Shape and Topology Optimization with Parametric Level Set Method and Partition of Unity Method

HO, Hon Shan

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

The Chinese University of Hong Kong
August 2010

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

Abstract of thesis entitled:

Shape and Topology Optimization with Parametric Level Set Method and Partition of Unity Method

Submitted by HO, Hon Shan

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in June 2010

Traditional structural optimization approaches can be referred to as sizing optimization, since their design variables are the proportions of the structure or material. A major restriction in the sizing problem is that the shape and the topology of the structure are fixed a priori. Undoubtedly, changes in shape (e.g., curved boundary) and topology (e.g., holes in a member) could produce more significant improvement in dynamic performance than modifications in size alone. A recent development of shape and topology optimization based on the implicit moving boundaries with the use of the renowned level set method is regarded as one of the most sophisticated methods in handling the change of the structural topology. In this thesis, we study the parametrization of the classical level set method for the structural optimization and the associated computational methodology.

First of all, the PDE form of the classical level set function Φ is parameterized with an analytical form of *Radial Basis Function* (RBF), which is real-valued and continuously differentiable. Such that the upwind scheme, extension velocity and reinitialization algorithms in

solving the discrete Hamilton-Jacobi equation can be waived in the numerical process, the whole framework is transformed into a standard mathematical programming problem in which the linear objective function can be directly optimized by a gradient algorithm - shape sensitivity. The minimization of the mean compliance is studied and presented to demonstrate the advantages of the parametrical method.

Parametrization substantially reduces the complexity of the original discrete PDE level set method. However, the result shows that the high number of RBF knots leads to dense coefficient matrices. Thus, it induces numerical instabilities, slow convergence and less accuracy in the process. Consequently, we then study the distribution of knots density for faster computation. By updating the movement of the knot, the knot moves towards the position where the change is directly determined by the shape sensitivity. In such case, we may use lesser number of knots to describe the properties of the system while the smoothness of the implicit function is satisfied. The sensitivity study is evaluated carefully and discussed in detail. Results show a significant improvement in the computational speed and stability.

Usually, a large-scale model will lead to bulk coefficient matrices in the RBF optimization and the linear function normally require $\mathcal{O}(N^3)$ flops and $\mathcal{O}(N^2)$ memory while processing. It is becoming impractical to solve as N goes over 10,000. In fact, the dense system equation matrix frequently leads to the numerical instabilities and the failure of the optimization. Finally, we introduce the method of *Partition of Unity* (POU) to deal with this problem. POU is often used in 3D reconstruction of implicit surfaces from scattered point sets. It breaks the global domain into smaller overlapping subdomains such that the implicit functions can be more efficiently interpolated. Meanwhile, the

global solution is obtained by blending all the local solutions with a set of weighting functions. The algorithm of POU is presented here, and we analyze and discuss the numerical results accordingly.

The study found significant improvement obtained in the structural optimization with the parametric level set method, both the stability and efficiency were given as the benefits of using the method of the parametrization.

摘要

傳統的結構優化方法可以稱為尺寸優化，因為他們的設計 變量的比例結構或材料。一個主要的限制，漿紗的問題是，形狀和拓撲結構是固定 的先驗。毫無疑問，改變形狀（例如，曲線邊界）和拓撲結構（例如，在一洞）可以顯著改善，生產更多的動力性能比單獨修改的大小。最近開發的形狀和拓撲優化基於隱含邊界移動與使用著名 的水平集方法被認為是一個最複雜的方法，在處理拓撲結構的變化。在這篇論文中，我們研究了參數化的經典水平集方法的結 構優化和相關的計算方法。

首先，在偏微分方程形式的經典水平集函數是參數化的分析與形式徑向基函數（RBF）神經網絡，這是實值和連續可微的。這樣的迎風格式，擴展速度和重新初始化算法在求解離散 的 Hamilton - Jacobi 方程可免除數值的過程中，整個架構轉化為一個標準的數學規劃問題，其中線性目標函數可以直接優化梯度算法 - 形狀的敏感性。最小化的平均符合研究，並提交證明的參數化方法的優 點。

參數化的複雜性大大降低了原離散 PDE 的水平集方法。但是，結果表明，大量的 RBF 海裡導致密集係數矩陣。因此，它誘使數值不穩定性，收斂速度慢，精度低的過 程。因此，我們又研究了節密度分佈的計算速度更快。通過更新運動中的結，結的位置，走向的變化直接決定了 形狀的敏感性。在這種情況下，我們可能會使用較少數節描述的屬性系 統，而平滑的隱函數是滿意的。的敏感性研究仔細評估和詳細討論。結果顯示，顯著改善了計算速度和穩定性。

通常，一個大規模的模式將導致大批係數矩陣中的 RBF 優化和線性函數通常需要觸發器和內存而處理。它正在成為不切實際的解決為它去。事實上，茂密的系統方程矩陣往往會導致數值不穩定和破壞的優化。最後，我們介紹的方法對(時間單位分解) POU 結構來處理這個問題。POU 結構常用於三維隱式曲面重建散亂點集。它打破了全球域名成更小的子域重疊，這樣的隱函數可以 更有效地插。與此同時，全球性解決方案得到了混合所有的局部解與一 組權函數。該算法的 POU 是這裡，我們分析並討論相應數值結果。

研究發現，取得了顯著的改善與優化結構參數水平集方法，無論是穩定和效率均給予的好處使用方法的參數化。

Acknowledgement

I would like take this opportunity to express my sincere gratitude to my supervisor, Professor Michael Yu Wang, for his patient guidance, helpful suggestions and generous support during these years. His great insight and wide knowledge in this field give the good idea to begin this work and to enable it on the right track to the finish line. It is my great honor to be one of his students.

I would like to extend my thanks to all members of the Computational Modeling and Design Laboratory at the Chinese University of Hong Kong. They are Dr. XIA Qi, Dr. WEI Peng, Dr. XING Xiang Hua, Dr. XIA Hong Jian, Ms. LUI Fung Yee and ZHANG Ji Wei.

This work is dedicated to Amy, Brooklyn and my family for their endless support and care. Also, I hope this can be read by my Dad who is living in God's land.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

An engineer working in the field of research and development is often found in the position of having to design a completely new structure. The conditions such as loads and constraints are usually known but the designer has the difficult certainty what is the aspect of the structure that makes his case right. The weight is known to be one of the main factors that contribute to the performance and the cost of the building, and for this reason the reduction of it is very frequently the key objective to be achieved. There are similar cases previously addressed but it is unlikely that a downsizing of the previous design gives concrete benefits. Otherwise, one can appeal to intuition and experience of the designer, but this approach can become long and exhausting, that the success of the project is not guaranteed. Alternatively, the problem of the product design can be facilitated by structural optimization tools. Historically, the structural optimization problem is related to the development of those areas where the lightweight component is a prerequisite for the exercise, just think of civil construction, industries

such as aerospace and automobile. Structural optimization has become widely accepted in engineering design in the past decades, and is becoming more and more important as the essential competitive power in the industries. Besides market-driven force, the rapid development of powerful computers and the software plays an important role in popularizing the structural optimization techniques in these areas.

In 1904 Michell [10] studied the topological optimization problem of the lattice structures, the development of the techniques related to this problem had significantly grown only in the recent years and led to the definition of that are now well established rules and practices. Today, the achievements in the field well beyond the satisfaction of the mass reduction (objective far more frequent!) and the field meets ever wider applications such as the dynamics, acoustics, heat transmission or even the modern micro-electromechanical applications. Recent decades have witnessed a rapid spread of instrument optimizations so that they have become irreplaceable in automotive and aircraft and begin to make their appearance in the world of the competition sports (cycling) or even in the design of the joint replacements. It gave fresh impetus to research in the field of the structural optimization. On the other hand, the recent development of the mathematical models for the behaviors of the materials and the surge of the processing power in the personal computer benefit the popularization of the optimization tools. We will see that the huge majority of these techniques inevitably focuses on the use of finite element computer codes. The choices of size, shape and topology refer to different aspects of the structural problem. If applied, these concepts help to tied to the quality and performance of the design component.

Traditional structural optimization approaches can be referred to as

sizing optimization, since their design variables are the proportions of the structure or material. In the earlier structural optimization solver, the finite elements were mainly the types of truss and thin shell. Size optimization contains the capability in modifying the cross-section or the thickness of the finite elements. The optimization is governed by mathematical algorithms to optimize the different objective functions, e.g. minimum weight and maximum strength. However, the major restriction in the sizing problem is that the shape and the topology of the structure are fixed a priori. Undoubtedly, changes in shape (e.g., curved boundary) and topology (e.g., holes in a member) could produce more significant improvement in the performance than the modifications in size alone. Therefore, a formulation that includes the determination of the geometry (including size and shape), the topology, and the material distribution in the optimization process is likely to produce a structure with substantially higher performance quality.

The shape optimization changes the surface geometry in a manner that a homogenous stress distribution is achieved. In shape optimization the boundary of the surface is represented explicitly or implicity, and it is modified during the optimization. The strain energy density is constant everywhere along the optimal free boundary, in fact, it is also the objective of the classical shape optimization methods based on a shape sensitivity analysis by Rozvany (1998) [64]; Sokolowski and Zolesio (1992) [83]. Physically, this indicates that the mutual energy form of the elastic structure reaches a constant value on the boundary. In most shape optimization applications, a Lagrangian formulation of boundary propagation was used to achieve the optimality condition and obtain an optimal shape of the structure. The boundary changes can be accomplished only if the connectivity of the boundaries does

not change since there is a sever limitation that only a structure of a fixed topology can be optimized.

An structural optimum topology can be arrived at by optimally adding holes and changing the connectivity of the structural design domain, which is actually implemented by redistributing material in an iterative and systematic manner Eschenauer (1994) [24] and Wang at al. (2006) [92]. No initial design proposal is needed! The topology optimization as a conceptual design tool has the highest importance in the developing process of all structural optimization methods because of its ability in achieving greatest savings. Topology optimization finishes with an optimal design concept for the creation of the finite element model for the subsequent size and shape optimization. It is regarded as the best method for solving the eigenvalues problem and for producing the best overall structure.

A generalized shape and topology optimization problem has been recently studied intensively through the finite element (FE) based structural analysis. One of the most recognized categories of methods is the one based on the homogenization approach first proposed by Bendsøe and Kikuchi [11], in which the structural form is expressed as the microstructures with voids and optimized the material distribution by the optimality criteria (OC) method. Another important variant approach developed upon homogenization is the so-called SIMP (Solid Isotropic Microstructure with Penalization) method [9, 66, 80], that has been widely used in modern engineering considerations because of its computational efficiency and simple implementation [65]. However, the ill-posed material density [12] affects the boundary smoothness and thus numerical instabilities may occur to slow down or even to stop the convergence of the local optimal shape [4, 89]. Another impor-

tant family of structural optimization method is known as Evolutionary Structural Optimization (ESO) method, which is introduced by Xie and Steven [99]. The material in the design domain which is not structurally active is regarded as non-effective element, will be removed slowly [100]. In structural shape and topology optimization problems, a growing popular family of methods is based on the implicit moving interfaces with the use of the renowned level set method [91]. This method was first introduced by the pioneers of the American mathematicians, Osher and Sethian [57], in 1988. It was instantly realized as a simple and versatile method for computing and analyzing tool for tracking the evolution of the boundary in 2D or 3D domain.

Sethian, Wiegmann and Wang et al. are among the scientists who extend the level set method to embed the free boundary of the solid structure on a fixed Eulerian mesh for the optimization with the framework of the shape sensitivity [3,78,90]. The great potential of the level set method in handling sharp corners, break apart and merge together enables itself with the powerful capability on the manipulation of topological changes, reserves the precision of the boundary representation in all circumstances. Nevertheless, it is rather difficult to implement the conventional discrete level set method as it is well-known that the PDEs are not easy to calculate [45]. Although some robust and sophisticated method such as upwind scheme, extension velocities and reinitialization algorithm have been proposed [57], it is not practical to solve these problems with a reasonable computational cost.

In regard to the complex implementation of the conventional discrete level set method, we present a parametric approach to the existing method and it is generally recognized as a more effective method in the topology optimization. The original time-dependent implicit level

set function is now replaced by *Radial Basis Functions* (RBFs), which are widely used in the approximating function of huge scattered data fitting [15, 17, 71]. The RBF builds the novel interpolation upon the parametric space rather, and hence the PDE nature of the Hamilton-Jacobi can be successfully transformed into mathematically convenient ODE [18,25,37]. Parametric interpolation of the RBF function replaces the integration of implicit PDE level set function that brings more advantages in the implementation [46], but also enables the framework of shape sensitivity for structural optimization.

Since a large-scale model will lead to dense coefficient matrices in the RBF optimization and the linear system will normally require $\mathcal{O}(N^3)$ flops and $\mathcal{O}(N^2)$ memory while processing. It is becoming impractical to solve for optimal shapes as N goes over 10,000 [41, 94]. In fact, the dense system equation matrix frequently leads to the numerical instabilities and the failure of the optimization. Finally, we introduce the method of *Partition of Unity* (POU) to deal with this problem. POU is often used in 3D reconstruction of implicit surfaces from large scattered point sets [30–32,53,87,88,97]. By using the technique of 'divide-and-conquer', it breaks the global domain into smaller overlapping open covers such that the implicit functions can be more efficiently interpolated and the global solution is obtained by blending all the local solutions with a set of shape functions. The algorithms of both RBF and POU are presented here, we analyze and discuss the numerical results accordingly.

1.2 Previous Findings and Problem Statement

In this thesis, we are concerned with the programming problems defined in the following way:

$$\begin{aligned} \text{Minimize} \quad & f(x), \\ \text{subject to} \quad & g(x) \leqslant 0, \\ & x \in \Omega, \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, and $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth and $\Omega \subset \mathbb{R}^n$ is n-dimensional. In recent several years, there are many software capable of optimizing the geometry function for various engineering goals available. The operation of the algorithms used is grouped into two main categories:

1. those referred to as the method of homogenization or formally similar power law approach (or SIMP: Solid Isotropic Material with Penalization);
2. those based on the principles of evolution.

In the first case one solves a formalized mathematical problem, as a result of additional assumptions and hypotheses becomes a traditional optimization problem. The solution is obtained then through one of many optimization algorithms operating on a number of finite dimensions (space variables). Nevertheless, as mentioned previously that the insufficient density model imperils the numerical stabilities and the smoothness of the boundaries [81], we will not consider it further in this thesis even if it provides a potential of a high efficient optimization framework.

The other methods are evolutionary approaches based on intuition, where the methods of the class embedded with complex mathematical operations and programming, they rely on simple rules borrowed from the evolutionary nature. The topological optimization problem

is not limited solely to description and explanation of the operational resources. The literature in the field is vast and among them, we focus on the method based on the implicit moving boundaries – Level Set Method [1,55–58]. The great potential of the level set method in handling complex geometry and tracking moving interfaces make it easier to manipulate the topological changes during optimization, though the classical level set method with discrete representation (i.e. the signed distance form) inheres in a significant limitation in increasing the numerical efficiency.

In order to eliminate the additional load consumed in the processing, e.g. upwind scheme, extension velocity and reinitialization algorithms, it is worthy to reconstruct the level set method with the parametric shape representation [86, 93]. Consequently, the extra procedures in stabilizing the discrete distance form can be removed to increase the overall throughput. This obvious benefit motivates numerous developments in researching an appropriate parametric framework to replace the original time-consuming one.

Radial basis functions (RBFs) method is a mature technique to reconstruct an admissible design with a shape function which is globally continuous and differentiable. This is a popular methodology in the area of the computer vision to interpolate the scattering points in the domain to reproduce smooth surface or boundary, as the associated system of the linear equations is guaranteed to be invertible under the control conditions on the locations of the data points [17]. They are very effective in the construction and modeling of geometric objects [70] and the field of the artificial intelligence [14]. RBFs are capable of solving PDEs based on an irregular data distributions because the interpolations of RBFs is complete and unique when the functions to

be approximated are of multiple variables or are given only by a great amount of scattering data [46].

Efforts were made to parameterize the traditional level set method with RBFs, the authors had successfully transformed the implicit model to represent the shape and topology of an admissible design in the parametric manner. However, these studies were biased on the capability of the convergence and the accuracy of the solution, it did not show significant improvement in the efficiency and the benchmark study of both methods was not provided for further investigation [93, 94, 101]. The effects of the knot configurations to both the optimization result and the computational effectiveness were not studied. In this thesis, we study the above questions in details and try to resolve the numerical issues arising from the use of the parametric method. In addition, we extend this parametric optimization to a meshless discretization – partition of unity method (PUM) so as to increase the overall computation power and to reduce the cost in the calculation.

1.3 Overview and Original Contributions

In CHAPTER 2, we introduce the background knowledge of the structural optimization with the implicit modeling of level set method. The algorithms of the level set based optimization are discussed. Then, the parametric approach of the structural optimization that combines the theory of the radial basis functions interpolation is studied in CHAPTER 3. In addition, a dynamic knots scheme for an optimal knots distribution of RBFs is devised and tested in CHAPTER 4. The numerical results show significant improvement in the efficiency without sacrificing the accuracy at all.

Parametric level set method solves the drawbacks inhered in the

conventional method and the next in line is naturally to study the partition of unity (POU) method to further improve the numerical efficiency in CHAPTER 5. The idea is simply to solve a large number of small, local problems instead of one large, global problem and to put the local solutions together by a POU scheme. Results are given to demonstrate the distinguished advantages by applying this technique to the structural optimization with RBFs.

At the end of CHAPTER 5, we combine the benefits provided from the optimal distribution of RBFs knots in the moving knot scheme and the fast local evaluation in POU method to yield a concrete technique to achieve a high level of computational efficiency with the capability to reasonably handle the computational complexity concerning sufficiently high number of radial basis functions. At this level of capability, we should be able to apply the RBF-based parametric level set method for shape and topology optimization of a large class of structures and mechanisms in the realistic engineering problems.

□ **End of chapter.**

Chapter 2

Structural Optimization & Level Set Method

Classical optimization methods used for finding the maxima and minima of functions and functionals have direct applications in the field of structural optimization. It implies here to encompass the techniques of ordinary differential calculus and calculus of variations. The exact solutions to a few relatively simple unconstrained or equality constrained problems have been studied in the literature using both techniques. Here, we will review some of important findings that are related to the work of this thesis so as to give a more comprehensive idea about the theory applied.

2.1 Shape and Topology Optimization

The optimization of the layout or the geometry of a structure or mechanism can be formulated as a topology or shape optimization problem [12]. In the former the optimal distribution of material within the design domain of the structure or mechanism is searched, while in the

latter the type of structure (e.g. truss or shell structures) is chosen first and its shape is optimized. This general idea is easily visualized by the following diagram Figure 2.1.

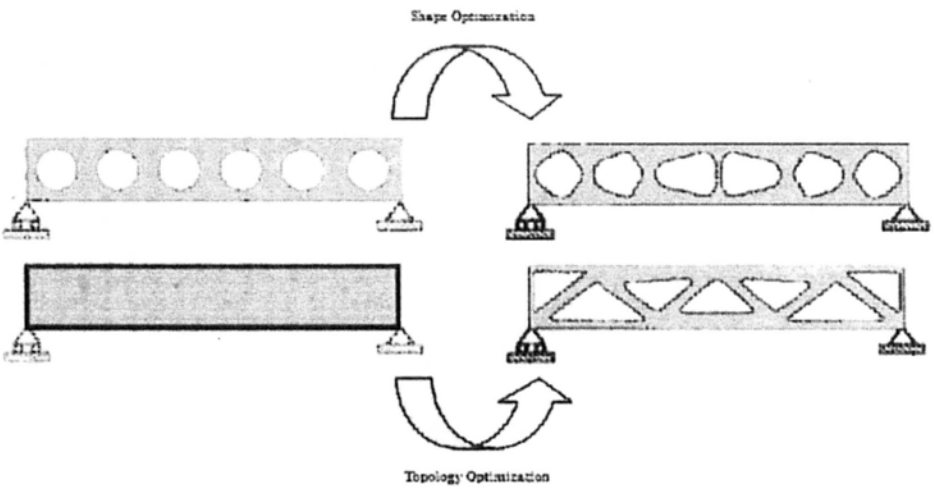


Figure 2.1: Process of shape and topology optimization

Structural optimization can be described as the problem of structural optimization using a typical setting of a linear elastic structure. Let $\Omega \subseteq D \subset \mathbb{R}^d\,(d = 2 \text{ or } 3)$ be an open and bounded set occupied by a set of n distinct material phases. As shown in Figure 2.2, the boundary of Ω consists of three parts, $\Gamma = \partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$, with Dirichlet boundary conditions on Γ_1 and Neumann boundary conditions on Γ_2. It is assumed that the boundary segment Γ_0 is traction free. The displacement field u in Ω is the unique solution of the linear elastic system. Hence, the general problem of design is specified as an optimization problem:

$$\begin{aligned} \underset{\Omega}{\text{Minimize}} \quad & J(u,\Omega) = \int_\Omega F(u)d\Omega, \\ \text{subject to} \quad & G(u,\Omega) = \int_\Omega g_j(u)d\Omega \leqslant 0 \quad (j = 1,\cdots,r)\,. \end{aligned} \tag{2.1}$$

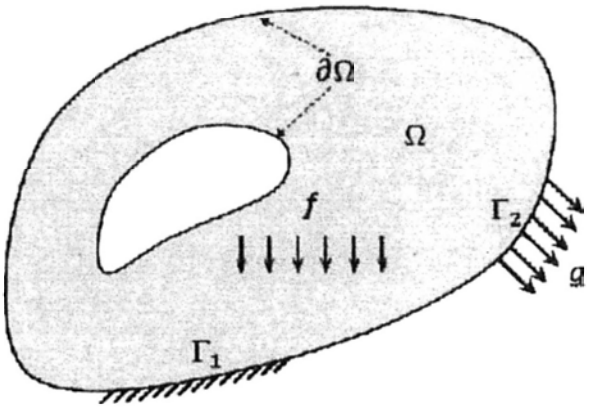


Figure 2.2: A general condition of a boundary under applied load and constraint

The design variable is actually the shape of the structure Ω and the objective function J depends on Ω in a mutual situations [62] – the integral of the explicit dependence is derived in Ω and the implicit dependence through u while it is the solution of the state equations as shown in the following elliptic partial differential equations:

$$\begin{aligned} -\text{div}\,\sigma(u) &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{in } \Gamma_1, \\ \sigma(u)\cdot n &= g \quad \text{on } \Gamma_2. \end{aligned} \tag{2.2}$$

where σ is the stresses, f the body force, u the displacement on the Dirichlet boundary Γ_1 and g the traction on the Neumann boundary Γ_2. Here we go with a set of r constraints that includes the limit on the amount of material for each phase in the admissible design. By rewriting (2.2) into its general weak variational form, the linear elastic system is expressed as

$$\begin{gathered} \int_\Omega \mathrm{E}\,\varepsilon(u) : \varepsilon(v) d\Omega = \int_\Omega f \cdot v d\Omega + \int_{\Gamma_2} g \cdot v d\Gamma, \quad \text{for all } v \in U, \\ U = \left\{ u : u \in H^1(\Omega), \quad u = u_0 \text{ on } \Gamma_1 \right\}, \end{gathered} \tag{2.3}$$

with the strain tensor ε, E the elasticity tensor, u_0 the prescribed displacement on Γ_1, f the applied body force, g the boundary traction force applied on Γ_2 such as an external pressure load exerted by a fluid, and n the outward normal to the boundary, with U denoting the space of kinematically admissible displacement fields and the symbol ':' representing the second order tensor operation. The goal of the optimization is to find a minimizer Ω for the optimization criterion $J(u, \Omega)$, which yields an optimized structure with respect to a specific function described by $F(u)$ (e.g. strain energy density). This is a standard notion of structural topology optimization [11, 12, 65].

2.2 Shape Derivatives

In order to minimize the objective function in (2.1), it is necessary to find out the gradient of change of $J(u, \Omega)$ with respect to a perturbation on the shape Ω and this process is known as *shape sensitivity analysis.* The variational result of the objective function is then called *shape derivative.* In the following context, we briefly introduce the general problem with the use of the variational method which is proposed by [4, 33, 59, 85]. A more detailed derivation process for the parametric shape gradient of the objective function involved with structure optimization problem will be presented in next chapter.

Suppose that Ω is a domain with a continuous and smooth boundary as shown in Figure 2.3. A vector field V is applied on the boundary for a short moment τ and assume there is a domain mapping from Ω to Ω^τ. With sufficient regularity assumptions, then we have

$$\begin{aligned} \delta J &\equiv J(u(\Omega^\tau), \Omega^\tau) - J(u, \Omega), \\ &= [J_\Omega + J_u u_\Omega]\, \delta\Omega, \end{aligned} \tag{2.4}$$

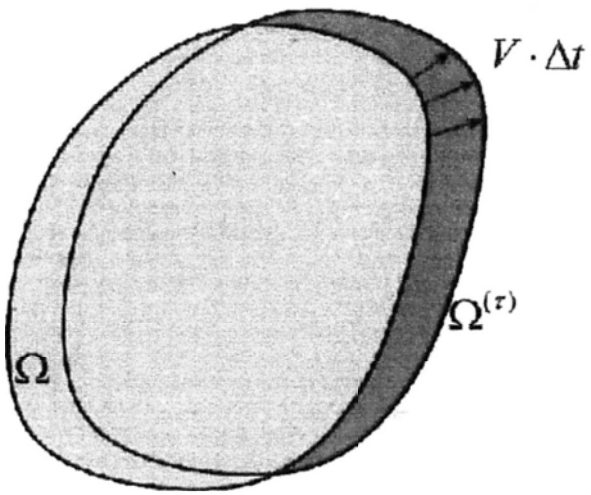


Figure 2.3: Perturbation of the boundary of Ω by a vector field V

where J_Ω is the derivative of the objective function J with respect to the shape Ω with the state variable u remained constant. As implied by (2.1), the objective funetion of the domain $J(u,\Omega) = \int_\Omega f(u(\Omega))d\Omega$ is varied as

$$\delta J = \int_\Omega \delta f(u) d\Omega + \int_{\delta\Omega} f(u) d\Omega. \tag{2.5}$$

The perturbation integral can be extended with the vector field as [4]

$$\int_{\delta\Omega} f(u)\, d\Omega \simeq \int_{\partial\Omega} f(u) V \cdot n ds, \tag{2.6}$$

substitute (2.6) into (2.5), then we get

$$\delta J = \int_\Omega f_u \delta u d\Omega + \int_{\partial\Omega} f(u) V \cdot n ds. \tag{2.7}$$

For $J(u,\Omega)$ expressed in the boundary integral $J(u,\partial\Omega) = \int_{\partial\Omega} f(u(\Omega))ds$, its weak form is written as [85]

$$\delta J = \int_{\partial\Omega} f_u \delta u ds + \int_{\partial\Omega} V \cdot n \left(\frac{\partial f}{\partial n} + f\kappa\right) ds, \tag{2.8}$$

where κ is the mean curvature defined by $\kappa = \operatorname{div} n$ and $\frac{\partial f}{\partial n} = \nabla f \cdot n$. Since the virtual displacement δu or u_Ω cannot be evaluated directly,

the *adjoint equation* will be applied to (2.8) to eliminate the term of the infinitesimal change.

