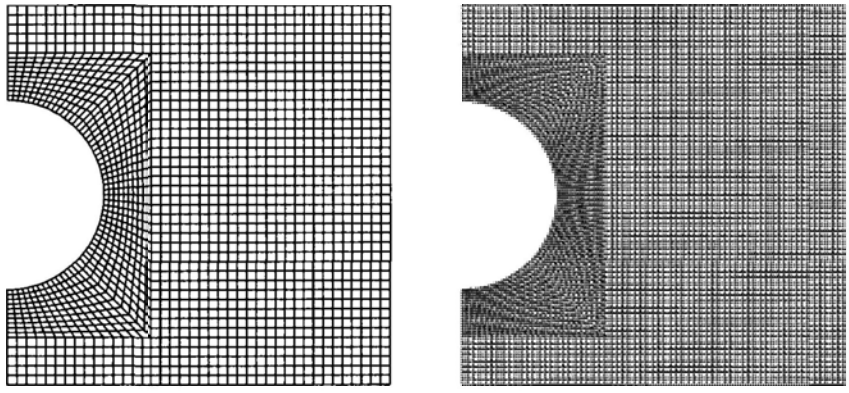


Figure 5.40: The initial design of the Michell structure with a semicircular support.

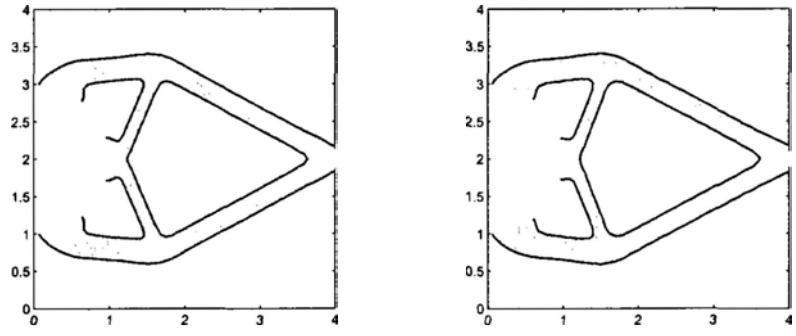


(a) mesh 1 (b) mesh 2

Figure 5.41: Two mapped meshes for the Michell structure with a semicircular support.

The reason for this phenomenon is that the time step size for mesh 1 is larger because the element size of mesh 1 is larger than that of mesh 2.

The convergence history on mesh 2 is shown in Figure (5.44). The bump in the objective function curve is due to a change of topology (bar elimination). When the bar is separated from the structure, the



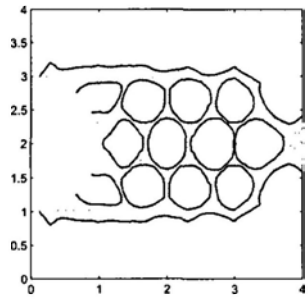
(a) mesh 1 (b) mesh 2

Figure 5.42: Final designs of the Michell structure with a semicircular support.

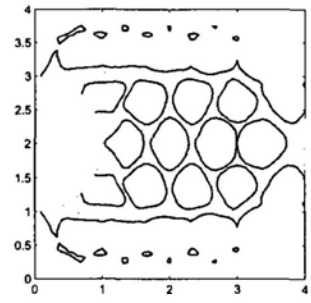
compliance increase rapidly. However, after several steps, the boundary is optimized and the compliance decreases gradually. Figure (5.45) demonstrates this process, which is related to the bump at about step 94 in Figure (5.44(a)).

5.4.2 Higher volume fraction

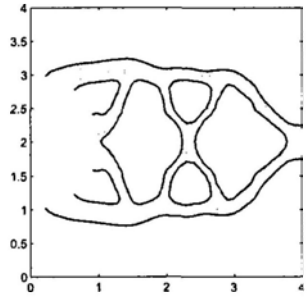
Now, the same problem is optimized subject to a higher volume fraction constraint. The limit of the volume fraction is specified as 0.5 this time. The finer mesh as shown in Figure (5.41(b)) is used. The optimization process converges after 172 steps with the NG2lump scheme. Figure (5.46) shows some intermediate designs and the final design. The final design in Figure (5.46(f)) is very different from the final design at lower volume fraction as shown in Figure (5.42(b)). The topology (number and positions of holes) at higher volume fraction is similar to the topology of the optimal cantilever beam in section 5.1. The convergence history is illustrated in Figure (5.47).