As shown in (2.4), we denote the state equation as $W(u, \Omega) = 0$ and find the variation of the state equation at 0 of the perturbation, i.e.

$$\delta W = [W_\Omega + W_u u_\Omega]\, \delta\Omega \equiv 0, \tag{2.9}$$

then, multiply the δW with a Lagrange multiplier λ and add the product with δJ in (2.4)

$$\begin{aligned} \delta J &= \delta J + \lambda \delta W, \\ &= [J_\Omega + J_u u_\Omega + \lambda\,(W_\Omega + W_u u_\Omega)]\, \delta\Omega, \\ &= (J_\Omega + \lambda W_\Omega)\, \delta\Omega + (J_u + \lambda W_u)\, u_\Omega \delta\Omega. \end{aligned} \tag{2.10}$$

Assume $(J_u + \lambda W_u) = 0$, we can simplify (2.10) as

$$\delta J = (J_\Omega + \lambda W_\Omega)\, \delta\Omega. \tag{2.11}$$

The only additional effort in this method is necessary to solve the adjoint equation $(J_u + \lambda W_u) = 0$ for the Lagrange multiplier λ and this parameter is important in the functional analysis later in the numerical process [62].

2.3 Level Set Method of Implicit Function

Numerical methods of shape and topology optimization based on the level set representation and on shape differentiation make possible topology changes during the optimization process. Level set method is a numerical technique for tracking interfaces and shapes. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid

without having to parameterize these objects (this is called the Eulerian approach) [57]. Also, the level set method makes it very easy to follow shapes that change topology on a Euler grid by solving a Hamilton-Jacobi partial differential equation.

Level set method represents the motion of the dynamic interfaces through the implicit level set function $\Phi(x)$ which is Lipschitz-continuous and the interface can be simulated with the zero iso-surface i.e. zero level set of the function $\{x \in \mathbb{R}^d \mid \Phi(x) = 0\}$ ($d = 2$ or 3). The evolution of the free boundary is described by solving the Hamilton-Jacobi equation [58].

DEFINITION 2.1 *Define the bounded domain $D \subset \mathbb{R}^d$ be the working domain in which all admissible shapes Ω are included, i.e. $\Omega \subset D$. The shape and topology of a structure are described by a level set function $\Phi(x)$ as follows*

$$\begin{aligned} \Phi(x) &= 0 \quad \forall x \in \partial\Omega \cap D, \\ \Phi(x) &< 0 \quad \forall x \in \Omega \setminus \partial\Omega, \\ \Phi(x) &> 0 \quad \forall x \in (D \setminus \Omega). \end{aligned} \tag{2.12}$$

As shown in Figure 2.4, the convention of the interior and the exterior is distinguished clearly by (2.12).

2.4 Level Set Equation

In the level set method, the normal velocity is defined as the advection velocity in the transport equation [58].

LEMMA 2.2 *The normal velocity is the product of the velocity vector field $V = \frac{dx}{dy}$ and the normal vector n of the boundary. Let v_n be the normal velocity on the boundary, such that $v_n = V \cdot n$ and $n =$*

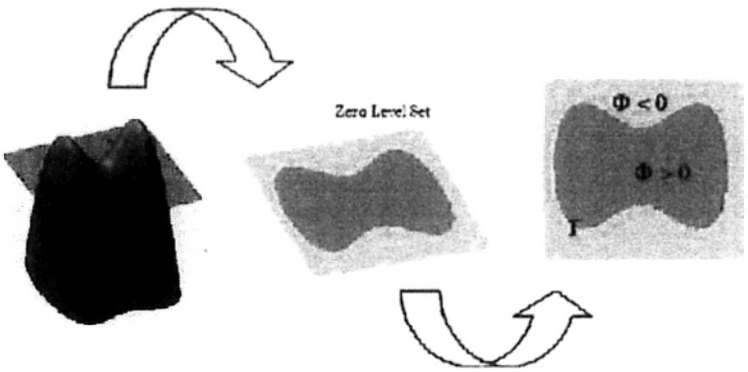


Figure 2.4: An illustration of the level set method and its implicit function

$\frac{\nabla\Phi}{|\nabla\Phi|}$. *The Hamilton-Jacobi equation is obtained by the differentiation of* $\Omega(t, x(t)) = 0$ *with respect to* t*, where* $x(t)$ *is a point on the boundary,*

$$\frac{\partial\Phi}{\partial t} + v_n \left|\nabla\Phi\right| = 0. \tag{2.13}$$

Remark Moving the boundary of the domain represented implicitly by the zero level set function, i.e. $\{\forall x \in \partial\Omega \mid \Phi(x) = 0\}$, along the descent gradient direction is equivalent to move the implicit function Φ by solving (2.13).

The partial differential equation (2.13) is formulated as the level set equation [57, 58, 76] and the optimization problem is transformed into a problem of finding the steady-state solution of the Hamilton-Jacobi equation. In order to move the boundary in the descent direction, the velocity v_n is determined from the sensitivity analysis in respect of the different minimization objectives.

2.5 Discrete Level Set Method

In order to discretize the domain into an Eulerian grid, we fix D as the unit square $D = (0, 1) \times (0, 1)$. For the discretization of the Hamilton-

Jacobi equation (2.13), we first define the mesh grid of D. We introduce the nodes P_{ij} whose coordinates are given by $(i\Delta x, j\Delta y)$, where Δx and Δy are the discretization steps in the x and y directions, respectively. Let us also denote $t^k = k\Delta t$ the discrete time for $k \in \mathbb{N}$, where Δt is the time step. We are seeking an approximation $\phi_{ij}^k \simeq \phi(P_{ij}, t^k)$.

Normally, the numerical scheme of the partial differential equations will be implemented with the simple finite difference method. However, it is not easy to get a sufficient stability for the convergence in the boundary value problem [28], we propose the first-order explicit upwind scheme for 2D instead

$$\phi_{ij}^{k+1} = \phi_{ij}^k - \Delta t\, g(D_-^x \phi_{ij}^k, D_+^x \phi_{ij}^k, D_-^y \phi_{ij}^k, D_+^y \phi_{ij}^k), \tag{2.14}$$

where

$$D_-^x \phi_{ij} = \frac{\phi_{ij} - \phi_{i-1,j}}{\Delta x}, D_+^x \phi_{ij} = \frac{\phi_{i+1,j} - \phi_{ij}}{\Delta x} \tag{2.15}$$

are respectively the backward and forward approximations of the x-derivatives of ϕ at P_{ij}. Similar expressions hold for the approximations D_-^y and D_+^y of the y-derivatives. The numerical flux is given by

$$g(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij}) g_{ij}^{(1)} + g_{ij}^{(2)}. \tag{2.16}$$

The numerical flux part $g_{ij}^{(1)}$ corresponds to the discretization of the first-order part of the Hamilton-Jacobi equation and is given by

$$g_{ij}^{(1)} = \max(v_{ij}, 0)\, G^+ + \min(v_{ij}, 0)\, G^-$$

with

$$\begin{aligned} G^+ = \Big[& \max\left(D_-^x \phi_{ij}, 0\right)^2 + \min\left(D_+^x \phi_{ij}, 0\right)^2 \\ & + \max\left(D_-^y \phi_{ij}, 0\right)^2 + \min\left(D_+^y \phi_{ij}, 0\right)^2 \Big]^{1/2}, \\ G^- = \Big[& \min\left(D_-^x \phi_{ij}, 0\right)^2 + \max\left(D_+^x \phi_{ij}, 0\right)^2 \\ & + \min\left(D_-^y \phi_{ij}, 0\right)^2 + \max\left(D_+^y \phi_{ij}, 0\right)^2 \Big]^{1/2}, \end{aligned}$$

and $v_{ij} = V_{ext}(P_{ij})$ is the extended normal velocity at point P_{ij}.

The numerical flux part $g_{ij}^{(2)}$ in (2.16) is the centered finite difference approximation of the second-order term of the Hamilton-Jacobi equation, i.e. $g_{ij}^{(2)} \simeq -2\mu P_c(\Omega)\mathcal{H}|\nabla\phi|(P_{ij})$.

This upwind scheme is stable under the following two conditions:

$$\left(\max_D |V_{ext}|\right) \frac{\Delta t}{\min(\Delta x, \Delta y)} \leqslant \frac{1}{2\sqrt{2}}, \tag{2.17}$$

$$4\mu P_c(\Omega) \frac{\Delta t}{\min(\Delta x^2, \Delta y^2)} \leqslant 1. \tag{2.18}$$

The condition (2.17) arises from the upwind part of the scheme corresponding to the discretization of the first-order term of the Hamilton-Jacobi equation. The condition (2.18) comes from the centered difference discretization used for the second-order term involving the curvature.

2.6 Reinitialization of Level Set Function

Let us start with a general remark on the numerical solution of (2.13). For the sake of numerical accuracy, the solution of the level set equation should be neither too flat nor too steep. This condition is fulfilled, e.g., if ϕ is the distance function, i.e. $|\nabla\phi| = 1$. Unfortunately, even if we start with a *signed distance function* $d(x)$ for the initial data ϕ_0,

$$\phi(x) = \begin{cases} d(x) & , \quad \forall x \in D\backslash\Omega \\ 0 & , \quad \forall x \in \partial\Omega \\ -d(x) & , \quad \forall x \in \Omega\backslash\partial\Omega \end{cases}, \tag{2.19}$$

where $d(x)$ is the *distance function* defined as

$$d(x) = \min(|x - x_i|), \quad \text{for } \forall x_i \in \partial\Omega, \tag{2.20}$$

the solution ϕ of the level set equation does not generally remain close to any distance function. We can perform a reinitialization of ϕ at a

time t by determining the solution $\varphi = \varphi(x, \tau)$ of the following equation, up to the stationary state:

$$\varphi_\tau + \mathcal{S}(\phi)\left(|\nabla\varphi| - 1\right) = 0 \quad \text{in } D \times \mathbb{R}^+, \tag{2.21}$$

$$\varphi(x, 0) = \phi(x, t), \quad x \in D, \tag{2.22}$$

Here $\mathcal{S}$ is an approximation to the sign function

$$\mathcal{S}(d) = \frac{d}{\sqrt{d^2 + |\nabla d|^2 \varepsilon^2}}, \tag{2.23}$$

with $\varepsilon = \min(\Delta x, \Delta y)$, where Δx and Δy stand for the space discretization steps in the x and y directions, respectively. Other choices are also possible for the approximate sign function. Readers can refer to [6] for more details.

2.7 Extended Normal Velocity

The normal velocity v_n in LEMMA 2.2 should be defined on the whole domain D for the well posedness of the level set equation (2.13). In Figure 2.5, the normal velocity v_n is only given on the Neumann boundary Γ_2 and we need to extend v_n to the domain D. Another reason for extending the velocity is to enforce the solution ϕ of the level set equation to remain close to the distance function. Indeed, if we are

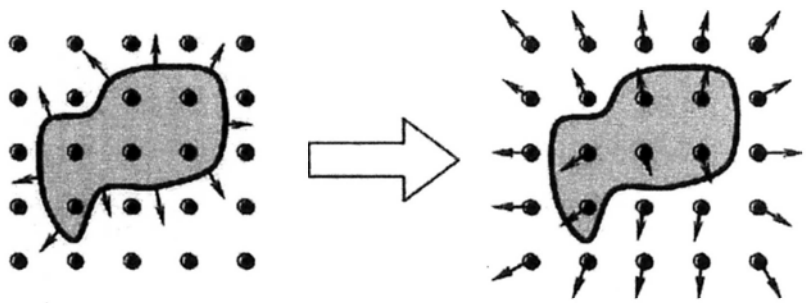


Figure 2.5: An illustration of the method of the extended normal velocity

able to compute an extended normal velocity V_{ext} such that

$$\nabla V_{ext} \cdot \nabla \phi = 0 \quad \text{in } D \times \mathbb{R}^+, \tag{2.24}$$

then it can be shown [103] that the solution ϕ to (2.13) satisfies the distance function $|\nabla \phi| = 1$. A way to construct an extension V_{ext} satisfying (2.24) at time t is to solve the following equation for q, up to the stationary state [58]

$$q_\tau + S(\phi)\frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla q = 0 \quad \text{in } D \times \mathbb{R}^+, \tag{2.25}$$

$$q(x,0) = p\,(x,t)\,, \quad x \in D, \tag{2.26}$$

where $p = v_n$ on the boundary Γ_2 and 0 elsewhere. The function S is the approximate sign function defined by (2.23).

2.8 Topological Derivative

The shortcoming of the method of shape derivative is that no nucleation of holes inside the domain are allowed. Numerical methods based on the shape derivative may therefore fall into a local minimum and hence they do not solve the inherent problem of ill-posed of shape optimization, which manifests itself in the existence of many local minima, usually having different topologies [3, 4].

The level set approach based on the shape sensitivity may get stuck at shapes with fewer holes than the optimal geometry in some applications to construct designs. To address this issue, [35] proposed a modified level set method that includes the topological derivative. This derivative measures the influence of creating a small hole in the interior domain.

The topological derivative is unlike the shape derivative, for which it is defined on local perturbations of the boundary of the domain,

that measures the influence of the hole creation at a certain point. The concept is to create a small ball $B_{\rho,x}$ with center x and radius ρ inside or outside the shape Ω and then consider the variation of the objective functional J with respect to the volume of this ball. For $x \in \overline{\Omega}$, the topological derivative $d_T\mathcal{F}(\Omega)(x)$ is defined as

$$d_T\mathcal{F}(\Omega)(x) := \lim_{\rho \to 0} \frac{\mathcal{F}(\Omega_{\rho,x}) - \mathcal{F}(\Omega)}{|B_{\rho,x} \cap \Omega|}, \tag{2.27}$$

where $\Omega_{\rho,x} = \Omega - \overline{B(\rho, x)}$. It means to subtract material at $x \in \overline{\Omega}$. But in some situations, it is reasonable to add material at $x \in D - \overline{\Omega}$ In this case, the "set-minus" must be replaced by "union" in (2.27). For simplicity, we will only derive the "set-minus" case and extend the solution to the "union" case by analogy.

In fact, there exist functionals that are not differentiable:

$$d_T|\partial\Omega| = \lim_{\rho \to 0} \frac{|\partial B_{\rho,x}|}{|B_{\rho,x}|} \simeq \lim_{\rho \to 0} \frac{\rho^{N-1}}{\rho^N} = \infty. \tag{2.28}$$

When the topological derivative does exist, by Taylor expansion we naturally have $F(\Omega_{\rho,x}) = F(\Omega) + \pi\rho^2 d_T\mathcal{F}(\Omega)(x) + o(\rho^2)$. That shows $F(\Omega_{\rho,x})$ is second-differentiable with zero first derivative at $\rho = 0$. Furthermore we assume that $F(\Omega_{\rho,x}) \in C^2(\mathbb{R}, \mathbb{R})$ with respect to ρ for small ρ.

We mark that it might not be easy to compute the shape derivative of the objective functional $\mathcal{F}(\Omega)$. However, the purpose of this paper is not to incorporate the topological derivative into the parametric level set method, which assumes that the information of the shape derivative is known already. For explicit computations of topological derivatives in shape optimization, etc., please refer to [5, 29, 34, 51, 82] for more information.

□ **End of chapter.**

Chapter 3

Parametric Optimization with RBF Implicit Modeling

This chapter discusses a structural optimization method that optimizes shape and topology based on the parametric method. The proposed method has the same functional capabilities as a structural optimization method based on the level set method incorporating signed distance functions. The advantage of the method is the simplicity of computation, since extra operations such as re-initialization of functions are not required [86, 93].

Structural shapes are represented by RBFs defined in the design domain, and optimization of this function is performed by solving a time-dependent equation. The algorithms used in this optimization are derived from the sensitivity analysis. The proposed method is applied to two-dimensional linear elastic problems such as the minimum compliance problem. The numerical examples provided illustrate the convergence of the objective function and the effect of the knot density on the optimal configurations.

3.1 Settings of the Problem

For the minimization of the linearized elasticity system, it can be expressed as

$$\begin{aligned} \min_{\Phi} \quad & J(u,\Phi) = \int_D \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) H(\Phi) d\Omega, \\ \text{s.t.} \quad & V(\Phi) \leqslant V_{max} \qquad \text{where} \quad V(\Phi) = \int_D H(\Phi) d\Omega, \end{aligned} \tag{3.1}$$

where $u \in \mathbb{R}^d$ is the displacement field, $\varepsilon(u)$ the strain field, C the elasticity tensor, V the volume constraint and $H(\Phi)$ the Heaviside function. The unconstrained optimization problem can be obtained by multiplying the volume constraint $V(\Phi)$ with a positive Lagrange multiplier λ and hence the system equation is now transformed to

$$\min_{\Phi} \qquad L(u,\Phi) = \int_D \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) H(\Phi) d\Omega + \lambda \left(V(\Phi) - V_{\max} \right). \tag{3.2}$$

LEMMA 3.1 *Let Ω be a smooth bounded open set upon the local perturbation of the boundary of the admissible domain, the shape derivative can be obtained by the differentiation of the Lagrangian L [4, 91] with respect to the artificial time t,*

$$\frac{dL}{dt} = \int_{\partial\Omega} \left(\lambda - \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) \right) v_n ds. \tag{3.3}$$

Remark Consequently, a normal velocity in the descent direction is defined as

$$v_n = -\left(\lambda - \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) \right). \tag{3.4}$$

The right-hand side of (3.4) shows that the normal velocity v_n on the boundary of the shape is related with the strain energy density. Conventionally, we substitutes the velocity (3.4) into (2.13) to solve for

the front of the moving boundary and hence the optimization can be proceeded.

3.2 Parametrization of Level Set Method

Parametric level set method solves the drawbacks inhered in the conventional method, in order to eliminate the additional load consumed in the processing, e.g. upwind scheme, extension velocity and reinitialization algorithms, we need to parameterize the conventional discrete level set equation. The level set function $\Phi(x,t)$ is a shape function with the order of C^0 continuity in space. For solving the Hamilton-Jacobi equation on a fixed Cartesian grid, the elliptic PDE of time-space domain is expressed as

$$\frac{\partial \Phi(x,t)}{\partial t} + v_n \left|\nabla \Phi(x,t)\right| = 0, \tag{3.5}$$

which is a finite difference of finite element method over a rectilinear mesh [58, 78, 91]. As the mesh is fixed on space, only the nodal values of the implicit function $\Phi(x)$ are calculated. It is important to characterize precisely the geometry by the nodal values of $\Phi(x)$ in Eulerian approach and hence the shape function can represent all the admissible designs smooth enough in the domain [38, 57]. In fact, only the smooth function with low order approximations $\mathcal{F} \in C^0$ shape are used [8, 19, 58]. It is because of the polynomial snaking problem that the interpolation in higher orders can easily lead to singularity and induce difficulty in achieving the convergence of the solution [36, 37, 68, 69].

Only the implicit function $\Phi(x)$ can be guaranteed to be continuous across the mesh but not its partial derivatives. In addition, the spatial truncation errors due to the low order accuracy can only be minimized by using aggressively fine mesh. Consequently, the mesh spacing must be small enough to arrest the changing of the derivatives accurately

and then the artifacts contamination to the numerical results can be avoided. However, this make the computation very costly and time-consuming, in result, this method is impossible to simulate the large scale problem in question. On the other hand, some extra numerical algorithms, e.g. upwind scheme and reinitialization and extended normal velocity, are necessary to apply into the process so as to maintain the stability and convergence of the solution.

All the drawbacks of the conventional method can be eliminated by the parametrization while retaining the geometrical benefits of the implicit representation. As proposed in [93], author transformed the level set function $\Phi(x)$ into an alternative implicit parametric shape representation by generalizing it with radial basis functions (RBFs).

Some of the advantages of radial basis functions are their insensitivity to the spatial dimension d, which makes the implementation of this method in higher dimensions much simpler than, e.g. finite elements. Another useful feature of radial basis functions is their radial symmetry and invariance under Euclidean transformations. Furthermore, in the context of scattered data interpolation it is known that some radial basis functions have spectral convergence orders (e.g. (reciprocal) multiquadrics, Gaussians).

Under this parametrization the level set function $\Phi(x)$ is derived as a time-dependent parametric form

$$\Phi = \Phi(x,t) = \varphi^{\mathrm{T}}(x) \cdot \alpha(t), \tag{3.6}$$

where $\varphi(x)$ are a set of radial basis functions and $\alpha(t)$ are the expansion coefficient corresponding to the artificial time in the iteration. Substitute (3.6) into the Hamilton-Jacobi equation (3.5) we can get the following form:

$$\varphi^{\mathrm{T}}(x)\frac{d\alpha(t)}{dt} + v_n \left|(\nabla\varphi(x))^{\mathrm{T}}\,\alpha(t)\right| = 0, \tag{3.7}$$

which indicates the relationship between the boundary propagation velocity v_n and the time derivative of the expansion coefficients $\dot{\alpha}$. It implies that if the velocity field is available, the iteration of the parameters α can be evaluated then [25, 36]. By combining the collocation method with the extended normal velocity, the original PDE based optimization problems are transformed into a set of simpler ODE initial condition problems and the problems can be solved by several different ODE solver such as the first-order forward Euler's method and higher-order Runge-Kutta, Rung-Kutta-Fehlberg, Adams-Bashforth, or Adams-Moulton methods.

Because of the numerical stability offered by RBFs, we will evaluate the solution of the elliptic partial differential equations with the popular method of the collocation approach based on radial basis functions. Since the level set function is continuously represented in a differentiable condition, the extra reinitialization process is no longer need to regularize the conventional discrete level set method. Even though this parametrization converts the level set method into more efficient method and simplifies the implementation, it still confines the level set based optimization method in a framework of differential equations but not actually transformed it into a mathematic programming approach.

In the later section, we propose another parametric approach based on sensitivity analysis to solve for the general minimization problem. The sensitivity algorithms determine the search direction in terms of the parameters and there are numerous sophisticated schemes for solving the equations, for example, the steepest descent method, the method of moving asymptotes (MMA) and the optimality criteria (OC) method. Nevertheless, it is still a popular way to solve this structural optimization with the collocation scheme [93, 94] and hence we will in-

troduce this method together with the sensitivity approaches in Section 3.4 in details.

3.3 Parametric Shape Representation

In order to eliminate the disadvantages arising from the use of the discrete method, we parameterize Φ with a new function that is globally continuous and differentiable. An implicit model based on radial basis functions (RBFs) is considered. RBFs are common in the application of scattered data interpolation, and guarantees to provide smooth boundary as it is invertible on the location of the data locations [15,93].

3.3.1 RBFs Implicit Modeling

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$; or alternatively on the distance from some other point c, called a center, so that $\varphi(\mathbf{x}, \mathbf{c}) = \varphi(\|\mathbf{x} - \mathbf{c}\|)$ [15]. Any function Φ that satisfies the property $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ is a radial function. The norm is usually Euclidean distance, although other distance functions are also possible. For example by using Lukaszyk-Karmowski metric it is for some radial functions possible [42] to avoid problems with ill conditioning of the matrix solved to determine coefficients α_i, since the $\|\mathbf{x}\|$ is always greater than zero.

RBFs method is a well-developed methodology to approximate or reconstruct an admissible design with a single function which is globally continuous and differentiable. They are popular for interpolating scattered data to produce smooth surface or boundary as the associated system of non-linear equations is guaranteed to be invertible

under mild conditions on the locations of the data points [17]. Radial basis functions techniques have become extremely useful, ranging from construction and modeling of geometric objects [70–72, 75], artificial intelligence [14], to simply solving PDEs based on irregular data distributions [26] since the interpolations based on radial basis functions are effective when the functions to be approximated are of multiple variables or are given only by a great amount of data or by scattered data. Some attractive features of radial basis functions such as the unique solvability of the interpolation problem and their smoothness and convergence make them very easy to be accepted in topology optimization field. Incorporating with the level set method, RBFs implicit modeling is regarded as an effective representation to reconstruct the shape and topology of an admissible design, as its global continuity maintain a high fidelity of the topological characteristics.

In the following we assume we are given a set of knots $\{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, along with a continuous function $\varphi\colon [0, \infty) \to \mathbb{R}$. We then refer to:

$$\varphi_i(x) = \varphi(\|x - x_i\|) \qquad x_i \in \mathbb{R}^d, \quad i \in \{1, \ldots, N\}, \tag{3.8}$$

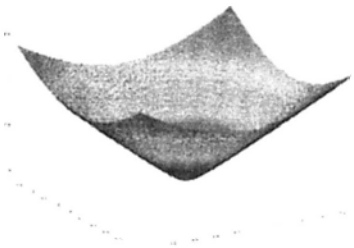
where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$, x_i is the position of the knots, and $\varphi\colon \mathbb{R}^+ \to \mathbb{R}$ with $\varphi(0) \geqslant 0$. There are numerous radial basis functions commonly used in computation, for which they are usually divided into two categories: *globally supported radial basis functions* (GSRBFs), including multiquadrics (MQs), reciprocal or inverse multiquadrics (IMQs), thin-plate spline, polyharmonic splines, Sobolev splines and Gaussians, the another one is *compactly supported radial basis functions* (CSRBFs) [95, 98]. The kernels of the globally supported RBFs and the CSRBFs proposed by Wendland [95] are listed in Table 3.1 and Table 3.2, respectively. Examples of these basis functions are displayed in Figure 3.1.

Name	$g(r)$ $(r = \|x - x_i\|)$	Parameters
Thin-plate spline	$r^2 \ln r$	$x \in R^2$
Cubic spline	r^3	$x \in R^3$
Polyharmonic splines	$r^{2n} \ln r$	$n \geq 1, x \in R^2$
polyharmonic splines	r^{2n-1}	$n \geq 1, x \in R^3$
Sobolev spline	$r^\nu K_\nu(r)$	$\nu > 0, K_\nu$:spherical Bessel function
Matern spline	$e^{-cr} K_\nu(cr)$	$\nu > 0, c > 0$
Exponential spline	e^{-cr}	$c > 0$
Gaussians	e^{-cr^2}	$c > 0$
Multiquadrics	$\sqrt{r^2 + c^2}$	$c > 0$
Inverse Multiquadrics	$\frac{1}{\sqrt{r^2+c^2}}$	$c > 0$
Compactly supported	$(1-r)_+^m p(r)$	$m \geq 2, p(r)$: polynormial of Wendland

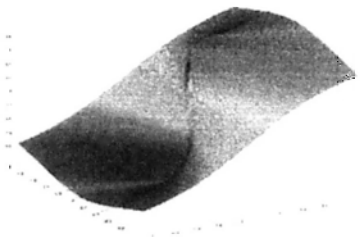
Table 3.1: Examples of Radial Basis Functions

Dimension	$g(r)$ $(r = \|x - x_i\|)$	Continuous Degree
d=1	$g_{1,0} = (1-r)_+$	C^0
	$g_{2,1} = (1-r)_+^3(3r+1)$	C^2
	$g_{3,2} = (1-r)_+^5(8r^2+5r+1)$	C^4
d=3	$g_{2,0} = (1-r)_+^2$	C^0
	$g_{3,1} = (1-r)_+^4(4r+1)$	C^2
	$g_{4,2} = (1-r)_+^6(35r^2+18r+3)$	C^4
	$g_{5,3} = (1-r)_+^8(32r^3+25r^2+8r+1)$	C^6
d=5	$g_{3,0} = (1-r)_+^3$	C^0
	$g_{4,1} = (1-r)_+^5(5r+1)$	C^2
	$g_{5,2} = (1-r)_+^7(16r^2+7r+1)$	C^4

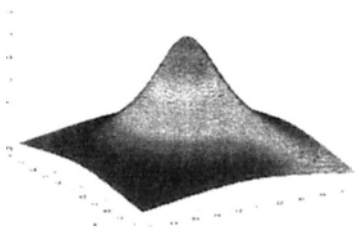
Table 3.2: Examples of Wendland Compactly Supported Radial Basis Functions [95]



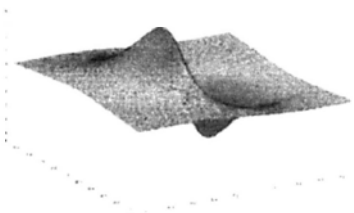
(a) Multiquadrics Splines



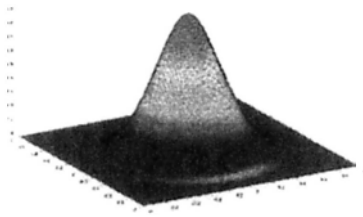
(b) Partial Derivative of Multiquadrics Splines in x direction



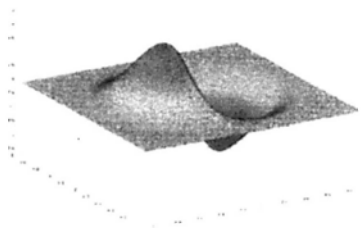
(c) Inverse Multiquadrics Splines



(d) Partial Derivative of Inverse Multiquadrics Splines in x direction



(e) C2-CSRBF Splines



(f) Partial Derivative of C2-CSRBF in x direction

Figure 3.1: Illustrations of RBFs Shape Functions

3.3.2 Collocation with RBFs

In the following we use the multiquadrics (MQs) as an example to demonstrate the idea of the RBFs modeling. The multiquadric kernel is defined as:

$$\varphi_i(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{x}_i)^2 + c^2}, \tag{3.9}$$

where $\mathbf{x}_i$ is the position of the knot and c is the free shape parameter which is regarded as a constant for all knots [27, 49] in most cases. In the RBFs implicit modeling, the MQs represent the scalar implicit function $\Phi(\mathbf{x})$ with N knots and MQs by interpolating them at their centers

$$\Phi(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \varphi_i(\mathbf{x}) + p(\mathbf{x}), \tag{3.10}$$

where α_i is the weighting or expansion coefficient of the radial basis function at the location of the i-th knot $\mathbf{x}_i$ and $p(\mathbf{x})$ is the first-degree polynomial to account for the linear and the constant portions of $\varphi(\mathbf{x})$ and to ensure polynomial precision [15]. For three dimensional problems, $p(\mathbf{x})$ is given by

$$p(\mathbf{x}) = p_0 + p_1 x + p_2 y + p_3 z, \tag{3.11}$$

where p_0, p_1, p_2 and p_3 are the coefficients of the polynomial $p(\mathbf{x})$ and (x, y, z) is the coordinate of the point $\mathbf{x} \in \mathbb{R}^3$. In order to ensure a unique solution, the coefficients in (3.10) must be subject to the following orthogonality or side constraints [15, 17]:

$$\sum_{i=1}^{N} \alpha_i = 0 \quad \sum_{i=1}^{N} \alpha_i x_i = 0 \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \sum_{i=1}^{N} \alpha_i z_i = 0, \tag{3.12}$$

and (3.10) can be rewritten as

$$\Phi(\mathbf{x}) = \boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\alpha}, \tag{3.13}$$

where

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \varphi_1(\mathbf{x}) & \dots & \varphi_N(\mathbf{x}) & 1 & x & y & z \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(N+4)\times 1}, \tag{3.14}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \dots & \alpha_N & p_0 & p_1 & p_2 & p_3 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(N+4)\times 1}, \tag{3.15}$$

The multiquadric collocation matrix $\mathbf{A}$ is theoretically invertible [15,37, 44], the generalized expansion coefficients $\boldsymbol{\alpha}$ is thus given by rewriting (3.10) into a matrix form

$$\boldsymbol{\alpha} = \mathbf{A}^{-1}\Phi \tag{3.16}$$

or expressed explicitly as

$$\begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{Bmatrix} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \varphi_2(\mathbf{x}_1) & \dots & \varphi_N(\mathbf{x}_1) & 1 & x_1 & y_1 & z_1 \\ \varphi_1(\mathbf{x}_2) & \varphi_2(\mathbf{x}_2) & \dots & \varphi_N(\mathbf{x}_2) & 1 & x_2 & y_2 & z_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varphi_1(\mathbf{x}_N) & \varphi_N(\mathbf{x}_N) & & \varphi_N(\mathbf{x}_N) & 1 & x_N & y_N & z_N \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \dots & x_N & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_N & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \dots & z_N & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{Bmatrix} \Phi(\mathbf{x}_1) \\ \Phi(\mathbf{x}_2) \\ \vdots \\ \Phi(\mathbf{x}_N) \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \tag{3.17}$$

It should be noted that the necessary degree of the polynomial item $p(\mathbf{x})$ in (3.10) to make the collocation matrix $\mathbf{A}$ positive definite is varied for different radial basis functions. The polynomial item is not necessary for the inverse multiquadrics (IMQs) and multiquadrics (MQs) [25] so that we will use them to evaluate the results in the following experiments.

3.4 Parametric RBFs based Level Set Method

A level set method with RBFs is proposed to transform the Hamilton-Jacobi PDE into a system of the first-order ordinary differential equations (ODEs) over the entire domain D to solve structural optimization problems with the use of the level set method efficiently, in such way, we will achieve with a significant mathematical convenience. As aforementioned, in the level set based topology optimization methods, moving the boundary of the shape along a descent gradient direction to find an optimal shape and topology is equivalent to transporting the scalar implicit function $\Phi(\mathbf{x})$ by solving the Hamilton-Jacobi equation (3.5) and [4] the optimal propagation of the front is performed by solving the Hamilton-Jacobi PDE. In the following studies, RBFs implicit modeling is used to interpolate $\Phi(\mathbf{x})$ with N knots by using IMQs centered at these knots. Since the Hamilton-Jacobi equation (3.5) is time-dependent, we assume that the space and time are separable and the time-dependent implicit function Φ is related to the generalized expansion coefficients $\boldsymbol{\alpha}$ of the RBF interpolant in (3.15). With these assumptions, the RBF interpolant of the RBFs interpolant in (3.13) in 3-dimensions becomes time-dependent as follows:

$$\Phi(\mathbf{x}, t) = \boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\alpha}(t), \tag{3.18}$$

and the orthogonality constraints in (3.12) can be re-written as

$$\sum_{i=1}^{N} \alpha_i(t) = 0 \quad \sum_{i=1}^{N} \alpha_i(t)x_i = 0 \quad \sum_{i=1}^{N} \alpha_i(t)y_i = 0 \quad \sum_{i=1}^{N} \alpha_i(t)z_i = 0, \tag{3.19}$$

substituting (3.18) into the Hamilton-Jacobi equation (3.5) yields

$$\boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x})\frac{d\boldsymbol{\alpha}}{dt} + v_n \left|(\nabla\boldsymbol{\phi})^{\mathrm{T}}\boldsymbol{\alpha}\right| = 0, \tag{3.20}$$

where

$$\nabla\phi = \frac{\partial\phi}{\partial x}\mathbf{i} + \frac{\partial\phi}{\partial y}\mathbf{j} + \frac{\partial\phi}{\partial z}\mathbf{k}, \tag{3.21}$$

$$\left|(\nabla\phi)^{\mathrm{T}}\boldsymbol{\alpha}\right| = \left[\left(\frac{\partial\phi^{\mathrm{T}}}{\partial x}\boldsymbol{\alpha}\right)^2 + \left(\frac{\partial\phi^{\mathrm{T}}}{\partial y}\boldsymbol{\alpha}\right)^2 + \left(\frac{\partial\phi^{\mathrm{T}}}{\partial z}\boldsymbol{\alpha}\right)^2\right]^{1/2}. \tag{3.22}$$

In (3.20), the time dependence of the Hamilton-Jacobi equation is due to the expansion coefficients as the expansion coefficients are explicitly related to time. At the initial time, all the time-dependent variables should be specified over the entire domain. Therefore, this initial value problem can be considered as equivalent as an interpolation problem since the expansion coefficients at the initial time will be evaluated as a solution of the interpolation problem, as shown in (3.13). Thus, solving the PDEs with the parametrization of RBFs at the beginning is actually an interpolation problem that is equivalent to solving an initial value problem. The original time-dependent initial value problem thus becomes an interpolation problem for the initial values of the generalized expansion coefficients $\boldsymbol{\alpha}$. To advance the initial values $\boldsymbol{\alpha}$ along with time, a *collocation method* is introduced. In Eulerian approach, all the nodes of the fixed mesh are taken as the fixed knots of the RBF interpolation for the implicit function Φ. As an extension, (3.20) is then applied to every knot of the RBF interpolation, rather than only the points at the evolutionary front. The normal velocity v_n in (3.20) is thus extended as v_n^e to all the knots in the domain D. This is illustrated in Figure 2.5, for which each node is considered as a knot of the RBF.

We use collocation method together with the orthogonality constraints in (3.19), then a set of ODEs can be obtained:

$$\mathbf{A}\frac{d\boldsymbol{\alpha}}{dt} + \mathbf{B}(\boldsymbol{\alpha}) = 0, \tag{3,23}$$

where

$$\mathbf{B}(\boldsymbol{\alpha}) = \begin{bmatrix} v_n^e(\mathbf{x}_1)\left|\left(\nabla\boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x}_1)\right)\boldsymbol{\alpha}\right| \\ \vdots \\ v_n^e(\mathbf{x}_N)\left|\left(\nabla\boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x}_N)\right)\boldsymbol{\alpha}\right| \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{(N+4)\times 1}. \tag{3.24}$$

In fact, (3.23) is a collocation formulation of the method of lines, in which the PDE problem is reduced to a simpler ODE problem by discretization [43]. The method of lines has a solid mathematical foundation and the convergence of the solution of the converted ODE problem to the solution of the original PDE problem has been rigorously proven. In (3.24), the spatial derivative $\nabla\boldsymbol{\phi}$ can be found analytically from (3.13) & (3.14) due to the RBF interpolation [43].

The set of coupled non-linear ODEs of (3.23) can be solved by several different ODE solvers such as the first-order forward Euler's method and higher-order Runge-Kutta, Runge-Kutta-Fehlberg, Adams-Bashforth, or Adams-Moulton methods [39]. For simplicity, only the first-order forward Euler's method is implemented into the solution algorithm for ODE initial condition problems. An approximation to (3.23) can be given by

$$\boldsymbol{\alpha}(t^{n+1}) = \boldsymbol{\alpha}(t^n) - \tau\mathbf{A}^{-1}\mathbf{B}\left(\boldsymbol{\alpha}(t^n)\right), \tag{3.25}$$

where τ is the time step. It is necessary to have the step size small enough to achieve the numerical stability due to the Courant-Friedrichs-Lewy (CFL) condition [58] and to reduce the truncation error due to the variation in a single step of the decent gradient direction and

the velocity field in (2.13) in the level set-based topology optimization methods. Once the solution in (3.25) is found at each time step, the time-dependent shape and topology can be updated by using (3.18).

With this numerical strategy implemented in the RBF-level set optimization method, the evolution of the boundary of the shape along a descent gradient direction is in fact by advancing the scalar implicit function Φ in time by solving the system of coupled non-linear ODEs (3.23) and the optimal propagation of the front can be found by using the approximated solution in (3.25).

In the conventional level set method, the upwind scheme is not employed to advance the front in this parametric method. Because of the low accuracy of finite difference method, it does have a tendency to lose characteristics of the surface in under-resolved regions in each step [60] or unwanted dissipation of the front [84]. In order to revive the condition of the level set function , the reinitialization procedure is applied to the neighborhood of the front to guarantee a good approximation of the normal, i.e. $|\nabla\Phi| = 1$ or the curvature of the front. Unfortunately, the error induced by the reinitialization is likely to accumulate as the number of time steps increases. Iterative reinitialization schemes based on signed distance functions have the potential disadvantage in the relative low-resolution of the switch function based on checking the sign of the level set equation, for which may cause the front to move undesirably [76]. In addition, the reinitialization procedure is usually time-consuming [102]. Hence, reinitialization should be avoided as much as possible. In practice, whether reinitialization is appropriate depends on the underlying problem is interested in only the zero level set of function or the entire function Φ.

Reinitialization produces a serious problem in the existing level set

method that new holes cannot be created within the admissible domain [3, 16]. In the parametric RBF-level set method, spatial derivatives of the level set function can be obtained analytically and a good behavior of the normal or curvature of the front can be maintained due to the infinite smoothness of the MQ splines [15]. Furthermore, reinitialization is not performed and the entire level set function Φ is taken into account. Thus, the RBF level set model is capable of hole nucleation and elimination of the dependency of the final optimal solution on the design initiation [16]. As suggested by [76], possible problems such as loss of mass or movement of the zero level set without reinitialization can be avoided if an appropriate extension velocity method is adopted. In this parametric method, a straightforward and efficient extension velocity method is adopted.

3.4.1 Extension Velocity

As mentioned previously, the normal velocity $v_n(x)$ at the front must be extended in the Eulerian approaches. In this parametric method using collocation of lines, the normal velocity $v_n^e(x)$ as shown in (3.24) is the extension velocity, which is defined over the entire design domain D as $v_n^e(x)\colon D \rightarrow \mathbb{R}$. The choice of the extension velocity method is important because it can directly affect the overall efficiency of the level set method [63]. To guarantee an accurate and efficient time advancement of the front, $v_n^e(x)$ must be defined carefully.

There are many approaches to construct the extension velocity and the original level set method introduced by Osher and Sethian [58] was concerned with the interface problems with geometric propagation velocities and thus a natural construction of an extension velocity was obtained, in which a signed distance function was used as a level func-

tion due to its simplicity. In other applications, many extension velocity methodologies have also been developed. In fluid simulations, the fluid velocity was chosen as the extension velocity [84]. An approach using less physical quantity to build an extension velocity field was developed by Sethian and Strain [77], in which a numerical simulation of dendritic solidification with a jump condition across the interface was presented.

In the parametric RBF-level set method, a physically meaningful extension velocity method is proposed for the topology optimization based on the implicit function. In accordance with the shape derivative (3.3), a natural extension of the normal velocity can be obtained if the strain field is defined over the entire design domain D by assuming $\varepsilon(u) = 0,\ u \in (D \setminus \Omega)$. Since both the strain energy density inside the design domain and the constraint-related Lagrange multiplier are included, this extension velocity is physically meaningful.

In practice, this extension introduces a discontinuity in speed as it closes to the boundary. It is because the strain field is not continuous across the front. To ensure a smooth progress of the front, this discontinuity must be eliminated. The boundary itself is smooth and continuously differentiable because of the RBF implicit modeling [36], but the magnitude of the normal velocity at the boundary may not be continuous and smooth enough due to the finite element modeling involved in the strain analysis. Therefore, the magnitude of the normal velocity along the front must be regulated to allow for a stable propagation along a decent gradient direction. To perform all these operations, the boundary must be explicitly captured.

However, in the field of level set methods, it is well known that the boundary is implicitly represented in level set methods and that all of the method may be executed on the underlying mesh [76, 90].

To make full use of this feature, all the smoothing operations are performed in a narrow band region, rather than only along a front curve. A narrow band region around the zero level curve (front) is defined as $\Xi = \left\{ x \in \mathbb{R}^d \middle| |\Phi(x)| \leqslant \hat{\delta} \right\}$, where $\hat{\delta}$ is the bandwidth. The extension velocity in the narrow band is further improved by applying a simple linear filter, radially linear 'hat' kernel [12, 80], to achieve a good smoothing effect, which can be expressed as

$$\hat{v}_n^e(x) = k^{-1}(x) \sum_{p \in N(x)} W_c\left(\|p - x\|\right) v_n^e(x), \tag{3.26}$$

where

$$k(x) = \sum_{p \in N(x)} W\left(\|p - x\|\right), \tag{3.27}$$

$$W\left(\|p - x\|\right) = \tau_{\min} - \|p - x\|, \tag{3.28}$$

in which $N(x)$ is the neighborhood of $x \in \Xi$ in the filter window and $\tau_{\min}$ the window size. Thus, the overall extension velocity is defined as

$$v_n^e(x) = \begin{cases} -\left(\lambda - \varepsilon(u)^{\mathrm{T}} C \varepsilon(u)\right) & \forall x \in \mathbb{R}^d \,|\Phi\,(x) < -\hat{\delta} \\ \hat{v}_n^e(x) & \forall x \in \Xi \\ -\lambda & \forall x \in \mathbb{R}^d \,|\Phi\,(x) > \hat{\delta} \end{cases}. \tag{3.29}$$

It is clearly that the new extension velocity field used to move the level set function is closely related with the normal velocity suggested by the physics in the entire design domain and thus the extension velocity conveys all information about the physics.

3.5 Shape Sensitivity Analysis

In this analysis we simply evaluate the solution of the linear system as illustrated by (3.2) so as to find out the rate of change of the objective function against the design parameter, e.g. expansion coefficient α. In

the conventional discrete level set approach, the design variables are the discrete values of the level set function over the whole design domain.

In this RBFs based parametric method, the design variables are the expansion coefficient α instead. In the corresponding interpolation scheme, x_i the positions of the knots are fixed in space, i.e. the nodes of the Eulerian grid, the space and time can then been completely separated.

THEOREM 3.2 *Lemma 2.2 shows that the Hamilton-Jacobi equation* (2.13) *is only time-dependent, the level set function* $\Phi(x,t)$ *can now be represented by N RBFs centered at N knots.*

Proof As time and space are separable, the time-dependent Φ is related to the generalized expansion coefficient α of the RBF interpolant as follows

$$\Phi(x,t) = \sum_{i=1}^{N} \alpha_i(t)\varphi_i(x), \tag{3.30}$$

and substitute (3.30) into (2.13) yields

$$\sum_{i=1}^{N} \dot{\alpha}_i \varphi_i(x) - v_n |\nabla\Phi| = 0. \tag{3.31}$$

Then, we now have the shape equation presented in parametric form.

The time-dependent property behaves only on the updating of the expansion coefficients α during the minimization process. If the Karush-Kuhn-Tucker conditions (KKT) is satisfied, there exist a local minimum that satisfies some regularity conditions. Consequently, we are only necessary to solve a set of algebraic equations for the optimal solution, rather than dealing with the boundary value PDEs [91] or ODEs with extension velocity [93].

3.5.1 Shape Sensitivity Algorithm

From (3.4) we can deduce the normal velocity v_n on the moving boundary by rewriting (3.31):

$$v_n = \frac{1}{|\nabla\Phi|}\sum_{i=1}^{N}\dot{\alpha}_i\varphi_i(x). \tag{3.32}$$

THEOREM 3.3 *In Lemma 3.1, the boundary integral is independent of the rate of change of the expansion coefficient $\dot{\alpha}_i$ which can be taken out from the integration. Then, the shape derivative $\frac{\partial J}{\partial \alpha_i}$ can be obtained by applying chain rule to the time derivative of the Lagrangian $\frac{dL}{dt}$.*

Proof Substitute (3.32) into the shape derivative (3.3)

$$\frac{dL}{dt} = \sum_{i=1}^{N}\int_{\partial\Omega}\left(\lambda - \varepsilon(u)^{\mathrm{T}}\mathrm{C}\varepsilon(u)\right)\left(\frac{1}{|\nabla\Phi|}\dot{\alpha}_i\varphi_i(x)\right)ds,$$

as the rate of change of expansion coefficient $\dot{\alpha}$ is independent to the boundary integral, it can be taken out from it and then we get

$$\frac{dL}{dt} = \sum_{i=1}^{N}\dot{\alpha}_i\int_{\partial\Omega}\left(\lambda - \varepsilon(u)^{\mathrm{T}}\mathrm{C}\varepsilon(u)\right)\frac{1}{|\nabla\Phi|}\varphi_i(x)\,ds. \tag{3.33}$$

Since L is dependent on t through $\dot{\alpha}_i$, we can derive the following expression by chain rule

$$\frac{dL}{dt} = \sum_{i=1}^{N}\frac{\partial L}{\partial\alpha_i}\cdot\dot{\alpha}_i = \sum_{i=1}^{N}\frac{\partial J}{\partial\alpha_i}\cdot\dot{\alpha}_i + \lambda\sum_{i=1}^{N}\frac{\partial V}{\partial\alpha_i}\cdot\dot{\alpha}_i. \tag{3.34}$$

As shown in (3.33) and (3.34), the sensitivities of the objective function and the volume constraint can be rapidly obtained by comparing them in the following fashion:

$$\frac{\partial J}{\partial\alpha_i} = -\int_{\partial\Omega}\varepsilon(u)^{\mathrm{T}}\mathrm{C}\varepsilon(u)\frac{1}{|\nabla\Phi|}\varphi_i(x)\,ds, \qquad i = 1,\ldots,N. \tag{3.35}$$

$$\frac{\partial V}{\partial \alpha_i} = \int_{\partial\Omega} \frac{1}{|\nabla\Phi|}\varphi_i(x)\,ds, \qquad i = 1, ..., N. \tag{3.36}$$

The proof is completed.

Therefore, we have successfully converted the general minimization problem into its parametric form. The sensitivity analysis resolves the shape and topology optimization problem in the level set framework through mathematic programming instead of solving the tedious PDEs or the costly ODEs system. The sensitivity algorithms determine the search direction in terms of the parameters and there are numerous numerical schemes for solving the equations, such as the steepest descent method and the method of moving asymptote (MMA) [93, 94].

3.5.2 Other Benefits from Sensitivity Analysis

In the sensitivity analysis framework, many sophisticated and well-established optimization algorithms driven by the proposed method can be used to solve the optimization problem. The CFL condition applied onto the step size in stabilizing the upwind scheme can be totally neglected. In [12,65], the step size is also a factor for stabilizing the optimization process, however, the principle is completely different from that of CFL condition. The issue of the CFL condition is merely related to the effect of the mesh size on the finite difference scheme, in contrast, the sensitivity analysis is insensitive to the size of the mesh or grid. It is clear that the sensitivity method is potential in handling the realtime engineering problems in large-scale systems and the numerical models will be expected to be accurate.

In convectional level set-based implicit shape structural optimization, the normal velocity field at the design boundary is explicitly calculated by using the steepest-gradient descent method, which is the clas-

sical approach for the unconstrained programming that chooses the reduction heading of the negative gradient as the downhill direction [50]. It is characterized with the conceptual simplicity, but it zig-zags its way towards the optimal solution because each subsequent searching direction is orthogonal to the previous one. In each iteration, the objective function around one particular point might be locally descent fast in terms, but the zigzag cannot ensure a globally steepest descent of the objective function but a crooked searching. At the start of the iterations, the descent of the objective function may be fast, but when the point is approaching to the optimum solution, the convergent speed will become slower and slower.

In the method of the sensitivity analysis, it is no longer need to calculate the velocities of all knots explicitly because they are substituted into the subsequent equation as an alternative way to evaluate the tendency for the optimal design, so that the application of the extension of the velocity [91] and the velocity smoothness schemes [93] are all unnecessary.

Conclusively, the movement of the design boundary is now governed by the rate of change of the expansion coefficient $\dot{\alpha}$ which is directly determined from the sensitivities of the objective and constraint functionals at all knots. In the next chapter, we discussed the sensitivity analysis intensively in term of its efficiency and accuracy. The effect of the knot distribution will be studied and POU will be introduced to improve the overall quality of the optimization algorithm. Details of the numerical implementation will be listed and the results will be given as the evidence of the improvement in CHAPTER 4.

□ **End of chapter.**

Chapter 4

RBFs with Dynamic Knots

In the conventional level set method, Φ must be discretized through a distance transform on a rectilinear grid. The numerical schemes such as upwind schemes, extension velocities and reinitialization algorithms must be applied; however, these may limit the capability of the level set method. For example, the process of the reinitialization forbids a level set function from nucleation of new holes in the interior regions of the material [4,16]. The other disadvantages are mainly raised from its discrete representation. Since the grid is *fixed* in space, only the nodal values of the implicit function $\Phi(x)$ are used directly.

In CHAPTER 3 we have already resolved the some of the above problems by parametrization [41,93,94], however, the spatial issue from using the fixed mesh is not completely addressed by the authors. For the parametric sensitivity method, the use of the globally supported basis functions means that the boundary integrals of (3.35) and (3.36) must be carried out for each RBF, a total of $N \times 2$ times. For a large N (e.g., 10^9), these integrals are hardly efficient to compute. This leaves a great doubt in the effect of the knot distribution to come into both the quality and efficiency of the optimization process in practice.

4.1 Dynamic Knots for Optimal Distribution

To tackle this complexity problem, compactly supported RBFs (CS-RBFs), as shown in Figure 3.1 (e) & (f) may be employed instead. They would result in band diagonal and sparse coefficient matrix and can noticeably improve the efficiency in the sensitivity computations of (3.35) and (3.36) [73,95,96]. But the limited range of the local support of the basis functions demands a denser set of the basis functions to be distributed in the design domain, substantially increasing the number of RBFs and hence compromising the computational efficiency gained in using the local supports. A modest number of CS-RBFs may give rise to irregular boundary shapes in the optimal structure obtained [94].

Therefore, the greatest benefits of the parametric level set method lie in reducing its numerical complexity while maintaining the fidelity of shape and topology representation. An important potential of the RBF parametric method is that we have more freedom to choose design variables and to control more parameters with the sensitivity analysis and that we have potentially more efficient strategies for the interpolation of the level set function with the globally supported basis functions. These developments, and their combinations, would make the RBF level set representation a truly powerful technique with significant efficiency in the numerical computations. The aim of this proposed research is to generalize the RBF parametric technique for the structural shape and topology optimization. Central to the research development are two key techniques: (1) a new parametric scheme of dynamic knots and (2) a use of the partition of unity (POU) scheme which will be discussed in next chapter in detail.