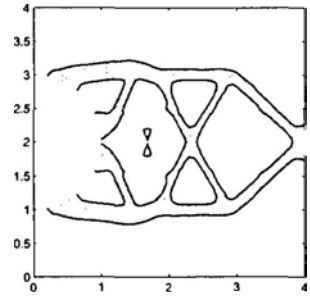
(a) mesh 1, step 15



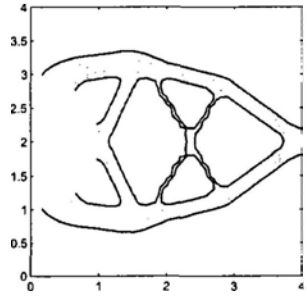
(b) mesh 2, step 15



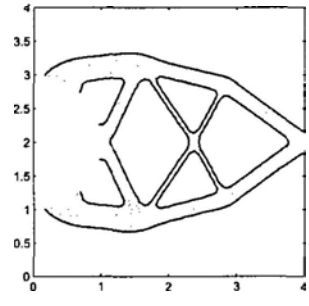
(c) mesh 1, step 27



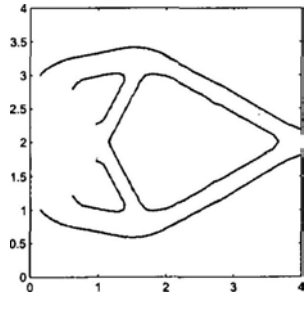
(d) mesh 2, step 27



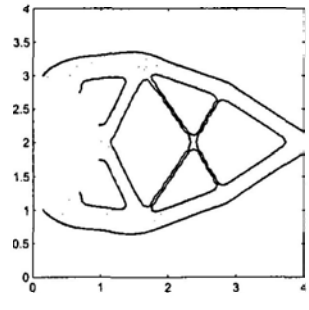
(e) mesh 1, step 70



(f) mesh 2, step 70

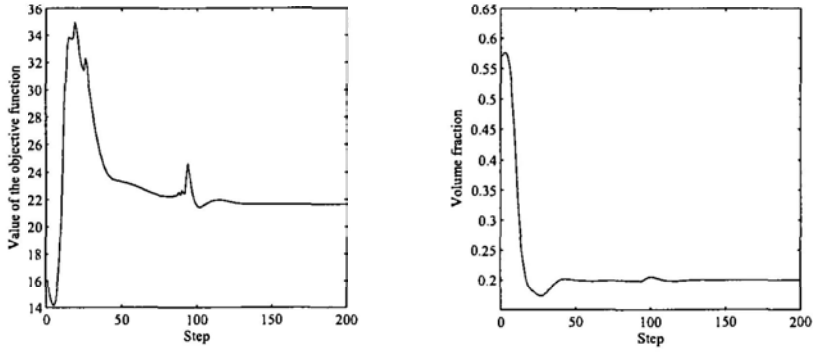


(g) mesh 1, step 88



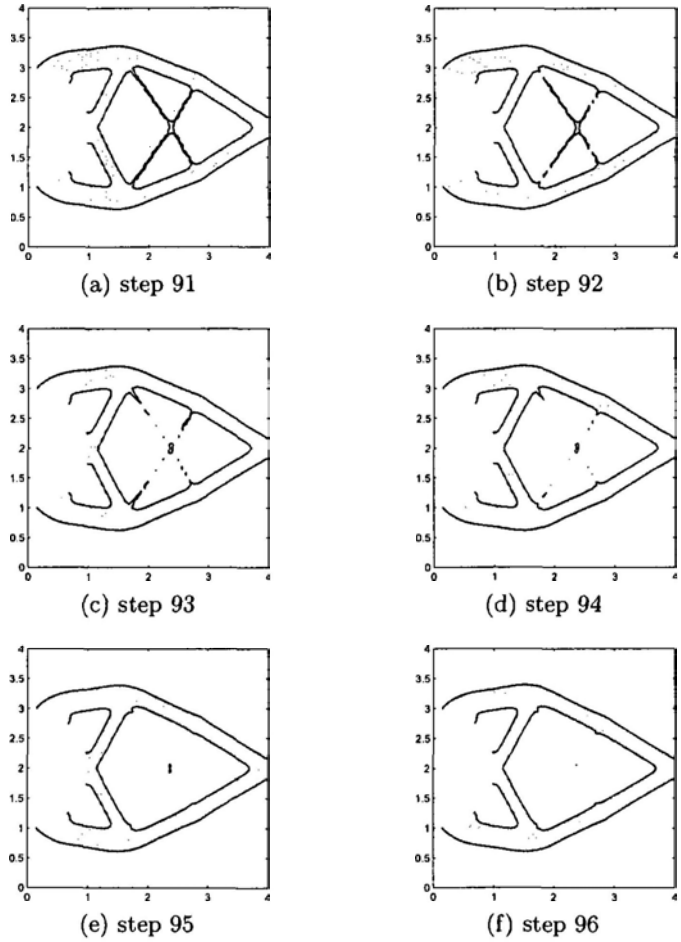