4.2 RBFs with Dynamic Knots

In the following contents, the first goal is to devise a numerical scheme for an optimal distribution of knots of the radial basis functions. Radial basis functions provide a powerful method for interpolation of the level set function. The concept is rooted to their use for multivariate interpolation to scattered data [74, 88, 96]. So, we introduce a scheme that can make use of the knot position as one of the design variables in the optimization problem. In this way, we can easily to move around the knots inside the domain where the topology changes are more rigorous. We choose IMQs as the implicit representation of the level set function Φ because of its stability and high sensitivity, which can be written as

$$\varphi_i(x) = \frac{1}{\sqrt{(x - x_i)^2 + c_i^2}}, \tag{4.1}$$

We assume the expansion coefficient as a constant so that the formulation of the RBFs interpolants (3.30) can be re-written as

$$\Phi(x, t) = \sum_{i=1}^{N} \alpha_i \varphi_i(x, t), \tag{4.2}$$

Our idea is inspired by the work in [54, 61] to consider the variation of the knots for the purpose of improving the performance of RBF interpolation to scattered data. But, unlike in [54, 61], our problem is not static, but rather dynamic. So, we need optimally distributed RBF knots in the design domain $\Omega(t)$ during the entire dynamic evolution process of the level set function $\Phi(x, t)$ changing in pseudo-time t. Thus, the knot positions must be dynamic as well, i.e., $x_i = x_i(t)$. In our dynamic knots scheme, we would regard the positions x_i of knots as design variables, besides the expansion coefficients α_i.

Remark We can modify THEOREM 3.2 and THEOREM 3.3 to enable the movement of the knot position and hence the knot can move towards the position where the change is more critical as determined by the shape sensitivity. In such case, we may use fewer number of knots to describe the properties of the system while the smoothness of the implicit function is satisfied.

THEOREM 4.1 *In the following we assume the expansion coefficient* α_i *as a constant throughout the optimization process; however, the positions of the knots* x_i *are now changed with time and* (4.1) *can be modified as*

$$\varphi_i(x,t) = \frac{1}{\sqrt{(x - x_i(t))^2 + c_i^2}}, \tag{4.3}$$

hence, (3.31) *is rewritten in a new form*

$$\sum_{i=1}^{N} \alpha_i \frac{\partial \varphi_i(x,t)}{\partial x_i} \cdot \dot{x}_i - v_n |\nabla \Phi| = 0, \tag{4.4}$$

where

$$\frac{\partial \varphi_i}{\partial x_i} = \left[\frac{1}{\sqrt{(x - x_i)^2 + c_i^2}} \right]^3 (x_i - x). \tag{4.5}$$

Then, the normal velocity in term of the rate of change of the knot position $\dot{x}_i$ *on the free boundary is*

$$v_n = \frac{1}{|\nabla \Phi|} \sum_{i=1}^{N} \alpha_i \frac{\partial \varphi_i(x,t)}{\partial x_i} \cdot \dot{x}_i. \tag{4.6}$$

In the following step, we substitute (4.6) *into* (3.3) *to obtain the shape derivative in term of* $\dot{x}_i$

$$\frac{dL}{dt} = \sum_{i=1}^{N} \alpha_i \int_{\partial\Omega} \left(\lambda - \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) \right) \frac{1}{|\nabla \Phi|} \frac{\partial \varphi_i(x,t)}{\partial x_i} ds \cdot \dot{x}_i. \tag{4.7}$$

Since L is still dependent on t through x_i, we can derive the following expression by chain rule again

$$\frac{dL}{dt} = \sum_{i=1}^{N} \frac{\partial L}{\partial x_i} \cdot \dot{x}_i = \sum_{i=1}^{N} \frac{\partial J}{\partial x_i} \cdot \dot{x}_i + \lambda \sum_{i=1}^{N} \frac{\partial V}{\partial x_i} \cdot \dot{x}_i. \tag{4.8}$$

As shown in (4.7) *and* (4.8)*, the sensitivities of the objective function and the volume constraint can be rapidly obtained by comparing them in the following fashion:*

$$\frac{\partial J}{\partial x_i} = -\alpha_i \int_{\partial\Omega} \varepsilon(u)^{\mathrm{T}} \mathrm{C} \varepsilon(u) \frac{1}{|\nabla\Phi|} \frac{\partial \varphi_i(x,t)}{\partial x_i} \, ds, \qquad i = 1, ..., N. \tag{4.9}$$

$$\frac{\partial V}{\partial x_i} = \alpha_i \int_{\partial\Omega} \frac{1}{|\nabla\Phi|} \frac{\partial \varphi_i(x,t)}{\partial x_i} \, ds, \qquad i = 1, ..., N. \tag{4.10}$$

Proof This result is a straightforward corollary of THEOREM 3.2 and THEOREM 3.3.

4.3 Optimization Algorithm

The knot positions are eventually permitted to change in each iteration of the design optimization. Their changes are determined through the design sensitivity analysis and are guided by the optimization process. With the RBF representation of the level set function, we can directly obtain the sensitivity of the objective and constraint functionals with respect to x_i (i.e. $\frac{\partial J}{\partial x_i}$, $\frac{\partial V}{\partial x_i}$), respectively.

To find the local minimum of the objective function as stated in (3.1), we make use of the steepest descent method to proceed with the search in the descent direction of the sensitivity functions at the current point as shown in (4.9) & (4.10). Normally, the gradient descent method is only suitable for the unconstrained optimization problem as

the discrepancy of the gradients between the objective and the constraint functionals may be large. However, the lagrange multiplier reduces the gap of the gradient difference and balances the effects caused by them. As a result, the descent method can be applied in this situation correctly.

From (4.8) the search direction can be defined as

$$\dot{x}_i = -\left(\frac{\partial J}{\partial x_i} + \lambda \frac{\partial V}{\partial x_i}\right), \qquad i = 1, ..., N, \tag{4.11}$$

and the position of knots can be updated

$$x_i^{n+1} = x_i^n + \tau \dot{x}_i, \qquad i = 1, ..., N, \tag{4.12}$$

where τ is the time step and it is fixed as a positive constant. In fact, the time step is chosen under the Courant-Friedrichs-Lewy condition (CFL condition) which is a necessary condition for convergence while solving certain partial differential equations (usually hyperbolic PDEs) numerically. For example, if a boundary is crossing a discrete grid, then the time step must be less than the time for the boundary to travel adjacent grid points. In practice, the time step varies from one model to another and hence some trials are necessary to determine the suitable time step by the actual experience.

Once the knots position is updated as x_i^{n+1}, we can calculate the corresponding new basis function φ_i^{n+1} by (4.3)

$$\varphi_i^{n+1} = \frac{1}{\sqrt{\left(x - x_i^{n+1}\right)^2 + c_i^2}}, \tag{4.13}$$

and hence the updated level set function is

$$\Phi^{n+1} = \sum_{i=1}^{N} \alpha_i \varphi_i^{n+1}, \tag{4.14}$$

thus, the implicit function is computed everywhere in D and then the boundary of the structure is propagated accordingly.

Algorithm 1 Our proposed algorithm is an iterative method and implemented as follows:

1: Choose initial $\Phi_0 \in D$, set $n = 0$.

2: **repeat**

3: Compute $u^n \in \Omega^n$.

4: Compute $\varphi^n \in \Omega^n$, i.e. solve the RBF equation (4.3).

5: Solve the descent gradient $\frac{\partial J}{\partial x_i}$, $\frac{\partial V}{\partial x_i}$ by solving (4.9) & (4.10).

6: Evaluate the search direction $\dot{x}_i$ in (4.11).

7: Update the position of knots x_i^{n+1} by (4.12).

8: Update the RBF value φ^n on each knot by (4.13).

9: Update the level set function Φ^{n+1} by (4.14).

10: Update the shape Ω^{n+1}.

11: Increment the iteration n.

12: **until** $\|J(\Phi^{n+1}) - J(\Phi^n)\| < TOL$.

The new shape Ω^{n+1} is characterized by the level set function Φ^{n+1} through the time step $\triangle t_\tau$, it is chosen in the way that

$$J(\Omega^{n+1}) \leqslant J(\Omega^n). \tag{4.15}$$

Here, we may also update the expansion coefficients α_i and the positions of the knots x_i simultaneously. As the implicit interface is related to both the parameters, this methodology will provide more accuracy to regulate the propagation of the boundary during optimization. However, the sensitivities of both parameters are greatly different

in scale so that we must be careful to choose the time steps to avoid discontinuous data happening in either update scheme.

In this study, we present the simpler process with the update scheme of changing knot position and steepest descent method to illustrate the capability of the parametric optimization, however, it is easily extended to the more complicated algorithm such as MMA and OC methods.

4.4 Numerical Results

Here, we present illustrative numerical examples throughout the spatial discretization which is done with piecewise bilinear finite elements on quadrilateral cells. In Section 4.4.2, we show the results using the variant of the optimization loop building up the entire shape function as described in Algorithm 1 and the results of the other variant are mainly the same.

We consider the minimum compliance problem with the volume constraint in Section 4.4.1 on the space-time domain. Assume the Young's modulus $E = 1$ for solid material, $E = 10^{-3}$ for void material (i.e. holes) and Possion's ratio $\nu = 0.3$. The "ersatz material" approach is adopted because it is simple but effective in topology optimization [4].

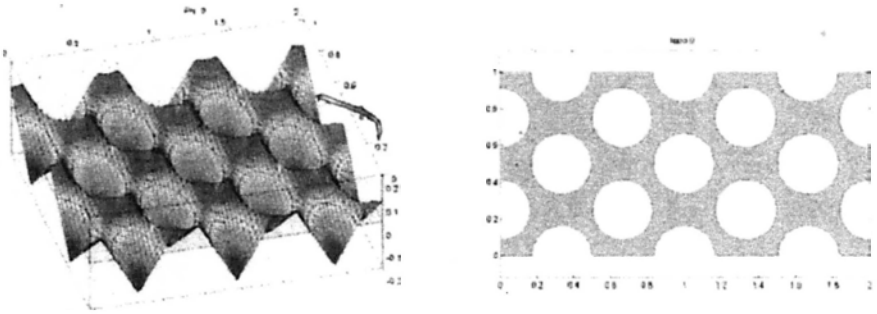
The positions of the knots initially coincide to the corresponding nodes of the mesh. As mentioned in Section 4.2, we only employ IMQs function in this numerical example and the shape parameter c_i will influence the final optimization result. Some studies have been done on choosing the optimal c_i but it is not the issue in this paper. Therefore, we select the shape parameter $c_i = h$, where h is the size of the finite element.

In order to satisfy the volume constraint, we employ the augmented Lagrangian method (ALM) to evaluate the Lagrange multiplier λ. The

iteration is terminated when the difference of two successive Lagrangian values is less than the prescribed tolerance $TOL = 10^{-6}$ or when the maximum number of iterations reaches the prescribed limit, which is set to 500 steps in this study.

All computations are done with the software packages *MATLAB*. The shape is obtained by plotting the contour of the height matrix calculated from the sum of the radial basis functions on all the nodes of the prescribed domain, i.e. $\Phi(x)$ with the matrix size of 81×41. For the detail of the contouring algorithm, readers may refer to the documentation of *MATLAB* in the section: *Graphics/Contour Plots*.

As shown in Figure 4.1(b), the shape is presented in term of the 2D contour of the level set function. The z-value of the current contour level is set to be $\Phi(x) = 0$ where is the location of the boundary, as it can be seen in Figure 4.1(a). The positive z-values represent the interior of the shape and vice versa.



(a) 3D Contour plot of $\Phi(x)$ (b) 2D Contour plot of $\Phi(x)$

Figure 4.1: Contour plot of level set function $\Phi(x)$ in 2D and 3D

4.4.1 Test problem

Short cantilever beam

The problem in CASE 1 & 2 is a short cantilever beam ($L : H - 2 : 1$) with a fixed boundary on the left, where no displacement is allowed as shown in Figure 4.2. A point load $P = 1$ is applied at the middle of the right end. The design domain is divided into a rectilinear mesh as 80×40 and the size of the finite element is $h = 0.025$, the volume fraction $\zeta = 0.5$ and the time step $t_\tau = 5 \times 10^{-3}$.

MMB beam

In order to show the extendability of the dynamic knots method, we trial it with the MMB beam with the test configurations which are different from the short cantilever beam used in CASE 1 and CASE 2. The problem in CASE 3 is a MBB beam ($L : H = 4 : 1$) with its left corner of the bottom is fully constrained, while the right corner is supported vertically and allowed to move horizontally as can be seen in Figure 4.3. A point load $P = 1$ applied at its middle top surface. The design domain is divided into a rectilinear mesh as 160×40 but only the righthand side of the domain is computed due to its structural

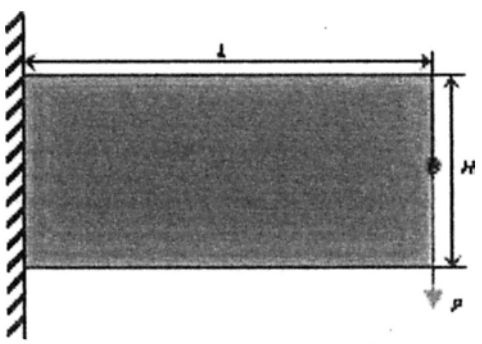


Figure 4.2: Design domain – Cantilever beam with dimension $L : H - 2 : 1$.

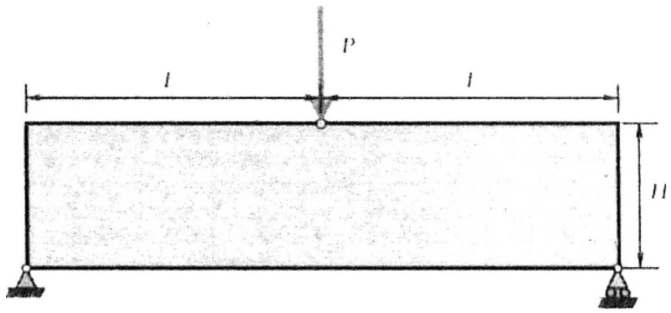


Figure 4.3: Design domain – MMB beam with dimension $L : H = 4 : 1$.

symmetry. The size of the finite element is $h = 0.025$, the volume fraction $\zeta = 0.5$ and the time step $t_\tau = 5 \times 10^{-3}$.

4.4.2 Computations of RBF Dynamic knots

CASE 1 – Optimization without optimal knot distribution

In this example, we discuss the optimal problem with fully loaded RBF knots on the grid, $81 \times 41 = 3321$ knots. Supposedly, it should give the maximum resolution of the shape representation and provide the best possible numerical stability to the simulation. However, the result shows that the over-populated knot density not only slows down the computation process but also prohibits some knots from moving to a stable position. The knots will move to the positions where the resultant strain energy density is the minimum, i.e. the minimum Lagrangian value. Consequently, the excess knots will remain on the undesirable positions that will eventually reduce the speed of the convergence.

As shown in Figure 4.4, the values of Lagrangian and mean compliance are rapidly converged to the constants in the first 30 steps, and these constants almost remain unchanged in the following iterations. However, we can see from the plot of the volume ratio (i.e. volume

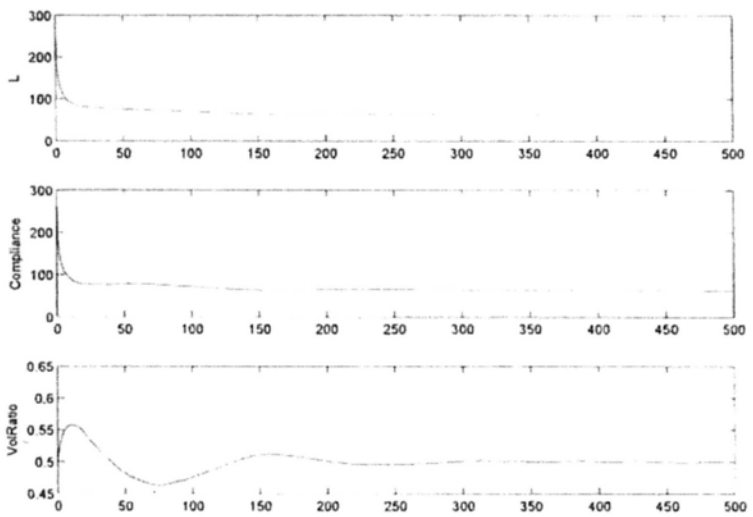


Figure 4.4: Convergence history of CASE 1

constraint) that the fluctuation is persisting until the end of the 300 steps.

In Figure 4.8, one can notice that the shape does not have significant change after 100 steps. In fact, the excessive number of knots creates a dense interpolation matrix in the RBF optimization that slows down the convergence of the local optimal shape, e.g. in the middle of the left hand side of the beam – the small 'island'. The average time consumed per iteration = 12.5 seconds and the solution cannot converge after 500 steps.

CASE 2 – Optimization with optimal knot distribution

In this case, we discuss the optimal problem with various loaded RBF knots on the rectilinear mesh. The numbers of the knots are tested in the following fashion: (a) $81 \times 21 = 1701$, (b) $41 \times 41 = 1681$ and (c) $41 \times 21 = 861$. From the observation, the smoothness of the

evolving boundary is still satisfied and the numerical stability remains unchanged, even though the resolution of the shape representation is reduced.

The results shows that the optimally-fitted knot density expedite the movement of the knots and significantly lower the computational cost because of the lesser knot number. All of the knots can readily move to the stable position as there is plenty of room favorable for their flow. Thus, the resultant strain energy density can reach the minimum in the least possible timeframe. The best result is found to be CASE 2c – the smallest possible knots size in this experiment.

As shown in Figure 4.5 (CASE 2c), the values of Lagrangian and mean compliance are immediately converged to the minimum level within the first 30 steps. At the same time, we can see from the plot of the volume ratio (i.e. volume constraint) that the variation has reached its end after 80 steps. No distinctive variation can be observed after 100 steps and even the solution is converged eventually under a harshest criteria (i.e. $TOL \leqslant 10^{-6}$).

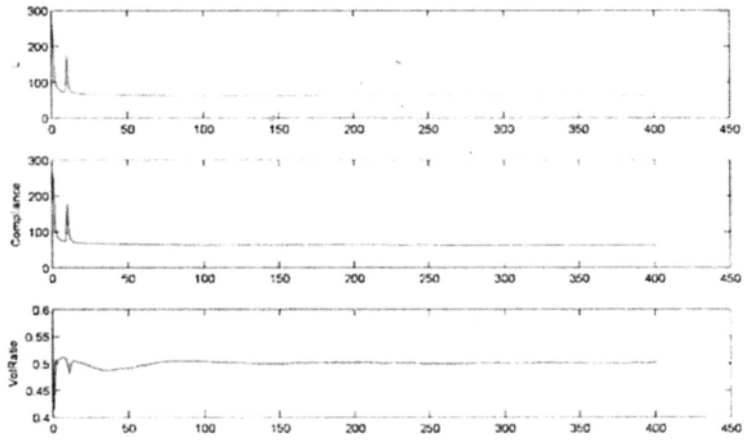


Figure 4.5: Convergence history of CASE 2c (41×21)

Table 4.1: Comparison of results for CASE 1 and CASE 2.

CASE No.	Knots size	No. of Knots	Time/Step (s)	Iteration	Compliance
1	81 × 41	3321	12.5	500	62.4
2a	81 × 21	1701	10.2	500	62.2
2b	41 × 41	1681	10.0	500	61.6
2c	41 × 21	861	8.6	402	60.8

In Figure 4.11, it is obvious that the shape has almost reached its optimal form within the first 100 steps. The knots readily move to its favorable positions and this help to remove the unwanted material from the domain in a few iterations. All the separated islands have been minimized instantly and they are completely eliminated after 200 steps. The average time consumed for each iteration has been reduced dramatically to 8.6 seconds and finally the converged solution is obtained at 402 steps.

In Table 4.1, we note that the significant improvement brought by the optimally-distributed knot algorithm. Apart from the impressive computational advantages, the final results generated from the set of CASE 2 are more accurate than the one evaluated from CASE 1. We remark that not even the higher quantity of knots will result in a better solution, it will jeopardize both the speed and the quality of the simulation. This fact implies that the parametric topology optimization can be beneficial by dividing the domain into many subsets for independent calculations. In the next section, we will investigate the possibility of the proposed philosophy and extend it into formulations.

Table 4.2: Comparison of results for CASE 3a and CASE 3b.

CASE No.	Knots size	No. of Knots	Time/Step (s)	Iteration	Compliance
3a	81×41	3321	12.6	500	77.8
3b	41×21	861	9.4	256	76.2

CASE 3 – Test example of MMB beam

It is similar to the previous cases that the fully-loaded RBF knots CASE 3a ($81 \times 41 = 3321$ knots) does not show the better results in both the quality and efficiency than CASE 3b ($41 \times 21 = 861$ knots). The problem of the over-crowded knot density in CASE 1 repeats again here in CASE 3a and the calculation speed is slowed down by the excessive knots in the domain. The details of the model settings are listed in Table 4.2.

As shown in Figure 4.6 (CASE 3a), the values of Lagrangian and mean compliance are gradually decreased to the constants after 150 steps. There are no significant changes for these constants afterwards. In Figure 4.12, one can notice that the change of the shape slow down significantly after 200 steps, there are some unwanted materials still remained on the final shape at step 500. The average time consumed per iteration = 12.6 seconds and the solution does not converge after finishing 500 steps.

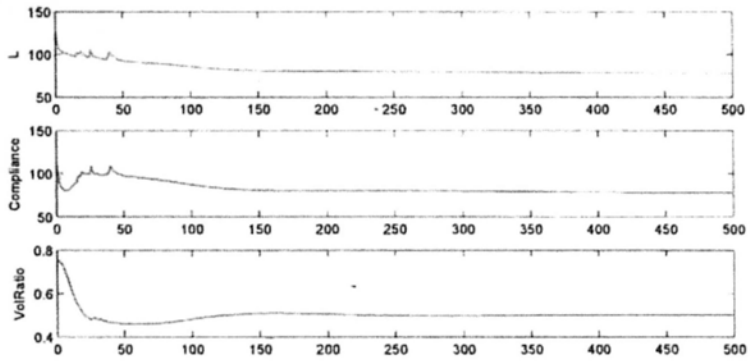


Figure 4.6: Convergence history of CASE 3a (81 × 41)

As shown in Figure 4.7 (CASE 3b), the values of Lagrangian and mean compliance are immediately converged to the minimum level within the first 60 steps. At the same time, we can see from the plot of the volume ratio (i.e. volume constraint) that the variation has reached its end after 100 steps. No distinctive variation can be observed after 150 steps. In Figure 4.13, the shape has almost reached its optimal form within the first 80 steps and the smaller number of knots enables each individual knot carrying higher normal velocity from the same sensitivity and this help to evolve the boundary of the shape further and faster in each iteration. All the unwanted materials have been completely eliminated after 100 steps. The average time consumed for each iteration has been reduced dramatically to 9.4 seconds and finally the converged solution is obtained at 256 steps.

4.5 Summary of Dynamic Knots Scheme

It is known that there is an intimate relation between the numerical stability and the approximation accuracy of the interpolation method. Generally speaking, the interpolation's approximation quality, numer-

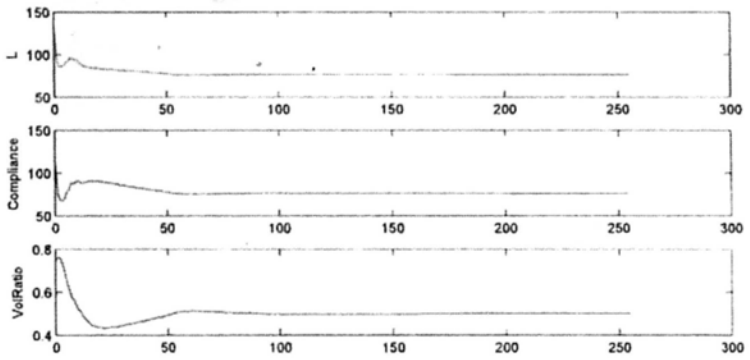


Figure 4.7: Convergence history of CASE 3b (41×21)

ical efficiency and stability all depend on the density and the distribution of the radial basis functions in the representation domain [61]. In view of numerical efficiency and stability, the separation distance between the knots cannot be too small. On the other hand, a good approximation of the level set function (thus, an accurate representation of the structure's shape) requires the separation distance suitably fine. Obviously, there must be a trade-off between these two criteria in determining how to distribute the radial basis functions, especially when simply using a uniform grid as in the common practice. It is even harder to define an optimal distribution of RBF knots in the design domain.

For our problem of structural topology optimization. The structure being optimized changes its shape in each update during the optimization process. It is not possible to use a fixed set of basis functions of reasonably sparse separations to represent the level set function with good approximation accuracy. Therefore, a relatively dense grid of knots is commonly used, at the expense of numerical efficiency and convergence rate in the optimization algorithms. The situation is similar to the use of radial basis functions for solving partial differential

equations [20, 73], where the choice of the RBF set can have a direct impact on the numerical solution. In addition, some of the knots move to the outer bound of the domain where contains no structural information of the shape. However, it will not deteriorate the accuracy because the resultant level set function $\Phi(x)$ is actually calculated from the fixed sampling points, i.e. the nodes on the fixed rectilinear grid. As in (4.2), each RBF $\varphi_i(x)$ interpolates with all the nodes and then gives the result in respect to its nodal position x on the fixed grid.

In this chapter we have successfully developed a dynamic knots method to replace the common practice of uniform grid distribution of knots and hence we can reduce the number of the radial basis functions in approximating the level set function. A significant reduction in the number of RBFs would increase the numerical efficiency, while a non-uniform distribution of the knots would help maintain the approximation quality and enhance the numerical stability. Apart from the improvement of the computational speed, the overall quality and the rate of the convergence are increased due to the higher maneuverabilities of the knots inside the domain. As it can be seen from the results shown in CASE 2c and CASE 3b, the computational cost has been dramatically decreased and it enables the possibility for solving the realtime engineering problems in the way of the structural optimization by the robustness of the level set method.

□ **End of chapter.**

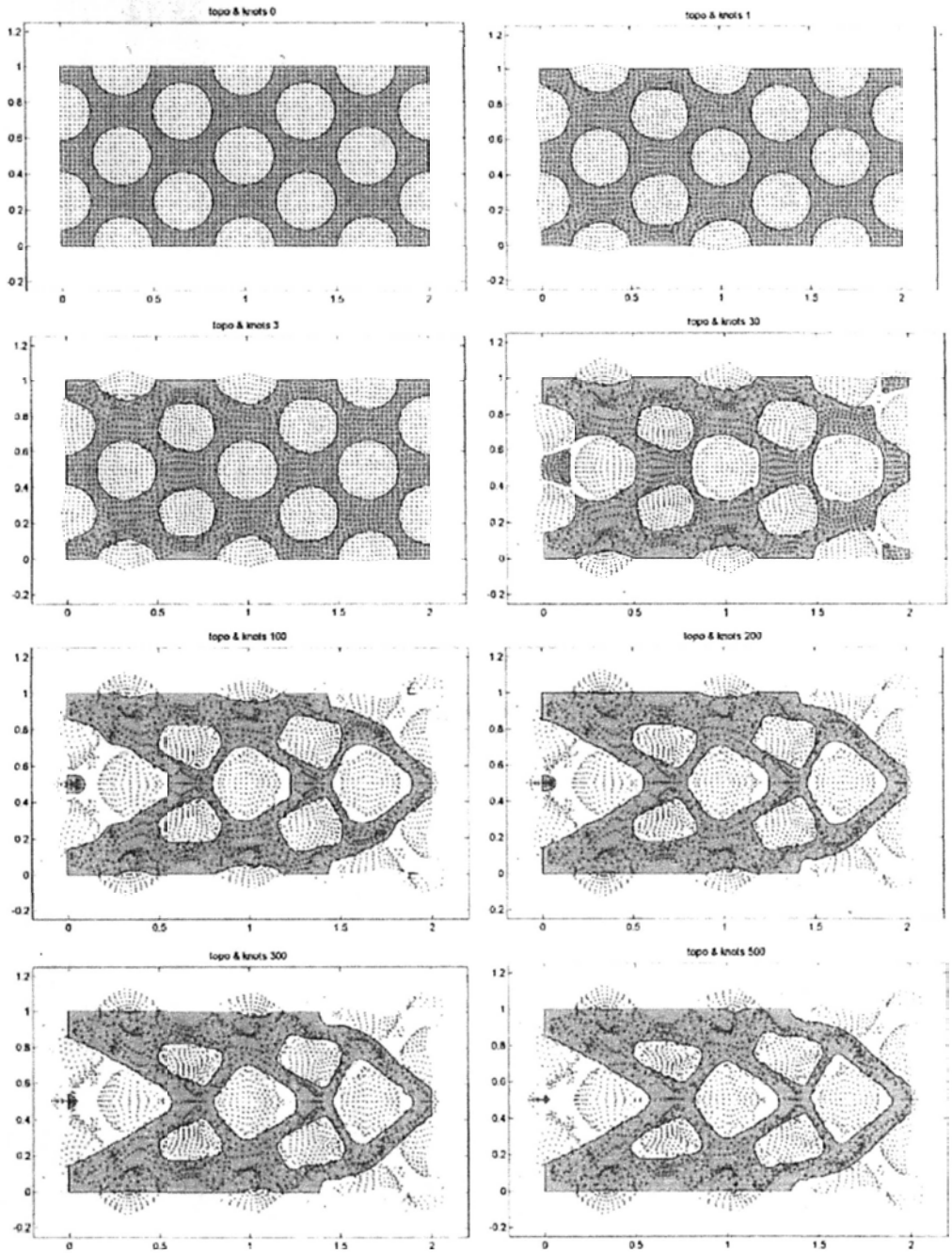


Figure 4.8: Optimization steps of CASE 1: $81 \times 41 = 3321$ knots, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 500 steps, time consumed per iteration = 12.5 second.

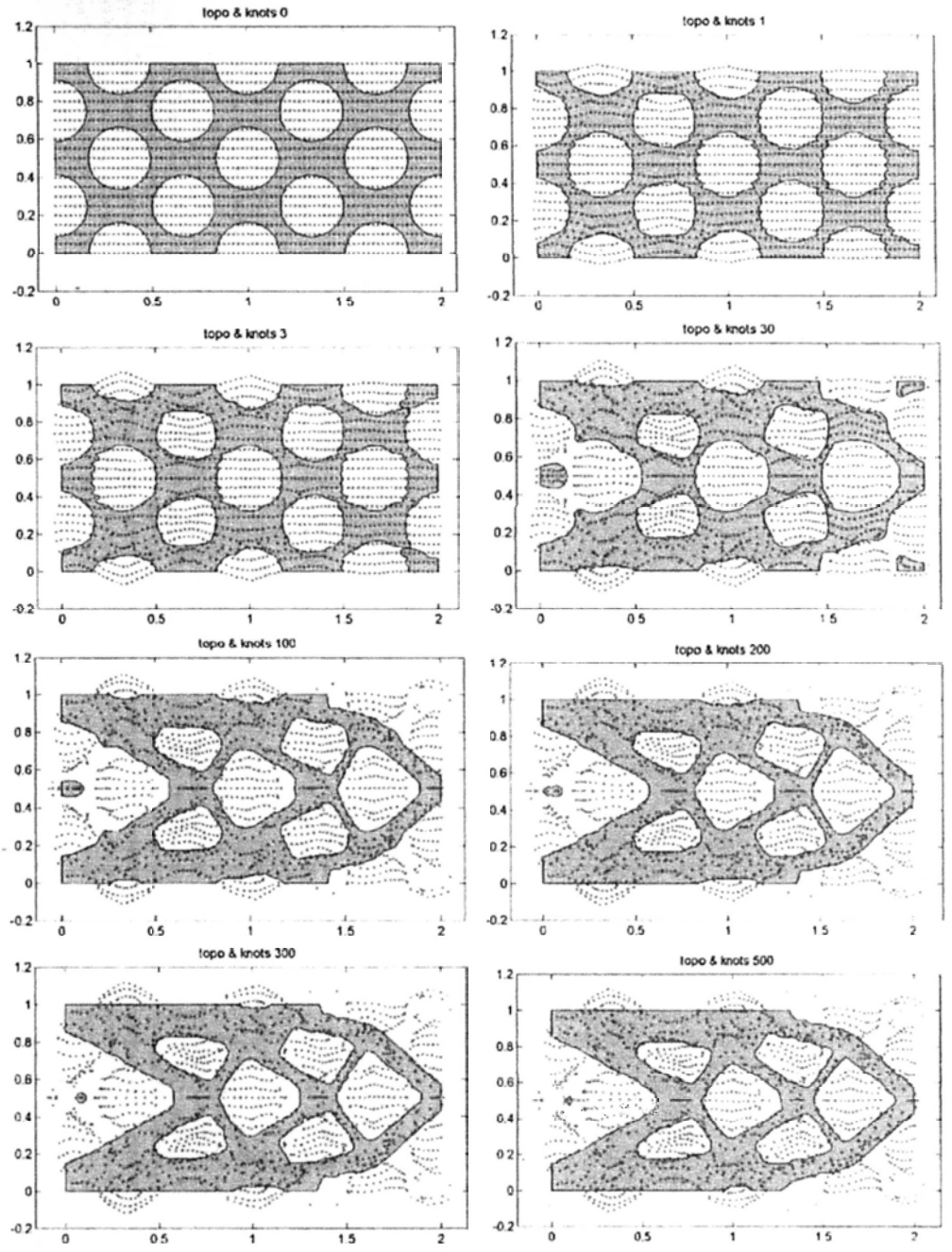


Figure 4.9: Optimization steps of CASE 2a: $81 \times 21 = 1701$ knots, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 500 steps, time consumed per iteration = 10.2 second.

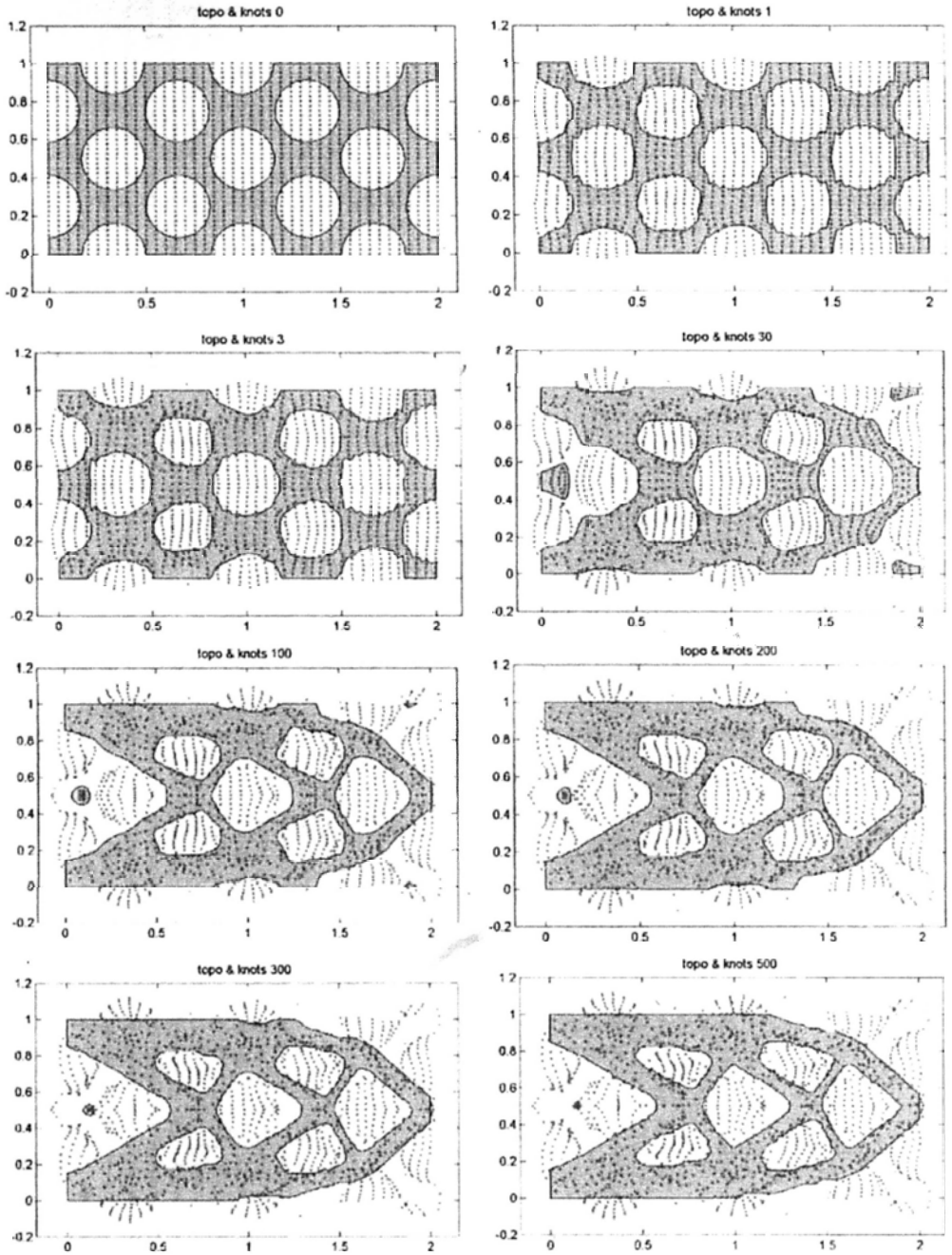


Figure 4.10: Optimization steps of CASE 2b: $41 \times 41 = 1681$ knots, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 500 steps, time consumed per iteration = 10.0 second.

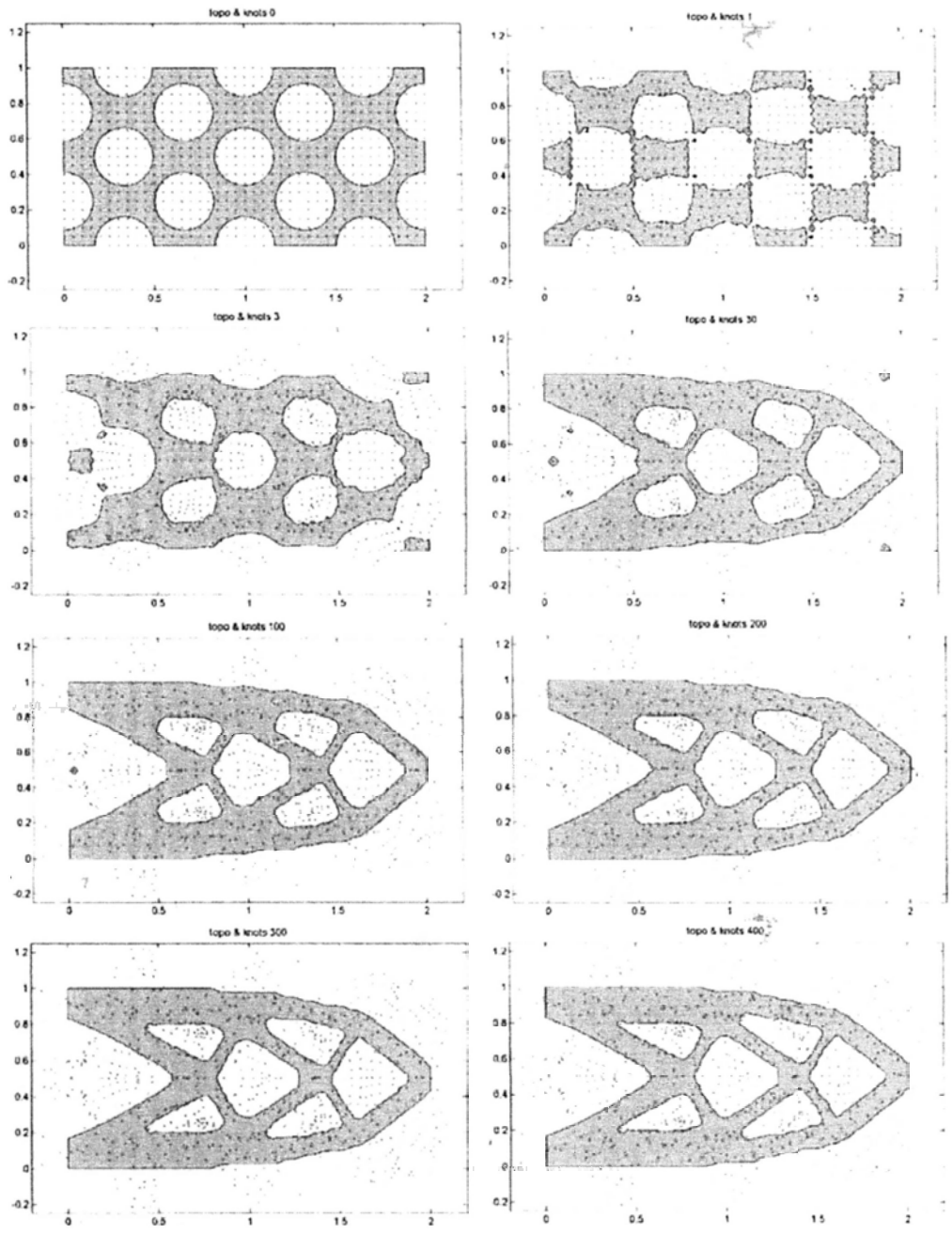


Figure 4.11: Optimization steps of CASE 2c: $41 \times 21 = 861$ knots, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, converged at Step 402, time consumed per iteration = 8.6 second.

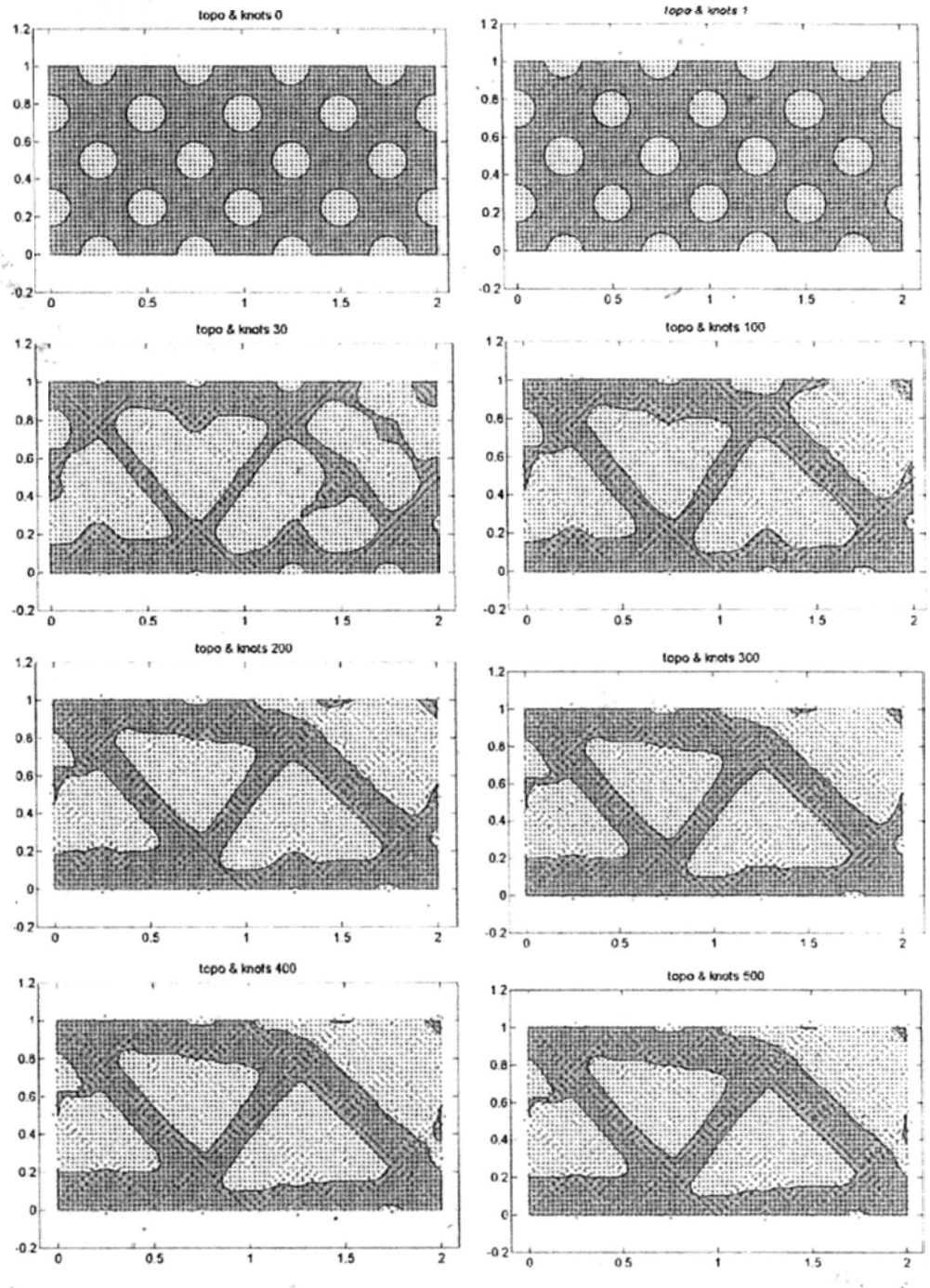


Figure 4.12: Optimization steps of CASE 3a: $81 \times 41 = 3321$ knots, $t_r = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 500 steps, time consumed per iteration = 12.6 second.

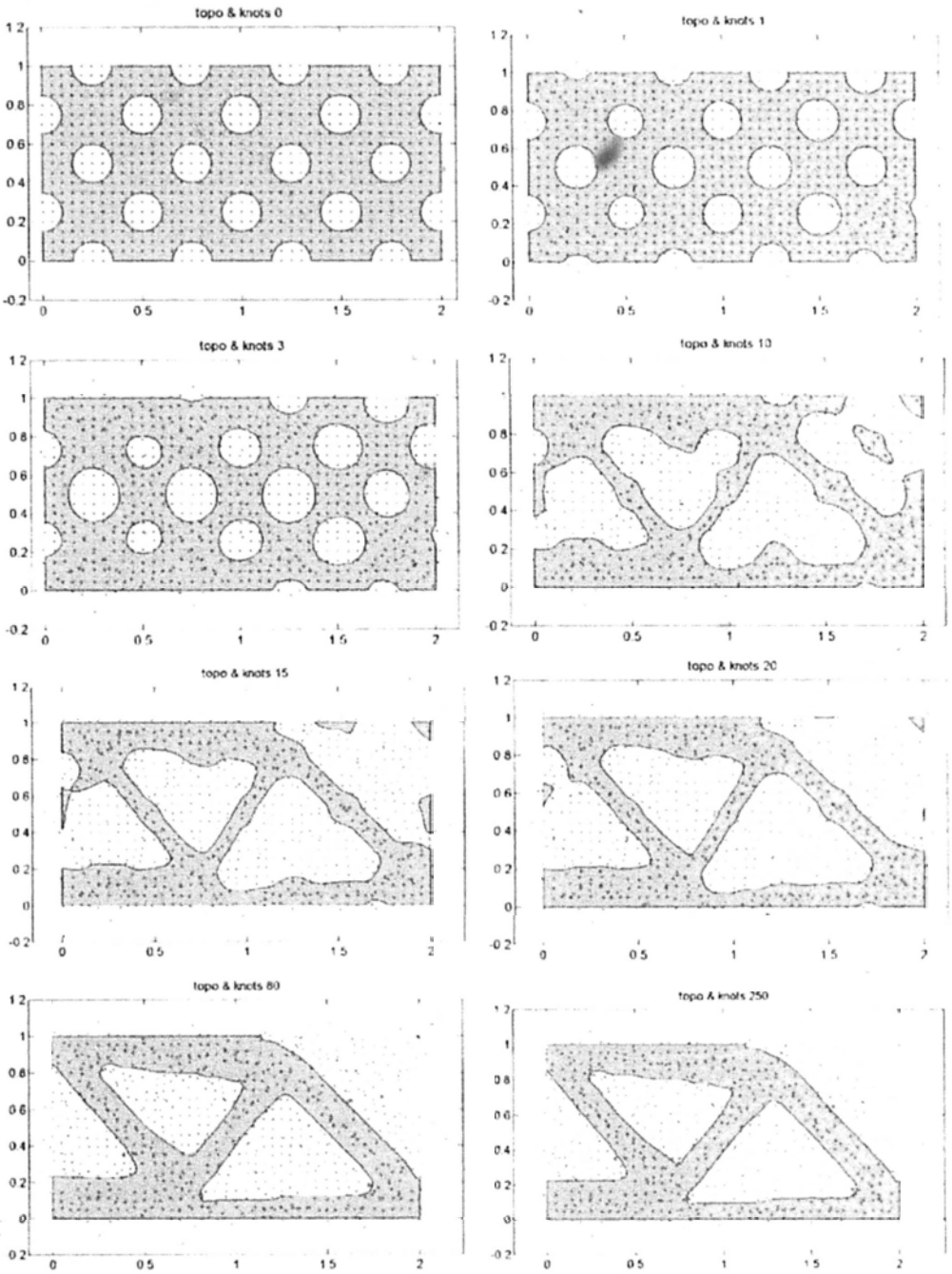


Figure 4.13: Optimization steps of CASE 3b: $41 \times 21 - 861$ knots, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, converged at Step 256, time consumed per iteration – 9.4 second.

Chapter 5

RBF-POU Method

The discretization of partial differential equations (PDEs) is usually obtained by covering the domain of interest by a suitable grid. Then, after time discretization, linearization, and space discretization (by finite elements, finite differences, or finite volumes), a linear system of discrete equations is set up and solved. Beside uniform structured grids which are most simple to handle, block structured grids and unstructured grids are also employed successfully. Furthermore, for dealing with non-smooth data or solutions, adaptive refinement is a must for efficiency reasons [2, 13, 22, 52].

However, all grid- or mesh-based methods are quite involved when it comes to time-dependent problems with complicated geometries, free moving boundaries, and interfaces. Then the geometry of the domain may change over time, the non-smooth part of the data or the non-smooth part of the solution changes with time, or the location of singularities of the solution may vary time dependently. In the Eulerian approach, an adaptive mesh technique must follow the time-dependent features of the data or solution by local refinement and coarsening of the mesh [13, 21, 52, 67]. But time-dependent adaptive mesh refine-

ment and coarsening is not simple, especially for three-dimensional (3D) problems. It is very tedious that programming is complicated, data structures are not easy to handle, and the storage overhead is significant. Besides, good local and global error estimators are necessary. Therefore, there exist only a few unstructured adaptive programs which are able to handle 3D application-oriented problems with time-dependent change of the geometry, the data, or the solution.

5.1 POU Method

5.1.1 Engineering Considerations

In real life engineering applications, a very time-consuming portion of the overall computation is the mesh generation from CAD input data. Typically, more than 70% of the overall computing time is spent by mesh generators. In result, especially within the engineering community, there is growing interest in other discretization methods which involve no mesh at all. These approaches are summarized under the term meshless or gridless methods. The main idea is to consider points only, i.e. we omit any fixed relation among the nodes such as element boundaries, and move these points in a time-dependent manner. The location of the points and the distribution of the points account for the description of the changing geometry, the change in data, and the time-dependent changing variation of the solution or its gradient.

5.1.2 Gridless Discretization

Generally speaking, there are two different types of meshless approaches – the classical particle methods [47, 48], and *gridless discretizations* based on data fitting techniques [7, 23]. They are truly Lagrangian

methods, i.e., they are based upon a time-dependent formulation or conservation law. In this content, We will only focus our discussion on the later method - gridless methods that follow a different approach. Patches or subdomains are attached to each point whose union forms an open covering of the domain. Then, local shape functions are constructed with the help of methods from data fitting. These shape functions are used in a Galerkin or collocation discretization process to set up a linear system of equations. Finally this system must be solved efficiently.

According to the data fitting method concerned, there are three approaches: (i) Shepard's method [79], which has a consistency of first order only, will be studied and used for data-fitting in the optimization process later; (ii) whereas there are also moving least squares method (MLSM) [40],which generalizes Shepard's approach implicitly to the case of higher order shape functions; (iii) and the partition of unity p-version method, which generalizes Shepard's approach explicitly to higher consistency orders.

5.1.3 Shepard's Method

The discretization is stated in terms of the points only. To obtain test and trial spaces, patches or subdomains $\omega_i \subset \mathbb{R}^d$ are attached to each point x_i whose union forms an open cover $\{\omega_i\}$ of the domain Ω, i.e. $\Omega \subset \bigcup \omega_i$. From the given weight functions W_i, local shape functions φ_i are constructed by Shepard's method. They form a partition of unity $\{\varphi_i\}$ such that

$$\sum_i \varphi_i \equiv 1 \qquad \text{on } \Omega. \tag{5.1}$$

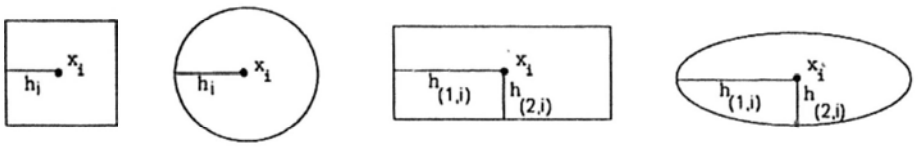


Figure 5.1: Typical examples of the shapes of a patch ω_i

Then, each shape function φ_i is multiplied with a sequence of local polynomials to gain higher degree shape functions. These are finally plugged into the weak form to set up a linear system of equations.