(h) mesh 2, step 88

Figure 5.43: Intermediate designs of the Michell structure with a semicircular support.



(a) Objective function (b) Volume fraction

Figure 5.44: Convergence history of the Michell structure with a semicircular support.

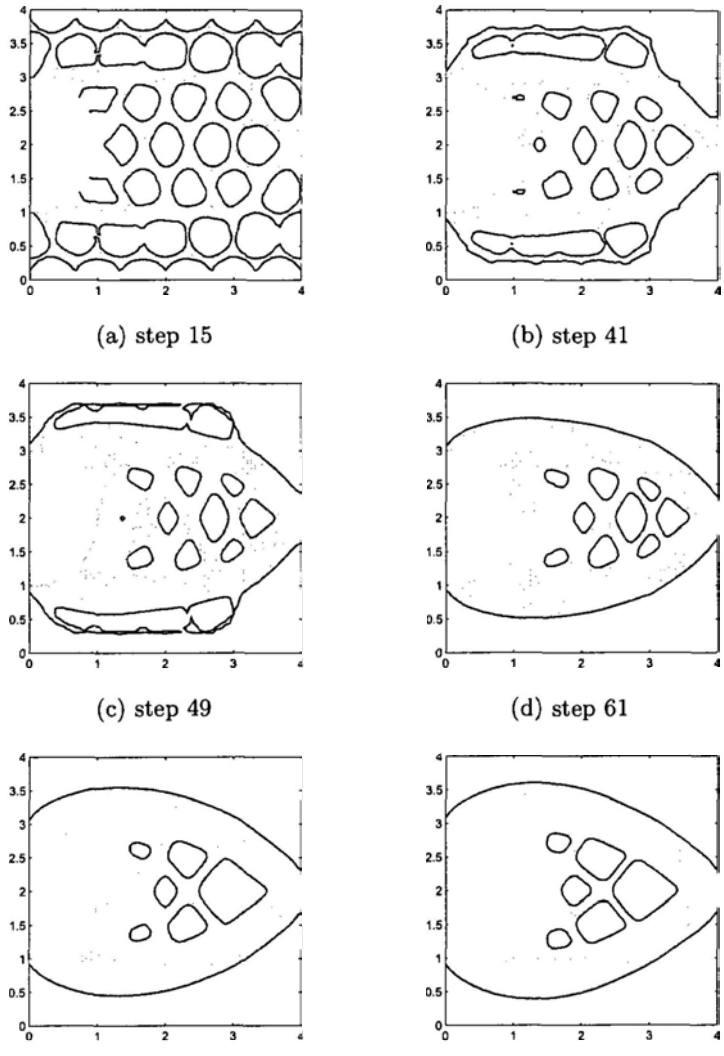


(a) step 91 (b) step 92

(c) step 93 (d) step 94

(e) step 95 (f) step 96

Figure 5.45: Topological change from step 91 to step 96.