Now we associate a certain weight function $W_i(x)$ to each subdomain ω_i. Since we decided to use rectangular patches only, i.e. $\omega_i = \otimes_{k=1}^{d} \omega_i^{(k)}$, $\omega_i^{(k)} = \left\{ x^{(k)} \in \mathbb{R}, \left| x_i^{(k)} - x^{(k)} \right| \leqslant h_i^{(k)} \right\}$, where $h_i \in \mathbb{R}$ is the half of the diameter of the patch ω_i., some examples are shown in Figure 5.1. The most natural choice is to use a product approach of one-dimensional local functions, i.e. $W_i(x) = \otimes_{k=1}^{d} W_i^{(k)}\left(x^{(k)}\right)$, where $\operatorname{supp}(W_i) = \overline{\omega}_i$. If we use $[0,1]$ as reference interval and define the affine linear mapping $T_i^{(k)} : [0,1] \to \omega_i^{(k)}$, $x^{(k)} \to T_i^{(k)}\left(x^{(k)}\right)$, we can define the respective weight functions as $W_i^{(k)}\left(x^{(k)}\right) = W\left(T_i^{(k)}\left(x^{(k)}\right)\right)$, where the one-dimensional weight function W can be any non-negative function. Furthermore, we obtain the mapping T_i from the reference element $[0,1]^d$ to ω_i as the product $T_i : [0,1]^d \to \omega_i$, $x \to T_i(x) = \otimes_{k=1}^{d} T_i^{(k)}\left(x^{(k)}\right)$.

In the next step, we should construct shape functions φ_i from these given weight functions W_i with the help of data fitting techniques. In general, a data fitting method produces an approximation $\tilde{\phi}$ of a function ϕ by

$$\tilde{\phi}(x) = \sum_{i=1}^{N} \phi_i \cdot \varphi_i(x), \tag{5.2}$$

where N is the number of patches, ϕ_i are given data or are derived from

that. Shepard's method uses the idea of inverse distance weighting, which leads to shape functions

$$\varphi_i(x) = \frac{W_i(x)}{\sum_{j=1}^{N} W_j(x)}, \tag{5.3}$$

where $W_i(x) = \|x - x_i\|^{-\beta}$. But since such shape functions have global support, they would lead to a dense stiffness matrix and a quadratic complexity of the method. We therefore use a localized version of Shepards approach. There are basically two variants. In [40] a locally supported singular weight function such as

$$W_i(x) = L_i(x)\|x - x_i\|^{-\beta}, \qquad \text{where } L_i \in C^{\infty} \text{ and } \operatorname{supp}(L_i) = \omega_i, \tag{5.4}$$

is used. This approach generates an interpolatory partition of unity, i.e. $\varphi_i(x_j) = \delta_{ij}$. Another approach is to employ a locally supported smooth weight function. We will discuss it in more details next section.

On the other hand, the amount of overlap of the cover patches and the choice of weight functions W_i affect the smoothness of the Shepard functions φ_i. If the cover is minimal, i.e. there is exactly one patch ω_j for every $x \in \overline{\Omega}$ with $x \in \overline{\omega_j}$, the partition of unity degenerates to the characteristic functions $\varphi_i = \mathcal{X}\omega_i$ independent of the chosen weight functions W_i. Thus, we see that small overlaps will cause very large gradients of φ_i close to the boundary of the respective support ω_i.

5.2 RBF-POU Model

The key concept of POU method is that we divide the design domain into numbers of overlapping patches, or subdomains as described in Figure 5.2. Let $\{\Omega_i\}_{i=1}^{M}$ be a collection of M overlapping patches which cover Ω such that $\Omega \subset \bigcup \Omega_i$. In order to yield the POU of the cover

$\{\Omega_i\}_{i=1}^M$, we need to define a collection of the non-negative shape functions $\{\phi_i\}_{i=1}^M$ with compact support and continuous functions such that its property as shown in (5.6). For each subdomain Ω_i, we use RBF to compute a local reconstruction function Φ_i. Then, the approximation $\tilde{\Phi}$ of the global function Φ can be defined by the sum of all the local interpolants with a set of shape functions $\phi_i(x)$ as

$$\tilde{\Phi}(x,t) = \sum_{i=1}^{M} \Phi_i(x,t)\,\phi_i(x), \tag{5.5}$$

and this set of positive, compactly-supported shape functions $\phi_i(x)$ are obtained from a set of weight functions W_i by an inverse distance weighting procedure

$$\phi_i(x) = \frac{W_i(x)}{\sum_j W_j(x)}, \tag{5.6}$$

where it satisfies $\sum \phi_i(x) = 1$. There are many choices for the smooth weight functions W_i and they are often defined as the composition of a distance function $D_i : \mathbb{R}^n \to [0,1]$ and a decay function $\mathcal{V} : [0,1] \to [0,1]$ such that $W_i(x) = \mathcal{V} \circ D_i(x)$ [88]. The distance function must be smooth at the boundary of Ω_i, then $D_i(x) = 1$. For an Euclidean box in 3D defined by two opposite corners S and T, the distance function

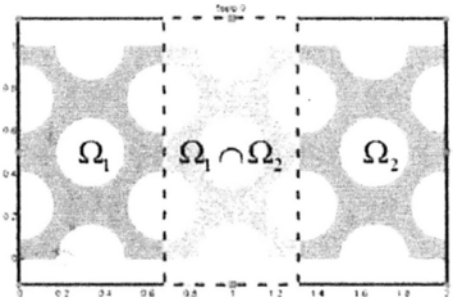


Figure 5.2: Plot of two subdomains Ω_i in the entire domain Ω under POU condition

D_i is expressed as [30, 97]

$$D_i(x) = 1 - \prod_{r \in (x,y,z)} \frac{4(x_r - S_r)(T_r - x_r)}{(T_r - S_r)^2}. \tag{5.7}$$

The decay function $\mathcal{V}$ must be continuous from the local interpolant Φ_i to the global function $\tilde{\Phi}$. The following functions are recommended for these continuity requirements:

$$\begin{aligned} C^0 &: \mathcal{V}^0(d) = 1 - d, \\ C^1 &: \mathcal{V}^1(d) = 2d^3 - 3d^2 + 1, \\ C^2 &: \mathcal{V}^2(d) = -6d^5 + 15d^4 - 10d^3 + 1. \end{aligned} \tag{5.8}$$

5.3 RBF-POU with Stationary Knots

5.3.1 Sensitivity algorithm

Now we assume that the positions of knots x_j are fixed during the optimization process and the expansion coefficients α_{ij} changes with time. For equation (5.5), the local reconstruction function Φ_i is now represented by RBF interpolant where $\varphi_{ij} = \varphi_i(\|x - x_j\|), x_j \in D$. Then, modify (3.30) as,

$$\Phi_i(x,t) = \sum_{j=1}^{N} \alpha_{ij}(t)\varphi_{ij}(x). \tag{5.9}$$

Substitute (5.9) into (5.5) for the new representation of $\tilde{\Phi}$ and the Hamilton-Jacobi equation (2.13) can be rewritten as

$$\sum_{i=1}^{M} \phi_i \sum_{j=1}^{N} \dot{\alpha}_{ij}\varphi_{ij}(x) - v_n \left|\nabla\tilde{\Phi}\right| = 0, \tag{5.10}$$

and hence the normal velocity v_n on the moving boundary can be obtained by rewriting (5.10)

$$v_n = \frac{1}{\left|\nabla\tilde{\Phi}\right|} \sum_{i=1}^{M} \phi_i \sum_{j=1}^{N} \dot{\alpha}_{ij}\varphi_{ij}(x). \tag{5.11}$$

As derived in THEOREM 3.3, we substitute (5.11) into (3.3) the shape derivative of the Lagrangian to construct the corresponding sensitivities

$$\frac{dL}{dt} = \sum_{i=1}^{M}\sum_{j=1}^{N} \dot{\alpha}_{ij} \int_{\partial\Omega} \left(\lambda - \varepsilon(u)^{\mathrm{T}} C \varepsilon(u)\right) \frac{1}{\left|\nabla\tilde{\Phi}\right|} \phi_i \varphi_{ij}(x)\, ds, \tag{5.12}$$

then by chain rule, we get

$$\frac{dL}{dt} = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{\partial L}{\partial \alpha_{ij}} \cdot \dot{\alpha}_{ij} = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{\partial J}{\partial \alpha_{ij}} \cdot \dot{\alpha}_{ij} + \lambda \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{\partial V}{\partial \alpha_{ij}} \cdot \dot{\alpha}_{ij}. \tag{5.13}$$

As shown in (5.12) and (5.13), the sensitivities of the objective function and the volume constraint can be derived by comparing them in the following way:

$$\frac{\partial J}{\partial \alpha_{ij}} = -\int_{\partial\Omega} \varepsilon(u)^{\mathrm{T}} C \varepsilon(u) \frac{1}{\left|\nabla\tilde{\Phi}\right|} \phi_i \varphi_{ij}(x) ds, \qquad i = 1, ..., N; j = 1, ..., M, \tag{5.14}$$

$$\frac{\partial V}{\partial \alpha_{ij}} = \int_{\partial\Omega} \frac{1}{\left|\nabla\tilde{\Phi}\right|} \phi_i \varphi_{ij}(x)\, ds, \qquad i = 1, ..., N; j = 1, ..., M. \tag{5.15}$$

5.3.2 Optimization algorithm

To find the local minimum of the objective function as stated in (3.1), we employ the steepest descent method to proceed with the search in the descent direction of the sensitivity functions at the current point as shown in (5.14) & (5.15).

From (5.13) the search direction can be defined as

$$\dot{\alpha}_{ij} = -\left(\frac{\partial J}{\partial \alpha_{ij}} + \lambda \frac{\partial V}{\partial \alpha_{ij}}\right), \qquad i = 1, ..., N; j = 1, ..., M, \tag{5.16}$$

and the expansion coefficient α_{ij} can be updated as

$$\alpha_{ij}^{n+1} = \alpha_{ij}^{n} + \tau \dot{\alpha}_{ij}, \qquad i = 1, ..., N; j = 1, ..., M, \tag{5.17}$$

where τ is the time step and it is fixed as a positive constant.

Once the expansion coefficient is updated as α_{ij}^{n+1} and substitute into (5.9) to get the updated local interpolant, thence the new level set function can be deduced from (5.5)

$$\tilde{\Phi}^{n+1} = \sum_{i=1}^{M} \phi_i \sum_{j=1}^{N} \alpha_{ij}^{n+1} \varphi_{ij}, \tag{5.18}$$

and then the boundary of the structure is propagated accordingly. Because the knot positions remain unchanged and the shape function is independent on time, i.e. $\phi_i(x)$ is constant throughout the process.

Algorithm 2 The proposed algorithm is an iterative method and implemented as follows:

1: Choose initial $\tilde{\Phi}_0 \in D$, set $n = 0$.

2: **repeat**

3: Compute $u^n \in \Omega^n$.

4: Compute $\varphi^n \in \Omega^n$, i.e. solve the RBF equation (4.1).

5: Solve the descent gradient $\frac{\partial J}{\partial \alpha_{ij}}$, $\frac{\partial V}{\partial \alpha_{ij}}$ by solving (5.14) & (5.15).

6: Evaluate the search direction $\dot{\alpha}_{ij}$ in (5.16).

7: Update the expansion coefficient α_{ij}^{n+1} by (5.17).

8: Update level set function $\tilde{\Phi}^{n+1}$ by (5.18).

9: Update shape Ω^{n+1}.

10: Increment n.

11: **until** $\left\| J\left(\tilde{\Phi}^{n+1}\right) - J\left(\tilde{\Phi}^{n}\right) \right\| < TOL.$

The new shape Ω^{n+1} is characterized by the level set function $\tilde{\Phi}^{n+1}$ through the time step $\triangle t_\tau$, it is chosen in the way that

$$J(\Omega^{n+1}) \leqslant J(\Omega^n). \tag{5.19}$$

5.3.3 Numerical result

CASE 4 – Optimization with stationary knot and POU

Here, we present an illustrative numerical example – CASE 4 and the test problem is the same as the one implemented in Section 4.4.1. We summarize the model settings used in this test and the computation time in Table 5.1.

Table 5.1: Comparison of results for CASE 4.

CASE No.	Knots size	POU	Time/Step (s)	Iteration	Compliance
4a	81×41	√	8.5	1000	62.4
4b	81×41	X	11.5	1000	62.4

Although we keep all the parameters same as them used in the previous cases, we must be careful that no direct comparison can be made between these two models. It is because we do not apply moving knots scheme in this case and the knots are fixed in space instead. As concluded by most of the authors [41, 93, 101] that the dynamic knots scheme is far more efficient than the method of changing expansion coefficient α_{ij} only.

For the simplicity of this feasibility study, we divide the entire domain into two subdomains only in CASE 4a, however, it is easy and straight forward to extend it into multiple partitions for the finer calculation. In addition, the test problem is optimized with a stationary

knots scheme labeled as CASE 4b for the comparison purpose, as it can be seen in Figure 5.11.

In Figure 5.10, one can notice that the shape does not have significant change after the first 100 steps. The outline of the optimized shape can only be obtained after 500 steps, especially for the elimination of the small 'island' in the middle of the left hand side of the beam. This result coincide with the step history of the method of the changing coefficient α_{ij}. We note that POU method reduces the computational time per iteration by more than 25%.

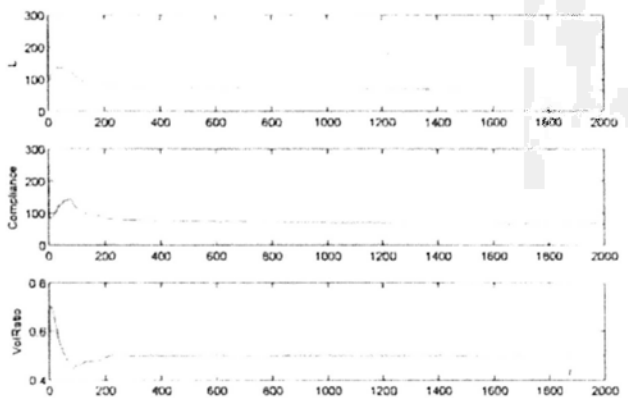


Figure 5.3: Convergence history of CASE 4a

As shown in Figure 5.3, the values of all the objective indexes such as Lagrangian, volume ratio and mean compliance are gradually converged to the steady level in between 200 – 250 steps. It is exactly the performance characteristic of structural optimization by the method of changing coefficient α. It proves that neither POU method improve nor degrade the quality of the computational result, in fact, POU methods only utilizes the computational resources during the progress of the optimization efficiently.

5.4 RBF-POU with Dynamic Knots

As shown in Section 5.3, we conclude that POU method will not carry any significant loss in the representation accuracy onto the shape, which is constructed with the parametric method of RBFs. Unlike (3.35) for the original global RBF method, the sensitivity computation is now local and much more efficient, for a modest N_i in each local region Ω_i Therefore, we can transform the general shape and topology optimization problem into a parameter optimization problem in the subdomains, enabling us to enhance computational efficiency without any significant loss in representation accuracy.

In our aforementioned two research tasks, the dynamic knots scheme is aimed to utilize each basis function to its maximum extend in the level set function representation. Thus, the benefit is to reduce the number of basis functions and hence the computational cost. The POU RBF scheme restrains the span of a set of basis functions within their local region and consequently saves efforts in the sensitivity integration computations. Naturally, if we combine these two schemes, we would have the best use of the two ideas.

Our final research task is to devise a combined method to allow knot positions to be design variables for the local RBFs within the subdomains of the POU decomposition of Ω. Therefore, the design sensitivity (4.9) would be generalized for the POU method to the variation of knot x_{ij} in the subdomain Ω_i and the updating schemes of (4.12) for the knot positions can also be utilized in the similar fashion.

It is expected that the investigation would yield a concrete technique that combines the dynamic knots method with the POU scheme to achieve a high level of computational efficiency with the capability to reasonably handle the computational complexity concerning more than

1,000,000 radial basis functions. At this level of capability, we should be able to apply the RBF-based parametric level set method for shape and topology optimization of a large class of three-dimensional structures and mechanisms and to make the method practically viable for industrial applications in aerospace, structural, and mechanical engineering. Therefore, we will test the 'dynamic knots RBF-POU' extensively with different configuration settings in this section and benchmark the results with the previous findings to validate our approach.

5.4.1 Sensitivity algorithm

In the following we assume the expansion coefficient α_{ij} as a constant throughout the optimization process, however, the positions of the knots x_j in each patch are now changed with time and (4.1) can be modified as

$$\varphi_{ij}(x,t) = \frac{1}{\sqrt{(x - x_j(t))^2 + c_j^2}}, \tag{5.20}$$

then, the local reconstruction function Φ_i is,

$$\Phi_i(x,t) = \sum_{j=1}^{N} \varphi_{ij}(x,t)\,\alpha_{ij}. \tag{5.21}$$

Substitute (5.21) into (5.5) for the new representation of $\tilde{\Phi}$ and the Hamilton-Jacobi equation (2.13) can be rewritten as

$$\sum_{i=1}^{M} \phi_i \sum_{j=1}^{N} \alpha_{ij} \frac{\partial \varphi_{ij}(x,t)}{\partial x_{ij}} \cdot \dot{x}_{ij} - v_n \left|\nabla \tilde{\Phi}\right| = 0, \tag{5.22}$$

where

$$\nabla \tilde{\Phi} = \frac{\partial \tilde{\Phi}}{\partial x_{ij}} = \sum_{i}^{M} \phi_i \sum_{j}^{N} \alpha_{ij} \frac{\partial \varphi_{ij}}{\partial x_{ij}} + \frac{1}{\sum_j W_j} \sum_i \frac{\partial W_i}{\partial x_i} \left(\Phi_i - \tilde{\Phi}\right) \tag{5.23}$$

and hence the normal velocity v_n on the moving boundary can be obtained by rewriting (5.22)

$$v_n = \frac{1}{\left|\nabla\tilde{\Phi}\right|}\sum_{i=1}^{M}\phi_i\sum_{j=1}^{N}\alpha_{ij}\frac{\partial\varphi_{ij}(x,t)}{\partial x_{ij}}\cdot\dot{x}_{ij}. \tag{5.24}$$

Then, we substitute (5.24) into (3.3) to obtain the shape derivative in term of x_{ij}

$$\frac{dL}{dt} = \sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_{ij}\int_{\partial\Omega}\left(\lambda-\varepsilon(u)^{\mathrm{T}}\mathrm{C}\varepsilon(u)\right)\frac{1}{\left|\nabla\tilde{\Phi}\right|}\phi_i\frac{\partial\varphi_{ij}(x,t)}{\partial x_{ij}}\,ds\cdot\dot{x}_{ij}\ , \tag{5.25}$$

then by chain rule, we get

$$\frac{dL}{dt} = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\partial L}{\partial x_{ij}}\cdot\dot{x}_{ij} = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\partial J}{\partial x_{ij}}\cdot\dot{x}_{ij} + \lambda\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\partial V}{\partial x_{ij}}\cdot\dot{x}_{ij}\ . \tag{5.26}$$

As shown in (5.25) and (5.26), the sensitivities of the objective function and the volume constraint can be derived by comparing them in the following way:

$$\frac{\partial J}{\partial x_{ij}} = -\alpha_{ij}\int_{\partial\Omega}\varepsilon(u)^{\mathrm{T}}\mathrm{C}\varepsilon(u)\frac{1}{\left|\nabla\tilde{\Phi}\right|}\phi_i\frac{\partial\varphi_{ij}}{\partial x_{ij}}\,ds, \quad i=1,...,N; j=1,...,M, \tag{5.27}$$

$$\frac{\partial V}{\partial x_{ij}} = \alpha_{ij}\int_{\partial\Omega}\frac{1}{\left|\nabla\tilde{\Phi}\right|}\phi_i\frac{\partial\varphi_{ij}}{\partial x_{ij}}\,ds, \quad i=1,...,N; j=1,...,M. \tag{5.28}$$

5.4.2 Optimization algorithm

To find the local minimum of the objective function as stated in (3.1), we employ the steepest descent method to proceed with the search in the descent direction of the sensitivity functions at the current point as shown in (5.27) & (5.28).

From (5.26) the search direction can be defined as

$$\dot{x}_{ij} = -\left(\frac{\partial J}{\partial x_{ij}} + \lambda \frac{\partial V}{\partial x_{ij}}\right), \qquad i = 1, ..., N; j = 1, ..., M, \tag{5.29}$$

and the position of knots x_{ij} can be updated as

$$x_{ij}^{n+1} = x_{ij}^{n} + \tau \dot{x}_{ij}, \qquad i = 1, ..., N; j = 1, ..., M, \tag{5.30}$$

where τ is the time step and it is fixed as a positive constant.

Once the knot position is updated as x_{ij}^{n+1} and substitute into (5.21) to get the updated local interpolant, thence the new level set function can be deduced from (5.5)

$$\tilde{\Phi}^{n+1} = \sum_{i=1}^{M} \phi_i \sum_{j=1}^{N} \alpha_{ij} \varphi_{ij}^{n+1}, \tag{5.31}$$

thus, the implicit function is computed everywhere in D and then the boundary of the structure is propagated accordingly.

Algorithm 3 The proposed algorithm is an iterative method and implemented as follows:

1: Choose initial $\tilde{\Phi}_0 \in D$, set $n = 0$.

2: **repeat**

3: Compute $u^n \in \Omega^n$.

4: Compute $\varphi^n \in \Omega^n$, i.e. solve the RBF equation (4.3).

5: Solve the descent gradient $\frac{\partial J}{\partial x_{ij}}$, $\frac{\partial V}{\partial x_{ij}}$ by solving (5.27) & (5.28).

6: Evaluate the search direction $\dot{x}_{ij}$ in (5.29).

7: Update the position of knots x_{ij}^{n+1} by (5.30).

8: Update the RBF value φ^n as in **Step 4**.

9: Update level set function $\tilde{\Phi}^{n+1}$ by (5.31).

10: Update shape Ω^{n+1}.

11: Increment n.

12: **until** $\left\| J\left(\tilde{\Phi}^{n+1}\right) - J\left(\tilde{\Phi}^{n}\right) \right\| < TOL.$

The new shape Ω^{n+1} is characterized by the level set function $\tilde{\Phi}^{n+1}$ through the time step $\triangle t_\tau$, it is chosen in the way that

$$J(\Omega^{n+1}) \leqslant J(\Omega^{n}). \tag{5.32}$$

5.4.3 Numerical result

Now we have successfully enabled the movement of the knot, so that the first study should be given to the effect of the number of the overlapping knots. It is because the size of the overlapping regions of the adjacent patches may impose different effects on the Shepard functions ϕ_i and may cause very large gradient of ϕ_i close to the boundary of the respective support Ω_i. In result, the accuracy of the global approximation $\tilde{\Phi}$ may eventually be jeopardized by the insufficient representation.

CASE 5 – Effect of the amount of overlap of the cover patches

We present here an illustrative numerical example – CASE 5 and the test problem is the same as the one implemented in Section 4.4.1. In this test we study three overlapping configuration settings and they are: (a) 1-knot, (b) 3-knots & (c) 5-knots in both directions of the rectangular patch i.e. n_x and n_y. We summarize the model settings and the computation time in the following Table 5.2 and the initial patch configuration of CASE 5b is shown in Figure 5.4.

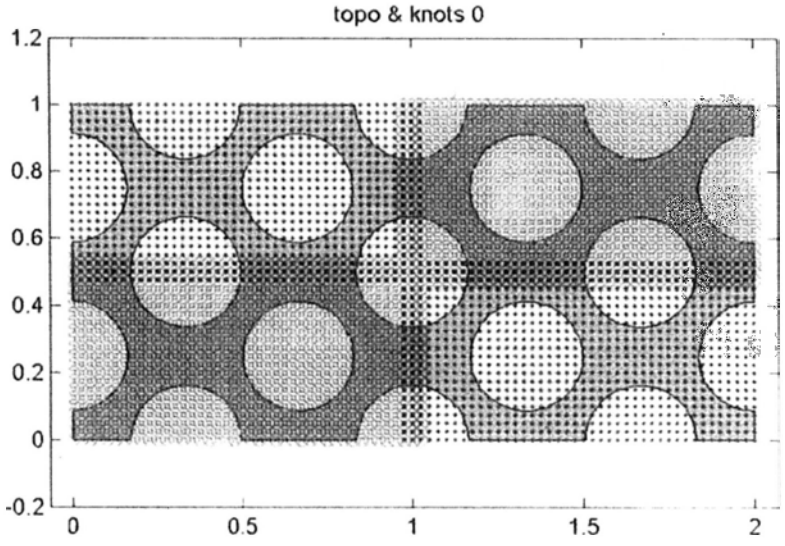


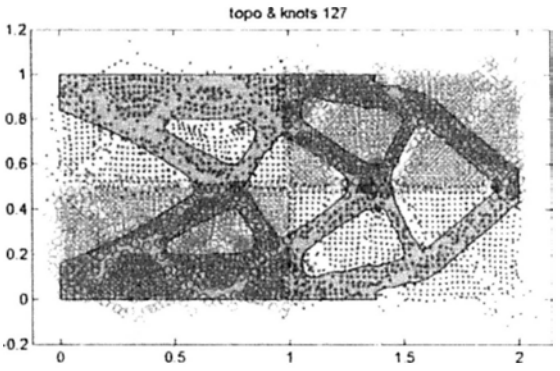
Figure 5.4: Patch pattern of CASE 5b

It is straight forward to understand that the computational time is directly related to the amount of the overlap cover. As shown in Table 5.2, CASE 5a scores the shortest computational time, for which there is only one knot situated in the overlapping region. In Figure 5.5, we capture the iteration step '127' and compare their topologies and find out the results are very similar in 3 cases.