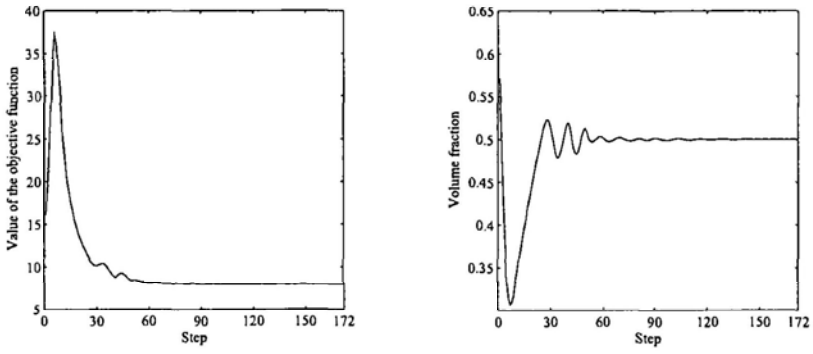


(a) step 15 (b) step 41

(c) step 49 (d) step 61

(e) step 90 (f) step 172 (final)

Figure 5.46: The final and intermediate designs of the Michell structure with a semicircular support at higher volume fraction.



(a) Objective function (b) Volume fraction

Figure 5.47: Convergence history of the Michell structure with a semicircular support at higher volume fraction.

□ **End of chapter.**

Chapter 6

Conclusions And Future Work

6.1 Conclusions

In this thesis, a finite element based level set method (FELSM) is employed in the structural topology optimization problems.

To discretize the level set equation and reinitialization equation on a finite element mesh, the SDFEM is used because it can produce stable numerical solutions of the advection dominated level set equations by introducing a diffusion term in the direction of the streamline. Formulations are derived in this thesis. For the discretized level set equation, the coefficient matrix is a constant, symmetric, and positive defined matrix. Furthermore, we find that it is quite similar to the mass matrix in structural dynamics and the lumping technique is used to replace the consistent coefficient matrix with a lumped one. With the lumped coefficient matrix, the cost in solving and storing is reduced remarkably. The accuracy of the FELSM is compared with the conventional finite

difference based level set method (FDLSM). It is shown through test cases that the accuracy of the FELSM is similar to the first-order finite difference scheme ENO1. Although higher-order FDLSM schemes, such as ENO2, ENO3, and WENO, are more accurate than the FELSM schemes in approximating the spatial derivatives, we demonstrate that higher-order accurate schemes fail to create higher-order accurate designs in the structural optimization problems because the low-order accurate velocity field is the bottleneck. In other words, the proposed FELSM is accurate enough to produce reliable optimal results for the minimum compliance design considered in this study.

In the discretized reinitialization equation, because of the insufficient stability effect, an extra diffusion term is added. We give the criterion for selecting the factor of the diffusion term. To prevent the boundary from moving during the reinitialization, we treat the fixing of the boundary as a Dirichlet boundary condition and enforce this boundary condition by both the Lagrangian multiplier method and the penalty method. Solving the system of reinitialization equations is more time consuming than solving the level set equation. Fortunately, the structural optimization problems are not very sensitive to the level set function's approximation to the signed distance function, hence it is not necessary to perform the reinitialization frequently and accurately. In this study, we carry out reinitialization every five steps and the numerical results show that the level set function is regular enough with this frequency of reinitialization.

Two velocity extension methods are introduced in this thesis. The modified natural extension method is natural and direct. It is a efficient and simple method for the whole domain scheme. Hence it is

used in the numerical examples in this study. The PDE-based velocity extension method has rigorous mathematical theory. However, solving the equation is not easy. Therefore, only formulations are derived and no implementation is performed in this work. Some related issues, such as the "ersatz" material approach, recovery of stresses, and the influence of stress concentrations and singularities, are also discussed.

Numerical examples, which involve regular and irregular domains, structured and unstructured meshes, are solved with the proposed method. Results illustrate the reliability of the method. In this study, only low-accurate elements are tested. It is shown that the four-node quadrilateral element works well. The three-node triangular element is not recommended due to the feature of constant strains. To calculate the right-hand-side term of the level set equation, two-point Gaussian quadrature scheme (for Q4 element) is necessary. The scheme using lumped coefficient matrix is recommended because it is quite efficient and, more important, it provides more stability effect than the scheme using the consistent coefficient matrix and schemes of FDLSM when the CFL number if large.

6.2 Future Work

There are lots of possibilities for future work of the presented method. The numerical examples clearly demonstrate the feasibility of this method in solving the minimum mean compliance problems. However, the performance of this method in frequencies optimization and stresses optimization problems is still needed to be investigated.