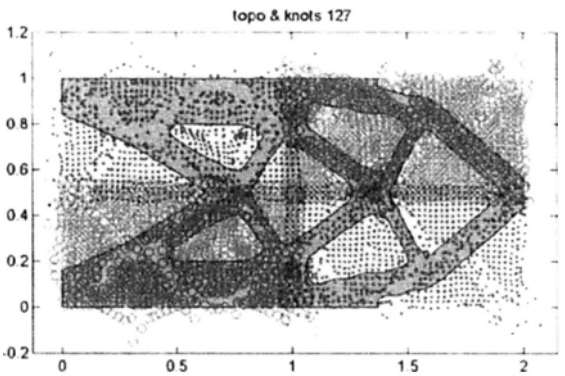
In fact, the rate of the topology change in 3 cases are almost the same and hence the amount of the overlap cover does not apply any noticeable effect on the simulation result in each step, but it does decide

CASE No.	Overlap size $n_x \times n_y$	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance
5a	41 × 21	2 × 2	3444	6.83	60.6
5b	42 × 22	2 × 2	3696	7.12	60.4
5c	43 × 23	2 × 2	3956	7.32	61.7

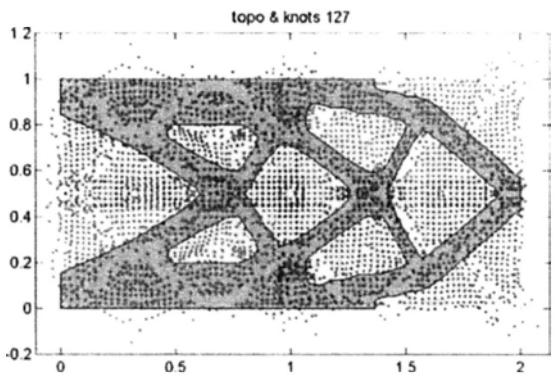
Table 5.2: Comparison of results for CASE 5.



(a) CASE 5a



(b) CASE 5b



(c) CASE 5c

Figure 5.5: Optimization step 127 of CASE 5

the speed of the calculation. As the total number of knot increases due to a bigger overlap cover, the original coefficient matrix with the dimensions $A = N \times N$ will enlarge in the rate of $(N + n_x) \times (N + n_y)$.

The results evidence that the overlap cover with one knot from the neighboring patch is sufficient, the boundaries of Ω_i can be well represented by the Shepard function ϕ_i and thus the accuracy of the global approximation $\tilde{\Phi}$ is maintained in a good condition. Consequently, we will only apply 1-knot overlap cover in the following experiments for higher efficiency.

CASE 6 – Effect of patch pattern

Finally, we need to find out the effect of the patch pattern ($P_y \times P_x$) to the simulation results and its overall efficiency, where P_x and P_y are the number of patch in x- and y-direction, respectively. Theoretically, the efficiency will be raised as the number of patch increases, however, as indicated in test CASE 5 that the total number of knots will also increase. It exists a contradiction between the number of patches and knots towards the performance of the optimzation.

In CASE 6a, we divide the domain into patches in the x-direction only and the model settings are shown in Table 5.3. Then, the domain will be cut into patches in y-direction only in CASE 6b as listed in Table 5.4. Follow that we discretize the whole domain in x- and y-direction simultaneously in CASE 6c and the settings are tubulated in the Table 5.5. In each case there are five model settings to be tested and all the simulations will be stopped at 500 steps unless it converges in the earlier stage. The overlap size is 1-knot in n_x and n_y in all cases. The volume fraction target is 0.5.

As can be seen in Table 5.4 and 5.5, the computation speed increases as the number of patch increases. Even though the number of knot increases with the increment of the patch, the efficiency does not drop but rises proportionally.

CASE No.	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance	Volume Fraction
i	1×2	3362	8.05	60.4	0.499
ii	1×4	3444	6.96	60.5	0.499
iii	1×5	3485	6.69	61.0	0.498
iv	1×8	3608	6.41	61.1	0.499
v	1×10	3690	6.29	60.8	0.499

Table 5.3: Comparison of results for CASE 6a.

CASE No.	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance	Volume Fraction
i	2×1	3402	7.95	60.5	0.499
ii	4×1	3564	6.87	60.8	0.499
iii	5×1	3645	6.69	60.5	0.498
iv	8×1	3888	6.49	60.8	0.499
v	10×1	4050	6.35	60.7	0.499

Table 5.4: Comparison of results for CASE 6b.

In order to visualize the effect induced by the numbers of the patches and knots, we plot the calculation time of the "sensitivity analysis" against the number of patch for CASE 6a & CASE 6b in Figure 5.6. Because the overall calculation consists of three parts: 1. FEM calculation, 2. sensitivity analysis & 3. updating processes. The processes (1) and (2) are only affected by the number of geometrical node which is kept constant throughout all cases. As the times of these two processes are fluctuated from step to step, it is more fairly to compare their computational costs consumed by the sensitively analysis.

CASE No.	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance	Volume Fraction
i	2×2	3444	6.86	60.4	0.499
ii	4×4	3696	6.07	60.5	0.499
iii	5×5	3825	5.98	61.0	0.498
iv	8×8	4224	5.93	61.1	0.499
v	10×10	4500	5.93	60.8	0.499

Table 5.5: Comparison of results for CASE 6c.

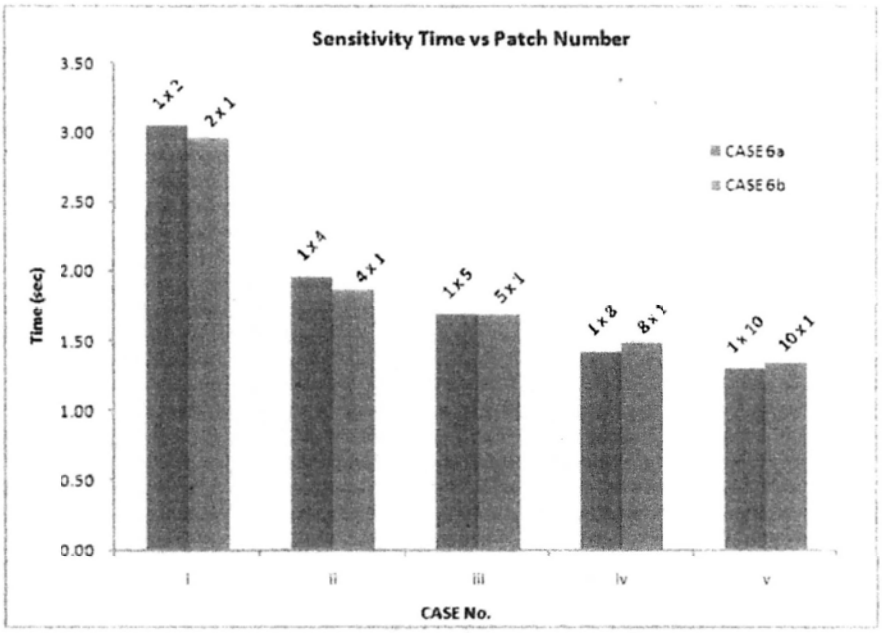


Figure 5.6: Plot of Sensitivity Time vs Patch Number for CASE 6a & 6b

As shown in Figure 5.6, CASE 6a and CASE 6b have similar patch divisions but in the opposite directions. The differences between these two cases in terms of number of knots and sensitivity time are less than 2%. The result shows that the computation time is mainly governing the number of patch as long as the difference in the knot amount is not significant large.

CASE No.	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance	Volume Fraction
i	5 × 10	4050	5.92	60.6	0.500
ii	10 × 5	4250	5.94	60.6	0.500
iii	10 × 20	5000	5.99	61.4	0.500
iv	20 × 10	5400	6.01	61.2	0.500
v	20 × 20	6000	6.14	62.3	0.500
vi	20 × 40	7200	6.50	94.3	0.500

Table 5.6: Comparison of results for CASE 6d.

In Figure 5.12 and 5.13, we show the simulation steps of CASE 6a(iv) [1 × 8] and CASE 6b(iv) [8 × 1] in details, respectively. The results ensure that the POU method does not impose significant quality problem onto the accuracy of the calculation. We can see in these figures that the boundary propagations in both cases are very close and the final topologies are approximately the same.

In order to further study the efficiency change against the number of patch and knots, we perform the last CASE 6d to accommodate more patch combinations and total knot numbers. The details of the model settings are listed in Table 5.6. In CASE 6d(i) [5 × 10], it finishes an iteration in 5.92 seconds which is the quickest in this study, and the total number of patch and knot are 50 and 4050, respectively. As shown in the Figure 5.14, the final result converges to the similar topology obtained from the typical dynamic knots scheme as well.

The results of CASE 6c (iv) & (v) are comparable to the quickest one, however, they use higher number of knots to represent the level set functions. The overviews of the comparison results are given in the following Figure 5.7 (number of patch vs sensitivity time) and Figure

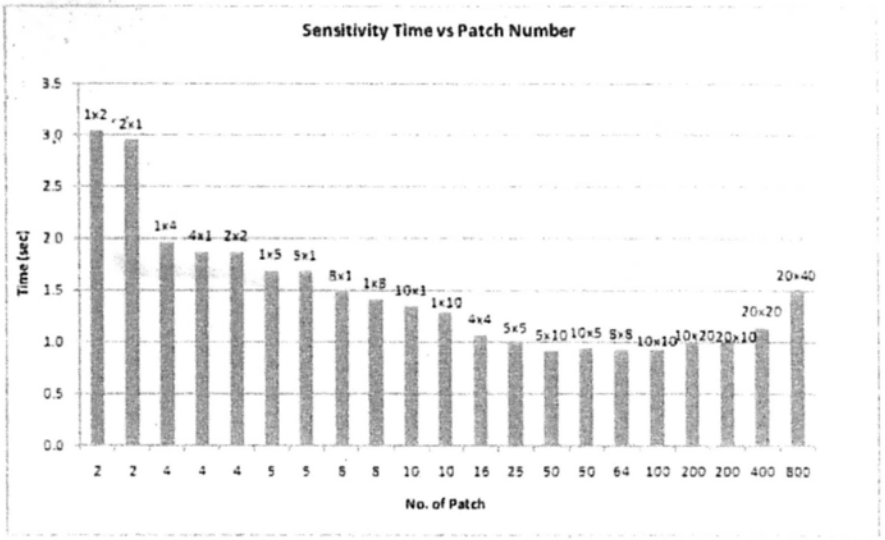


Figure 5.7: Plot of Sensitivity Time vs Patch Number

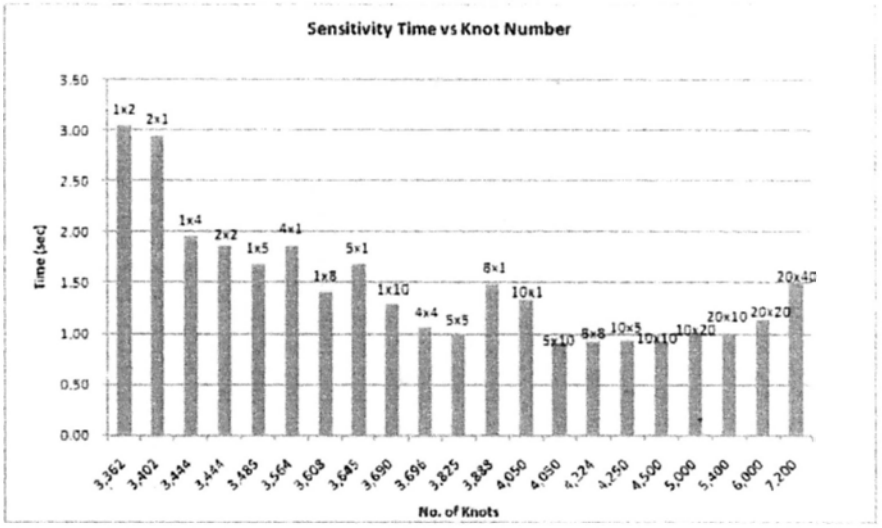


Figure 5.8: Plot of Sensitivity Time vs Knot Number

5.8 (number of knot vs sensitivity time). Then, we observe from the results that as the patch number goes over 100, the sensitivity time increases even though the patch number is also increasing. It is because the knot number increases significant and causes the knot density too high that actually slow down the calculation. In CASE 6d(vi) the numbers of patch and the knot are the highest, however, the over-

populated knot density deteriorates the simulation quality as we have already commented in the previous chapter. As a result, the final topology is not satisfied as shown in Figure 5.16.

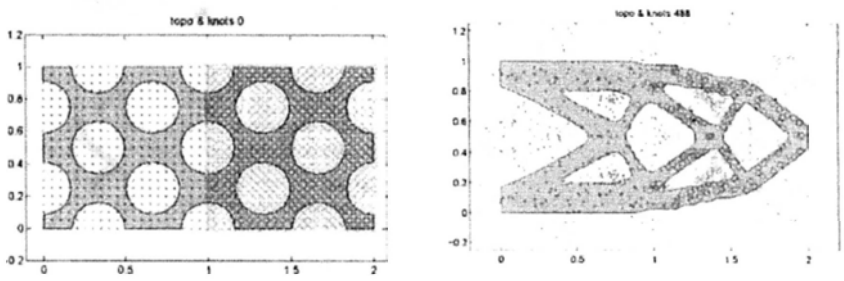
5.5 Summary of POU with Dynamic Knots

As demonstrated in the numerous and intensive experiments, we conclude that the POU method can reduce the computation time without jeopardizing the quality of the simulation, as long as the number of patch and knot in a reasonable ratio. In this study, we find out that there should be one division of patch cover Ω_i for every 8 knots, i.e. the diameter of the patch $2h_i$, such that the best result happened to CASE 6d(i) [5×10] accordingly.

On the other hand, since the combination of the continuous RBFs implicit function φ_i and the compactly supported shape function ϕ_i provides sufficient representation over the whole domain so that it does not need to place large amount of knot in the overlap cover. Otherwise, the excessive number of knots in the overlap cover slows down the overall performance as shown in CASE 5. It is because the original coefficient matrix with the dimensions $A = N \times N$ will enlarge in the rate of $(N + n_x) \times (N + n_y)$, as happened in CASE 6d(vi).

In this study, we realize that POU method is stable against different arrangements of the patch pattern and the computational efficiency will be very close as long as the total number of patch is same. This effect has been conducted in CASE 6 in detail. As can be seen in Figure 5.7, the cases with the same number of patch use similar sensitivity time in an iteration and the maximum discrepancy is less than 5%.

Finally, we combine the techniques of POU method and dynamic knot to minimize the total number of knot, e.g. 861 knots in CASE 2c.



(a) Step = 0 (b) Step = 488

Figure 5.9: Optimization result of CASE 7: Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 1 \times 2$, total knots number = 882, time consumed per iteration = 5.5 seconds.

Now, we shall have utilize each basis functions φ_i to its maximum capability in representing the implicit level set function Φ. POU method confines a smaller set of basis functions within the local cover ω_i and hence saves computational cost in the sensitivity integration process.

CASE 7 – RBF-POU with optimal knot distribution

Patch Dim (knot)	Patch Pattern	Total No. Knots	Time/Step (sec)	Compliance	Volume Fraction
21×21	1×2	882	5.5	60.1	0.499

Table 5.7: Comparison of results for CASE 7.

In the last experiment – CASE 7, we examine the ultimate performance of the combined method with the test case as of Section 4.4.1 – short cantilever beam. The model settings are in the following fashion: $(P_y \times P_x = 1 \times 2)$, total number of knot = 882 & 1-knot in the overlap cover, i.e. each patch with a dimensions of 21×21. The result is listed in Table 5.7; the initial and final topologies are shown in Figure 5.9(a) & Figure 5.9(b), respectively. The simulation is converged at step 488. The sensitivity time in this case is 0.5 second and it is comparable

to the fastest example in previous study – CASE 6d(i), whereas its sensitivity time is 0.92 second.

□ **End of chapter.**

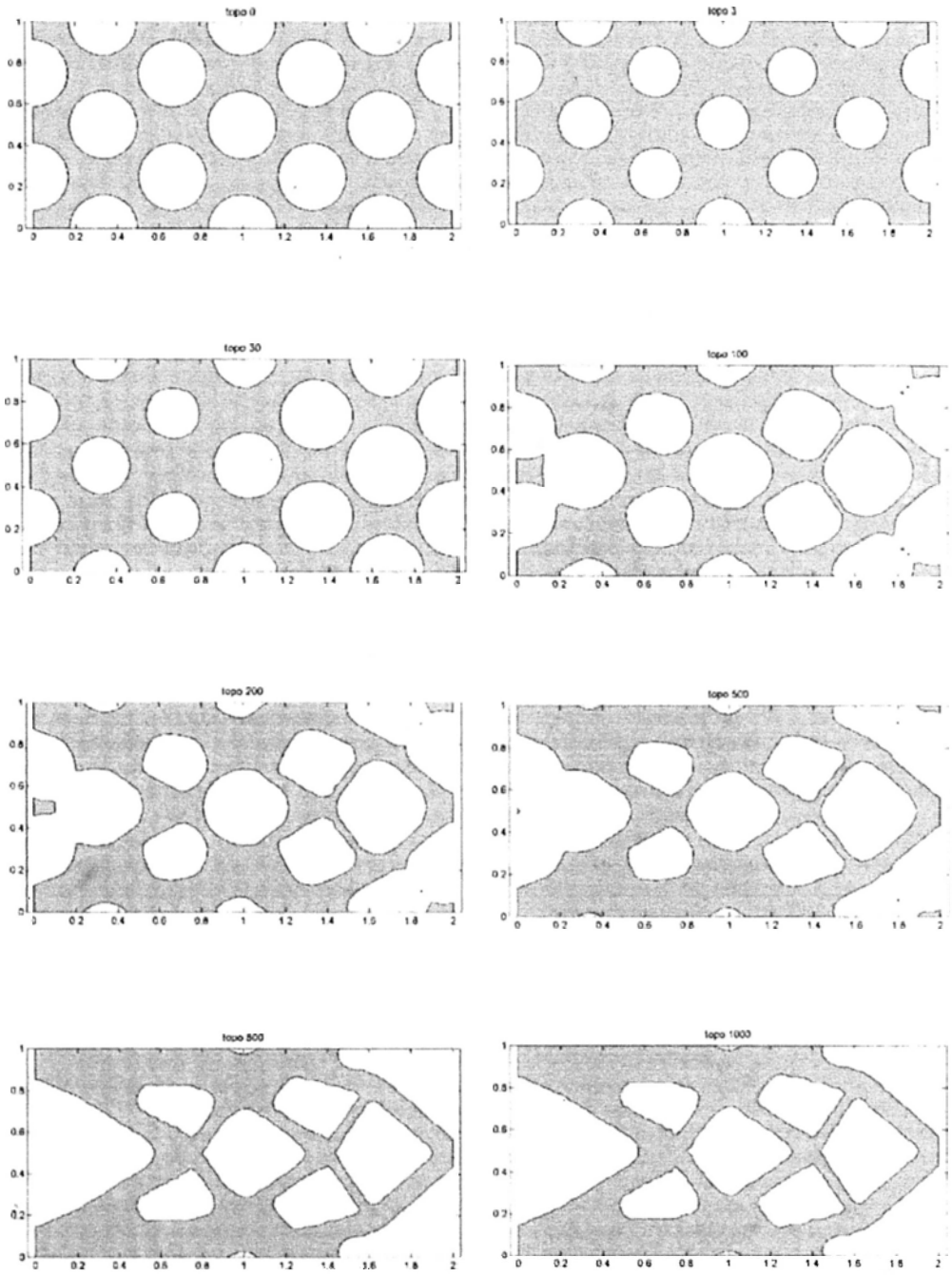


Figure 5.10: Optimization steps of CASE 4a: Changing α_{ij} scheme with POU, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 1000 steps, time consumed per iteration = 8.5 seconds.

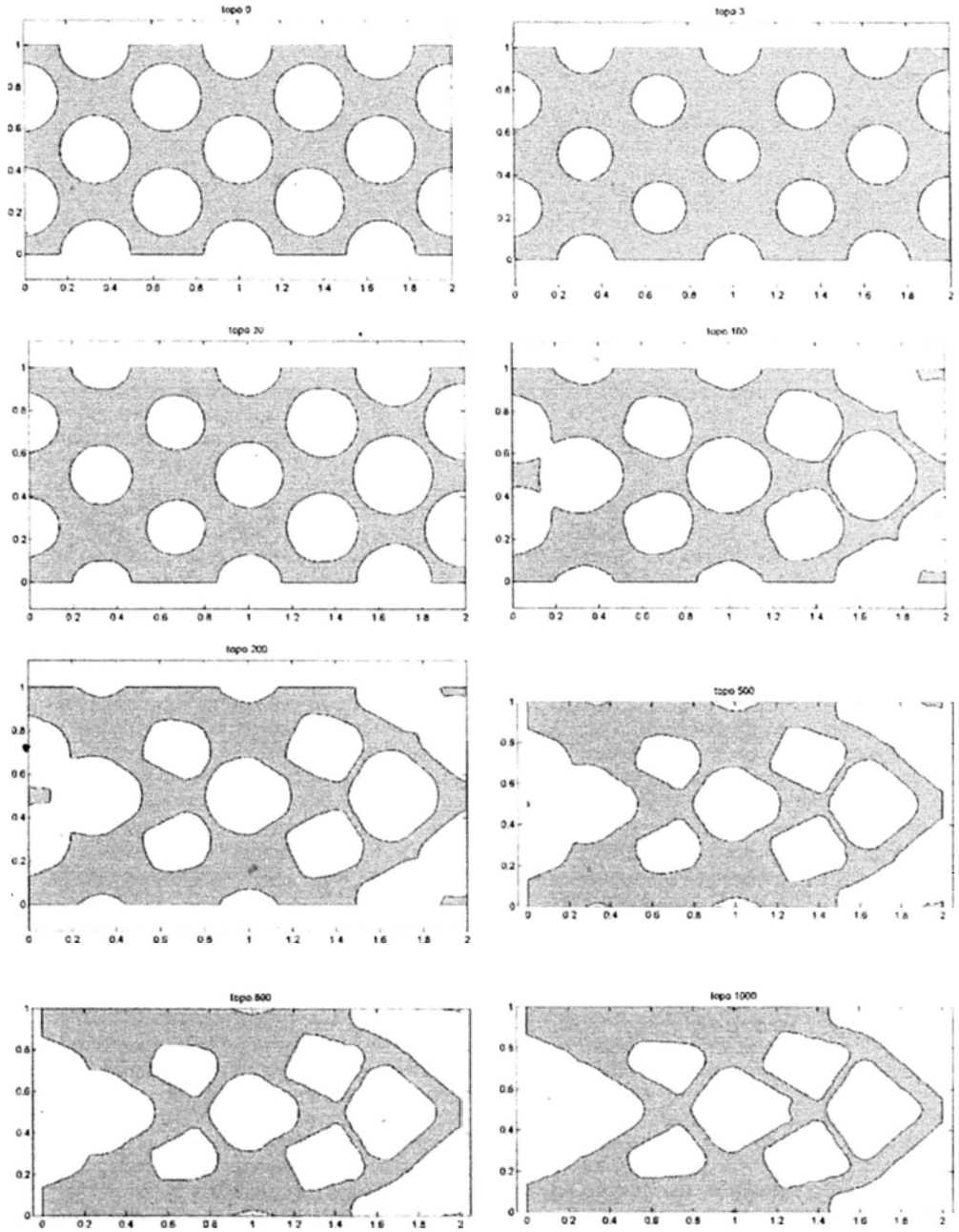


Figure 5.11: Optimization steps of CASE 4b: Changing α_{ij} scheme without POU, $t_\tau = 5 \times 10^{-3}$ and $c_i = 0.025$, no convergence after 1000 steps, time consumed per iteration = 11.5 seconds.

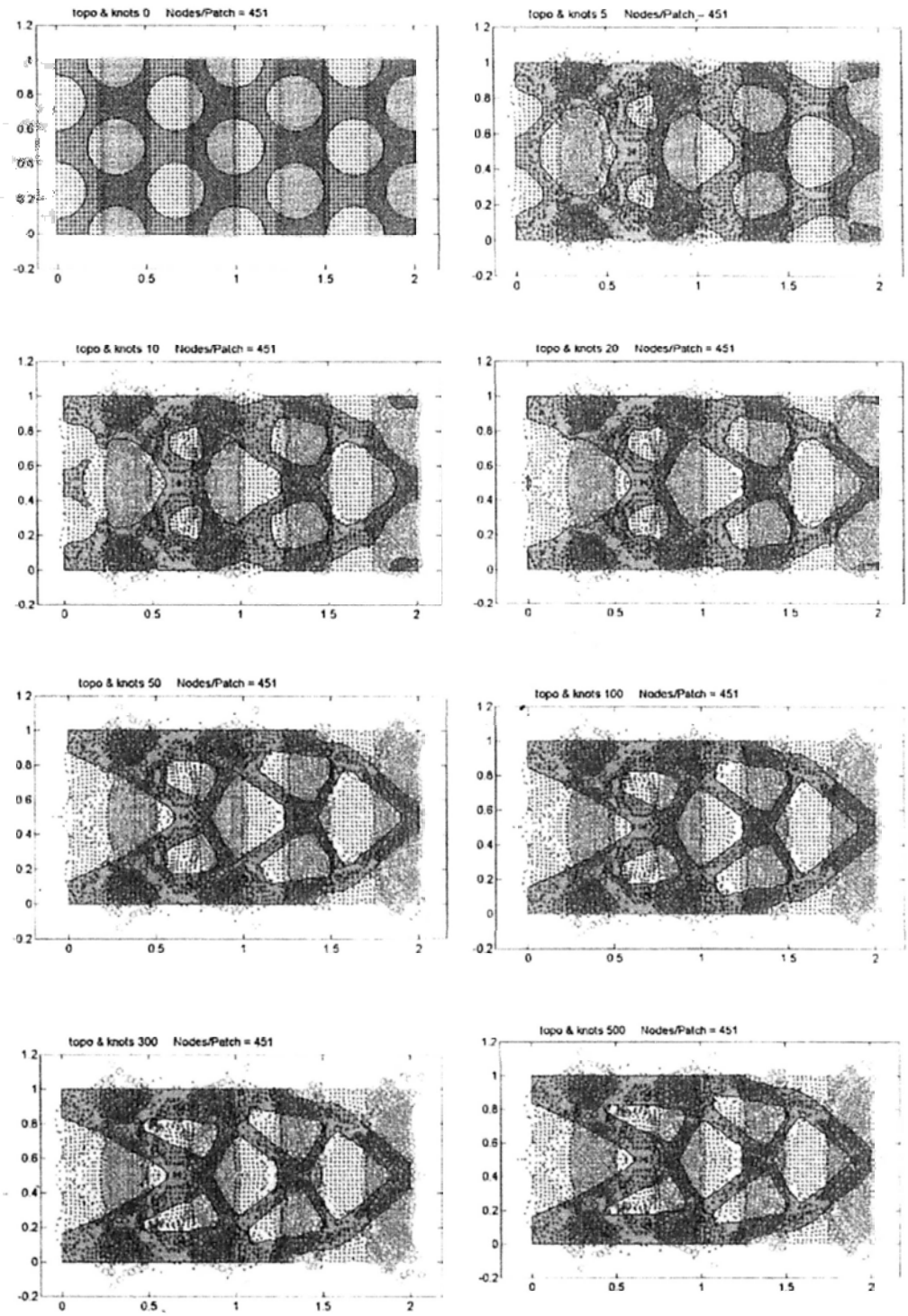


Figure 5.12: Optimization steps of CASE 6a(iv): Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 1 \times 8$, total knots number = 3608, time consumed per iteration = 6.41 seconds.