For the purpose of practical application, the presented method

should be extended to three dimensional problems. Although there is no theoretical difficulty, some numerical issues are still needed to be considered. Algorithms for conventional level set methods have been extensively researched for 3D problems, but finite element based level set methods are still in the initial stage.

In this study, only four-node quadrilateral elements and three-node triangular elements are used. In the future, high-order accurate elements will be studied. It is apparent that the high-order elements can improve the accuracy of velocity computation. However, the performance of high-order elements in the level set stage is still needed to be verified.

The implementation of the narrow band scheme is also a remarkable topic. By virtue of the lumped coefficient matrix, the system of level set equations are decoupled completely and equation on each node can be solved independently. This might make the narrow band scheme more attractive.

□ **End of chapter.**

Bibliography

[1] 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.

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

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

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

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

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

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

[8] 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.

[9] 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.

[10] 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.

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

[12] Stanley J. Osher and Fadil 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Michael Wang and Shengyin 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.

[20] 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.

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

[22] Zhen Luo, Michael Yu Wang, Shengyin Wang, and Peng Wei. A level set-based parameterization method for structural shape and topology optimization. *Int. J. Numer. Methods Eng.*, 76(1):1–26, 2008.

[23] 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.

[24] P. Wei and M. Y. Wang. Piecewise constant level set method for structural topology optimization. *Int. J. Numer. Methods Eng.*, accepted, August 2008.

[25] O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics.* Elsevier Butterworth-Heinemann, 2005.

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

[27] Peter Hansbo. Explicit streamline diffusion finite element methods for the compressible euler equations in conservation variables. *J. Comput. Phys.*, 109(2):274–288, 1993.

[28] A. Gravouil, N. Mos, and T. Belytschko. Non-planar 3d crack growth by the extended finite element and level sets-part ii: Level set update. *Int. J. Numer. Methods Eng.*, 53(11):2569–2586, 2002.

[29] Sven Groß, Volker Reichelt, and Arnold Reusken. A finite element based level set method for two-phase incompressible flows. *Comput. Visual. Sci.*, 9(4):239–257, 2006.

[30] Anna-Karin Tornberg and Björn Engquist. A finite element based level-set method for multiphase flow applications. *Comput. Visual. Sci.*, 3(1):93–101, May 2000.

[31] M. Weber, A. Blake, and R. Cipolla. Sparse finite elements for geodesic contours with level-sets. In *Proceedings of the 8th European Conference on Computer Vision (ECCV'04)*, number 3022 in Lecture Notes in Computer Science, pages 391–404. Springer-Verlag, Berlin, Germany, 2004.

[32] A.N. Brooks and T.J.R. Hughes. Streamline upwind/petrov-galerkin methods for advection-dominated flows. In *Proceedings Third International Conference on Finite Element in Fluid Flows*, Banff, Canada, 1980.

[33] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, New York, 1988.

[34] Thomas J. R. Hughes, Leopoldo P. Franca, and Gregory M. Hulbert. A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-

diffusive equations. *Comput. Methods Appl. Mech. Engrg.*, 73(2):173–189, May 1989.

[35] Jack Chessa, Patrick Smolinski, and Ted Belytschko. The extended finite element method (xfem) for solidification problems. *Int. J. Numer. Methods Eng.*, 53(8):1959–1977, 2002.

[36] Vinay S. Rao, Thomas J. R. Hughes, and Krishna Garikipati. On modelling thermal oxidation of silicon ii: numerical aspects. *Int. J. Numer. Methods Eng.*, 47(1-3):359–377, 2000.

[37] Stphane Valance, Ren de Borst, Julien Rthor, and Michel Coret. A partition-of-unity-based finite element method for level sets. *Int. J. Numer. Methods Eng.*, 76(10):1513–1527, 2008.

[38] B. Cockburn and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16:173–261, 2001.

[39] Daniele A. Di Pietro, Stefania Lo Forte, and Nicola Parolini. Mass preserving finite element implementations of the level set method. *Appl. Numer. Math.*, 56(9):1179–1195, 2006.

[40] Emilie Marchandise, Jean-François Remacle, and Nicolas Chevaugeon. A quadrature-free discontinuous galerkin method for the level set equation. *J. Comput. Phys.*, 212(1):338–357, 2006.

[41] B. N. Jiang and Graham F. Carey. A stable least-squares finite element method for non-linear hyperbolic problems. *Int. J. Numer. Methods Fluids*, 8(8):933–942, 1988.

[42] Bo Nan Jiang. *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics.* Springer, 1998.

[43] Kang-Soo Park and Sung-Kie Youn. Topology optimization of shell structures using adaptive inner-front (aif) level set method. *Struct. Multidisc. Optim.*, 36(1):43–58, July 2008.

[44] Kang-Soo Park and Sung-Kie Youn. Topology optimization of cone type shell structure employing adaptive inner-front level set method (aiflsm). In *proceeding of 7th World Congress on Structural and Multidisciplinary Optimization(WCSMO-7)*, pages 1977–1986, 2007.

[45] J. Chessa and T. Belytschko. An extended finite element method for two-phase fluids. *J. Appl. Mech.*, 70(1):10–17, 2003.

[46] Ching-Long Lin, Haegyun Lee, Taehun Lee, and Larry J. Weber. A level set characteristic galerkin finite element method for free surface flows. *Int. J. Numer. Methods Fluids*, 49(5):521–547, 2005.

[47] M. Quecedo and M. Pastor. Application of the level set method to the finite element solution of two-phase flows. *Int. J. Numer. Methods Eng.*, 50(3):645–663, 2001.

[48] Anton Smolianski. Finite-element/level-set/operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces. *Int. J. Numer. Methods Fluids*, 48(3):231–269, 2005.

[49] Laurent Bourgouin, Hans-Bernd Mühlhaus, Alina Hale, and Antonin Arsac. Towards realistic simulations of lava dome growth using the level set method. *Acta Geotechnica*, 1(4):225–236, December 2006.

[50] Z. Liu, J.G. Korvink, and R. Huang. Structure topology optimization: fully coupled level set method via femlab. *Struct. Multidisc. Optim.*, 29(6):407–417, June 2005.

[51] http://www.femlab.com.

[52] Hashem M. Mourad, John Dolbow, and Krishna Garikipati. An assumed-gradient finite element method for the level set equation. *Int. J. Numer. Methods Eng.*, 64(8):1009–1032, 2005.

[53] Shintaro Yamasaki, Shinji Nishiwaki, Kazuhiro Izui, and Masataka Yoshimura. Eigen-frequency optimization method, based on the level set method using a geometric re-initialization scheme. In *Proceedings of APCOM07 - EPMESC XI*, December 2007.

[54] S. Yamasaki, S. Nishiwaki, K. Izui, and M. Yoshimura. A structural optimization method for stiffness and vibration problems, based on the level set method using a new geometric re-initialization scheme. In *proceeding of 7th World Congress on Structural and Multidisciplinary Optimization(WCSMO-7)*, pages 1937–1946, 2007.

[55] Ashok D. Belegundu and Tirupathi R. Chandrupatla. *Optimization Concepts And Applications In Engineering.* Prentice Hall, 1 edition, April 1999.

[56] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, 1999.

[57] 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.

[58] 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.

[59] 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.

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

[61] 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.

[62] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A pde-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[63] 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.

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

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

[66] Xu-Dong Liu, Stanley Osher, and Tony 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] Alexander N. Brooks and Thomas J. R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, September 1982.

[70] I. E. Tezduyar and D. K. Ganjoo. Petrov-galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion prob-

lems. *Comput. Methods Appl. Mech. Engrg.*, 59(1):49–71, November 1986.

[71] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals.* Elsevier, 2005.

[72] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Prentice-Hall, 1987.

[73] http://www.algor.com.

[74] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148(1):2–22, 1999.

[75] P. Wei and M. Y. Wang. A structural optimization method with XFEM and level set models. In *Proc. of 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE2008)*, Izmir, Turkey, 2008.

[76] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery (spr) and adaptive finite element refinement. *Comput. Methods Appl. Mech. Engrg.*, 101(1-3):207–224, 1992.

[77] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.

[78] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving stefan problems. *J. Comput. Phys.*, 135(1):8–29, 1997.

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

[80] 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.

[81] X. M. Wang, M. Y. Wang, and D. M. Guo. Structural shape and topology optimization in a level-set-based framework of region representation. *Struct. Multidisc. Optim.*, 27:1–19, 2004.