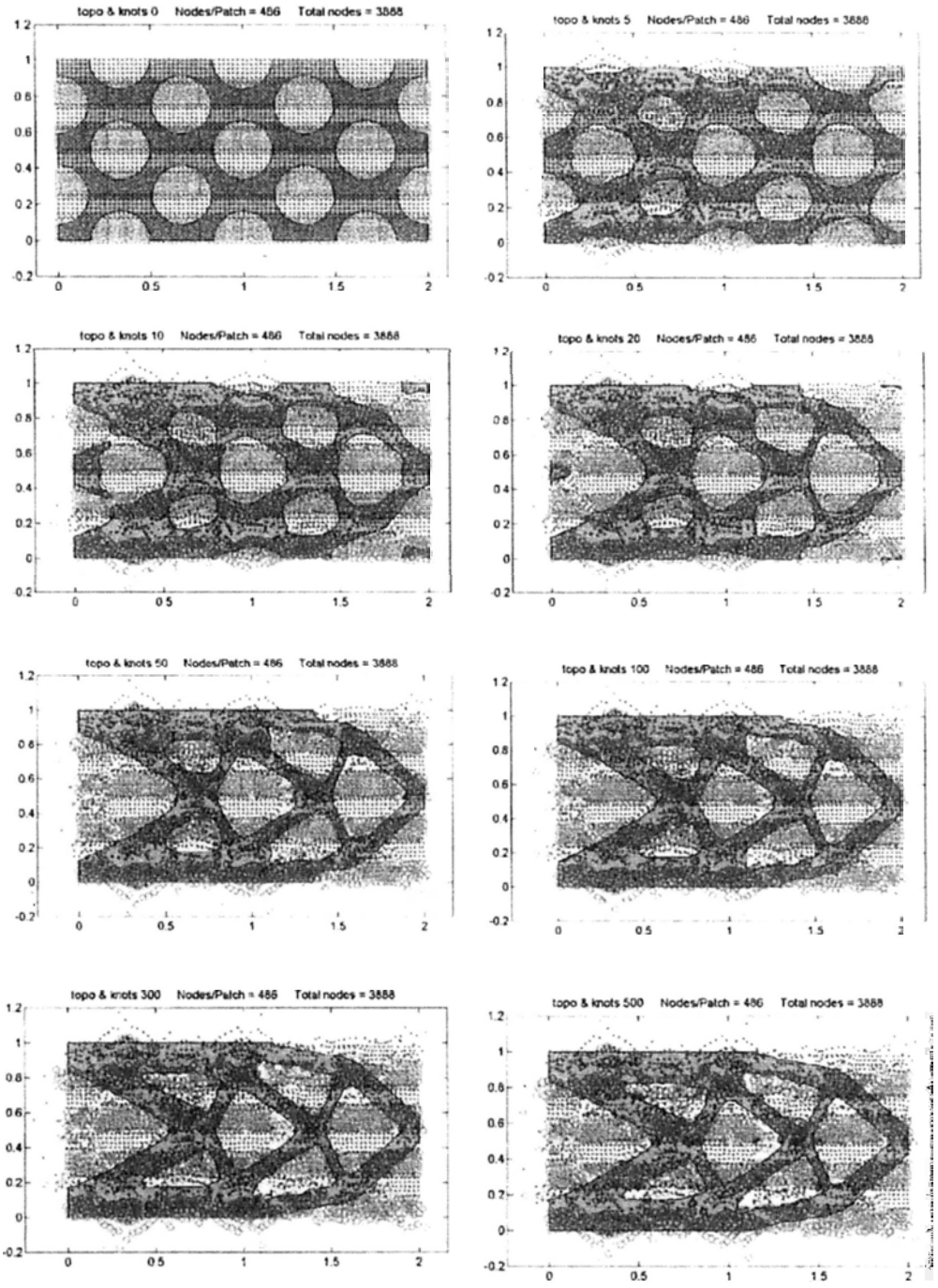


Figure 5.13: Optimization steps of CASE 6b(iv): Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 8 \times 1$, total knots number = 3888, time consumed per iteration = 6.49 seconds.

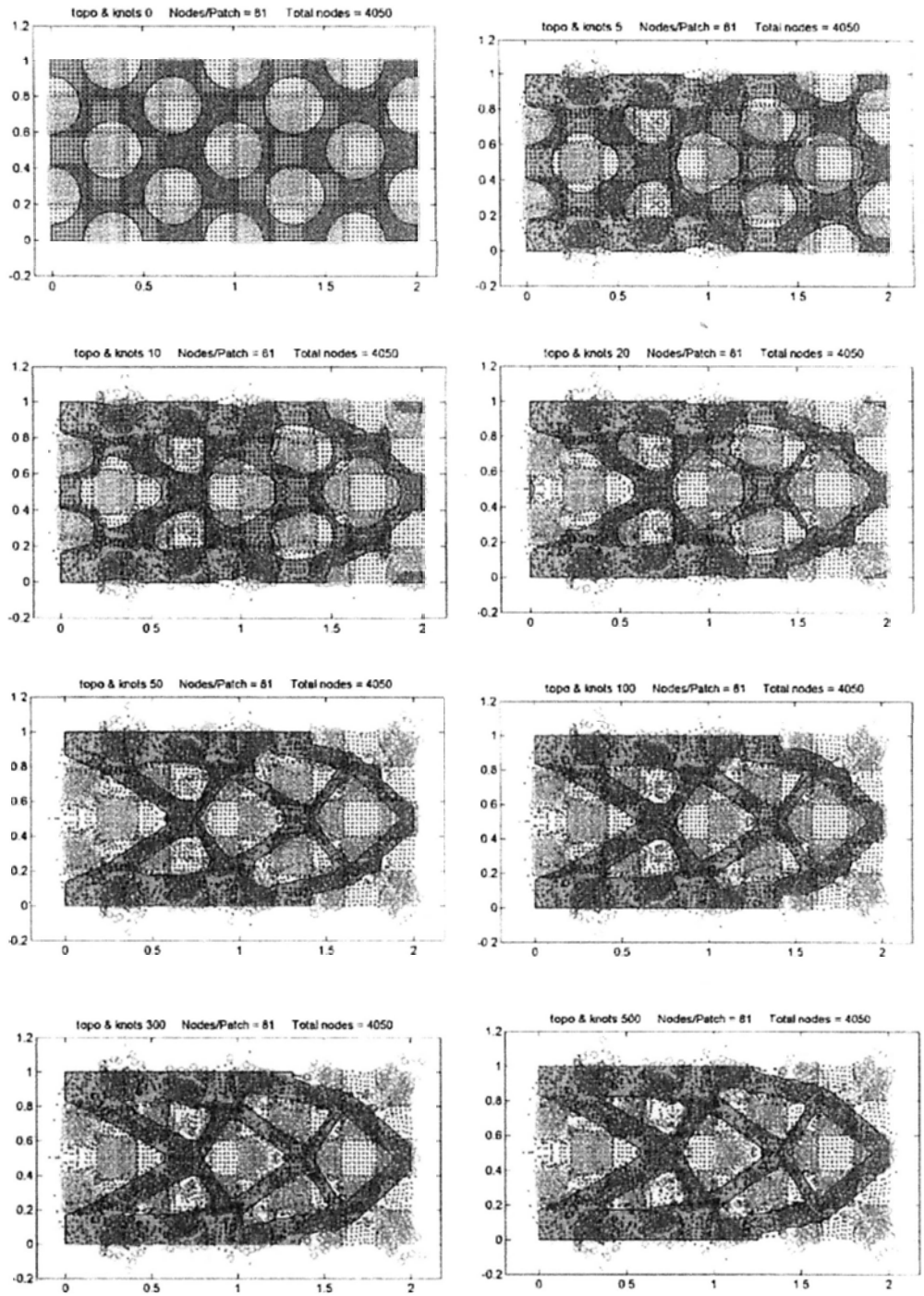


Figure 5.14: Optimization steps of CASE 6d(i): Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 5 \times 10$, total knots number = 4050, time consumed per iteration = 5.92 seconds.

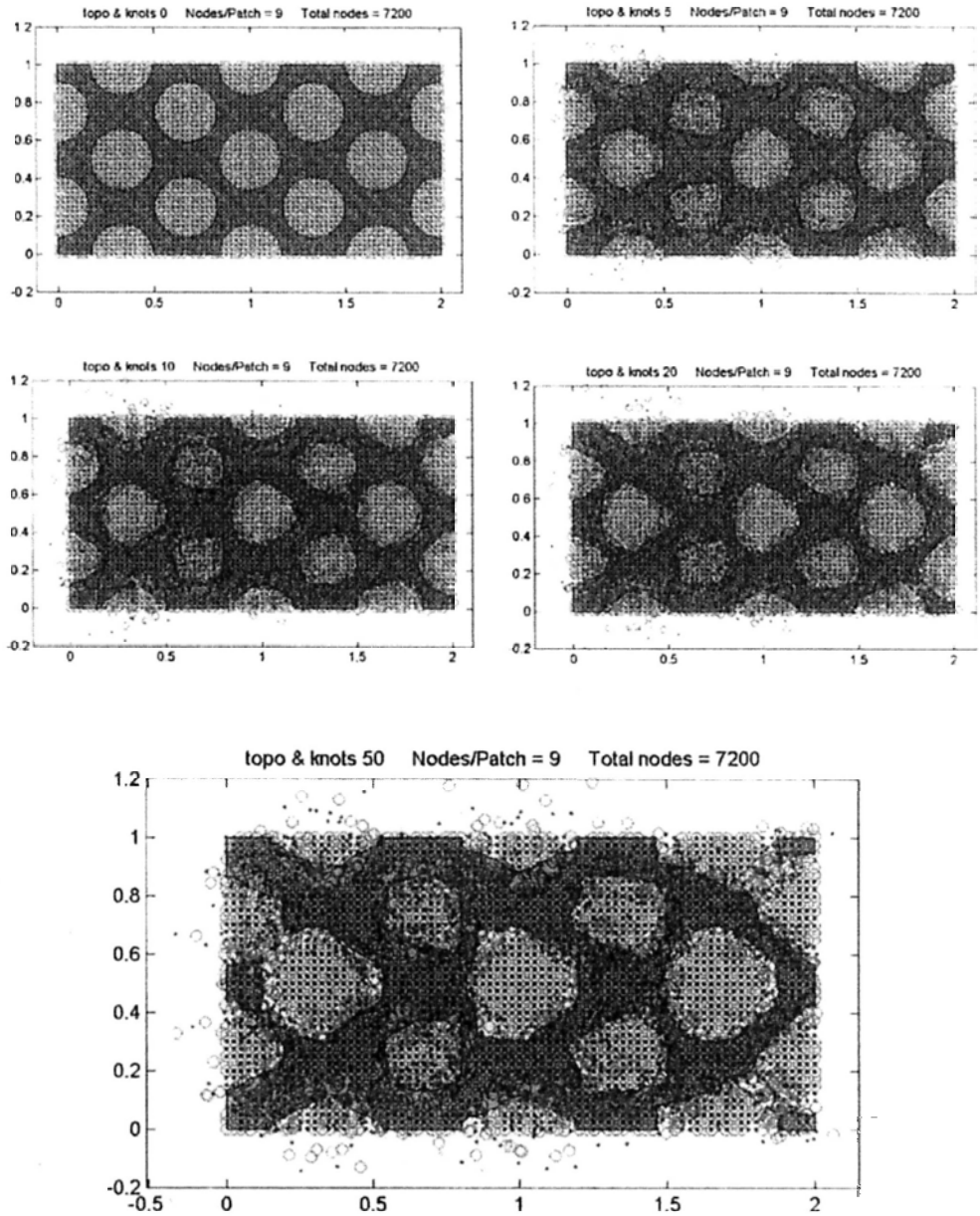


Figure 5.15: Optimization steps of CASE 6d(vi): Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 20 \times 40$, total knots number = 7200, time consumed per iteration = 6.50 seconds.

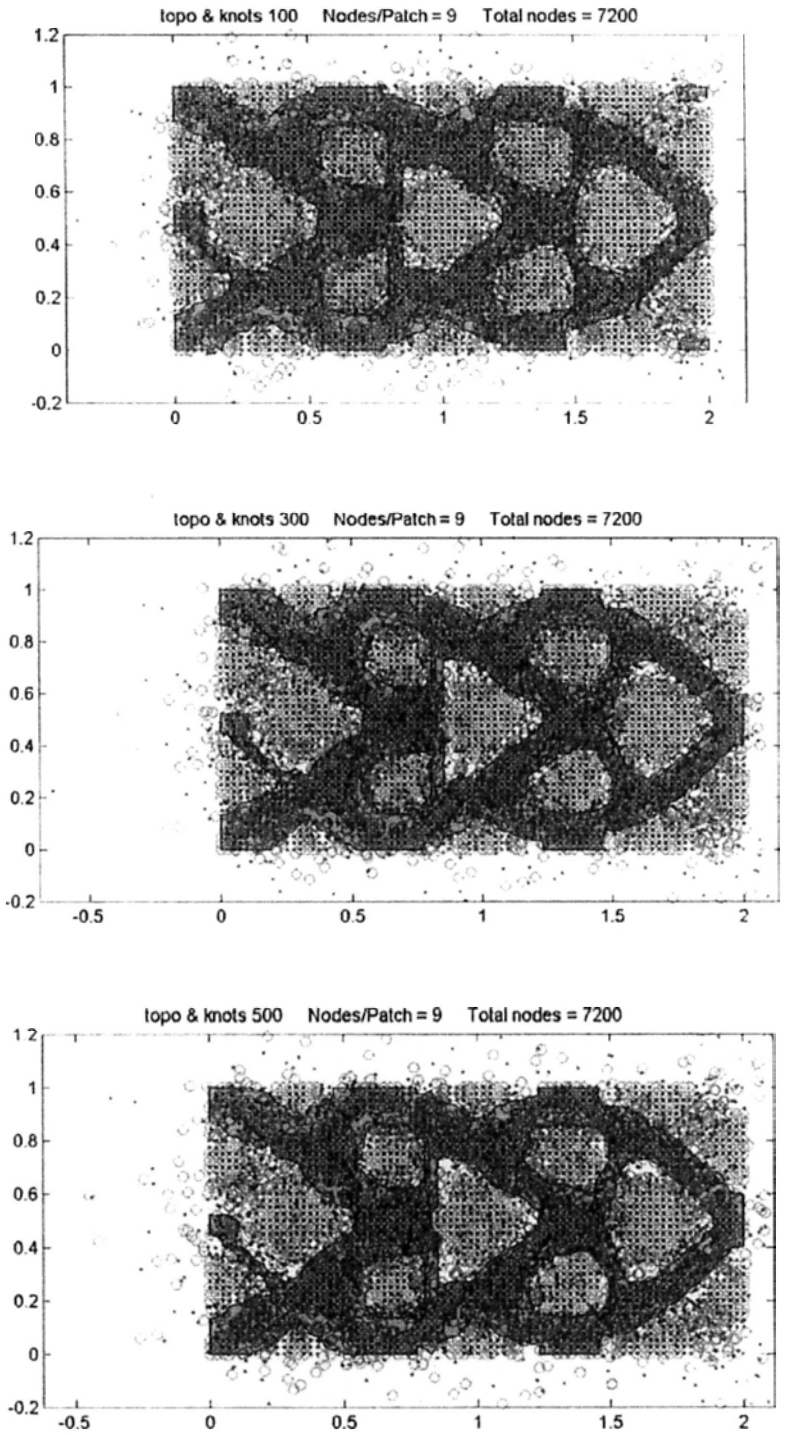


Figure 5.16: Optimization steps of CASE 6d(vi) (cont'): Changing x_{ij} scheme with POU, patch pattern $(P_y \times P_x) = 20 \times 40$, total knots number = 7200, time consumed per iteration = 6.50 seconds.

Chapter 6

Conclusion

6.1 Conclusion

In this study, we have successfully transformed the conventional discrete topology optimization problem into its parametric form by the use of the radial basis function. The flexibility of the parametrization not just provide an easier implementation of the problem, but also put forward the shape derivative into an useful shape sensitivity analysis, which allow the user to compute the problem with more sophisticated mathematical programming methods that offer higher efficiency and accuracy. On the other hand, the parameterized algorithms can be easily extended to solve for the optimal solution in multi-physical requirements such as minimum stress and eigenvalues.

In order to further improve the computational efficiency, we implement a gridless discretization method to divide the domain Ω into smaller open cover Ω_i for more efficient sensitivity integration. POU is a time-dependent Lagrangian method and the parametrization of level set method makes the implementation simple and straight forward. Then, we use the data fitting method – Shepard's method, which has

a consistency of first order, to approximate the exact solution Φ well locally with the shape functions such that $\tilde{\Phi}(x,t) = \sum_i^M \Phi_i(x,t)\phi_i(x)$.

The theoretical formulations, the calculation algorithms as well as the computational results have been implemented and presented in this thesis in full detail. Three major schemes of RBF parameterizations are studied here:

A. RBFLS with dynamic knots algorithm.

B. RBFLS-POU with stationary knots algorithm.

C. RBFLS-POU with dynamic knots algorithm.

The numerical experiments show that *Scheme A* performs with an overwhelming advantages in both the quantitative and qualitative aspects as compared to the original moving knots method. Definitely, the dynamic knots method consumes lesser computational power and offers the users with the converged answers in a shorter timeframe. However, the nature of the globally-supported RBF such as IMQs increase the computational complexity as the number of knots go over 100,000. It is rather difficult to keep the number below this level when we are dealing with the models in 3D. *Scheme B* is our next approach for tackling this complexity problem, as its standard numerical algorithms are efficient for recursive domain decomposition over multi-levels. The numerical results show that the computational efficiency is improved while without any significant loss in representation accuracy.

Finally, we have successfully coupled above mentioned schemes into a *combined method* as implemented in 'Scheme C' so that we use the knot positions as the design variables for the local interpolant Φ_i within the subdomains of the POU decomposition of the domain Ω. Consequently, we can deduce the generalized shape sensitivities for POU

method as $\frac{\partial J}{\partial x_i}$ and $\frac{\partial V}{\partial x_i}$ for the optimization. From the experiments of CASE 5 & 6, we show that the approximation quality of the global function $\tilde{\Phi}$ has not been deteriorated even though the overlapping knots move out of the overlap covers. It is an encouraging results that we may use lesser knots to complete a simulation with a quality of being as fidelity as the work done in CASE 7.

6.2 Future Work

In our next research, we will extend this numerical schemes to the 3D models and the realtime engineering problems. The effects and relations among the knot positions, the number of knots, patch configurations and the representation quality will be intensively studied and numerically observed.

In addition, we will focus on multi-material microstructure designs of multi-physics. These areas are considered as the significant application of the structural topology optimization. They have found their ways in designing materials with unusual properties such as negative Poisson's ratios and zero thermal expansion coefficients [12]. It would be informative to compare the results of our approach with those obtained with the homogeneous methods and discrete level set method. With the benchmarking results, we shall demonstrate with outstanding numerical efficiency, flexibility in handling topological changes, fidelity of boundary representation, and high degree of automation of the proposed the parametric RBF-POU based level set method for 3D optimization problems.

□ **End of chapter.**

Bibliography

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

[2] J. C. Aguilar and J. B. Goodman. Anisotropic mesh refinement for finite element methods based on error reduction. *J. Comput. Appl. Math.*, 193(2):497–515, 2006.

[3] G. Allaire, F. D. Gournay, F. Jouve, and A.-M. Toader. Structural optimization using topological and shape sensitivity via a level set method. Internal report 555, Ecole Polytechnique, France, 2004.

[4] G. Allaire, F. Jouve, and A. M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, February 2004.

[5] S. Amstutz. Sensitivity analysis with respect to a local perturbation of the material property. *Asymptotic Analysis*, 49(1-2):87–108, September 2006.

[6] S. Amstutz and H. Andrä. A new algorithm for topology optimization using a level-set method. *Journal of Computer Physics*, 216(2):573–588, 2006.

[7] T. Belytschko, Y. Krongauz, D. Organ, and M. Fleming. Meshless methods: An overview and recent developments. In *Computer Methods in Applied Mechanics and Engineering*, pages 3–47, 1996.

[8] T. Belytschko, S. Xiao, and C. Parimi. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering*, 57(2):1177–1196, 2003.

[9] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural and Multidisciplinary Optimization*, 1(4):193–202, December 1989.

[10] M. P. Bendsøe. Optimization of structural topology. In *Shape and Material*. Springer, Berlin, 1995.

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

[12] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Advances in design and control. Springer Verlag, Berlin Heidelberg, 2nd edition, 2003.

[13] M. J. Berger and J. E. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. Technical report, Stanford, CA, USA, 1983.

[14] D. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[15] M. D. Buhmann. Radial basis functions: Theory and implementations. In *Cambridge Monographs on Applied and Compu-*

tational Mathematics, volume 12. Cambridge University Press, New York, 2004.

[16] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1):344–362, February 2004.

[17] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *ACM SIGGRAPH 2001*, pages 67–76, Los Angeles, CA, 2001.

[18] T. Cecil, J. L. Qian, and S. Osher. Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions. *Journal of Computational Physics*, 196(1):327–347, 2004.

[19] T. F. Chan and X.-C. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *J. Comput. Phys.*, 193(1):40–66, 2004.

[20] A. D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito. Exponential convergence and *h*-c multiquadric collection method for partial differential equations. *Numerical Methods for Partial Differential Equations*, 19(5):571–594, April 2003.

[21] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):79 – 112, 1989.

[22] P. Deuflhard, P. Leinen, and H. Yserentant. Concepts of an adaptive hierarchical finite element code. *IMPACT Comput. Sci. Eng.*, 1(1):3–35, 1989.

[23] C. A. Duarte and J. T. Oden. A new meshless method to solve boundary-value problems, 1995-2005.

[24] H. A. Eschenauer, H. A. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural and Multidisciplinary Optimization*, 8:42–51, 1994.

[25] G. E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In A. L. Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface fitting and multiresolution methods*, pages 131–138. Vanderbilt University Press, Nashville, TN, 1st edition, 1997. International Conference on Curves and Surfaces (3rd : 1996 : Chamonix-Mont-Blanc, France).

[26] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93(1):73–82, 1998.

[27] R. Franke. Scattered data interpolation: Tests of some method. *Mathematics of Computation*, 38(157):181–200, 1982.

[28] P. FulmańSki, A. Laurain, J. F. Scheid, and J. SokołOwski. A level set method in shape and topology optimization for variational inequalities. *International Journal of Applied Mathematics and Computer Science*, 17(3):413–430, October 2007.

[29] S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for pde systems: The elasticity case. *SIAM J. Control Optim.*, 39(6):1756–1778, 2000.

[30] M. Griebel and M. A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic

PDEs. *SIAM Journal on Scientific Computing*, 22(3):853–890, August 2000.

[31] M. Griebel and M. A. Schweitzer. A particle-partition of unity method – part ii: Efficient cover construction and reliable integration. *SIAM Journal on Scientific Computing*, 23(5):1655–1682, January 2002.

[32] M. Griebel and M. A. Schweitzer. A particle-partition of unity method – part iii: A multilevel solver. *SIAM Journal on Scientific Computing*, 24(2):377–409, August 2002.

[33] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation (Advances in Design and Control)*. Advances in design and control. SIAM, Philadelphia, 2003.

[34] M. Hassine, S. Jan, and M. Masmoudi. From differential calculus to 0-1 topological optimization. *SIAM J. Control Optim.*, 45(6):1965–1987, 2007.

[35] L. He, C.-Y. Kao, and S. Osher. Incorporating topological derivatives into shape derivatives based level set methods. *Journal of Computational Physics*, 225(1):891–909, 2007.

[36] E. Kansa. Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics–i surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19(8-9):127 – 145, 1990.

[37] E. J. Kansaa, H. Powerb, G. E. Fasshauerc, and L. Ling. A volumetric integral radial basis function method for time-dependent

partial differential equations. I. Formulation. *Engineering Analysis with Boundary Elements*, 28:1191–1206, 2004.

[38] N. H. Kim and Y. Chang. Eulerian shape designsensitivity analysis and optimization with a fixed grid. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3291 – 3314, 2005. Structural and Design Optimization.

[39] E. Kreyszig. *Advanced Engineering Mathematics*. Wiley, 9 edition, November 2005.

[40] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[41] B. F. Y. Lui, M. Y. Wang, and Q. Xia. Parametric shape and topology optimization via radial basis functions, partition of unity and level set method. In *Proceedings of 5th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, June 2007.

[42] S. Lukaszyk. A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33(4):299–304, March 2004.

[43] N. K. Madsen. The method of lines for the numerical solution of partial differential equations. In *Proceedings of the SIGNUM meeting on Software for partial differential equations*, pages 5–7, New York, NY, USA, 1975. ACM.

[44] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, December 1986.

[45] I. M. Mitchell. A toolbox of level set methods. Technical Report TR-2004-09, University of British Columbia, Canada, 2004.

[46] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, pages 89–98, Washington, DC, USA, 2001. IEEE Computer Society.

[47] K. Nanbu. Direct simulation scheme derived from the boltzmann equation. i. monocomponent gases. *Journal of the Physical Society of Japan*, 49(5):2042–2049, 1980.

[48] H. Neunzert and J. Struckmeier. Particle methods for the boltzmann equation. *Acta Numerica*, 4(-1):417–457, 1995.

[49] G. M. Nielson. Scattered data modeling. *Computer Graphics and Applications, IEEE*, 13(1):60–70, 1993.

[50] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, August 2000.

[51] A. A. Novotny, R. A. Feijó, E. Taroco, and C.Padra. Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 192(7-8):803 –829, 2003.

[52] J. T. Oden, T. Strouboulis, and P. Devloo. Adaptive finite element methods for the analysis of inviscid compressible flow: Part i. fast refinement/unrefinement and moving mesh methods for unstructured meshes. *Comput. Methods Appl. Mech. Eng.*, 59(3):327–362, 1986.

[53] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Proceedings of ACM SIGGRAPH 2003 SESSION: Surfaces*, pages 463–470, New York, July 2003. ACM.

[54] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. *Graph. Models*, 67(3):150–165, 2005.

[55] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.

[56] S. Osher, B. Merriman, D. Peng, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[57] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: next term algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988.

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

[59] O. Pironneau. Optimal shape design for elliptic systems. In H. Cabannes, M. Holt, H. Keller, J. Killeen, and S. Orszag, editors, *System Modeling and Optimization*, volume 38 of *Springer Series in Computational Physics*, pages 42–66. Springer-Verlag, Heidelberg, 1982. Invited Plenary Lectures.

[60] H. Pitsch and L. D. de Lageneste. Large-eddy simulation of premixed turbulent combustion using a level-set approach. *Proceedings of the Combustion Institute*, 29(2):2001 – 2008, 2002.

[61] E. Quak and M. S. Floater, editors. *Tutorials on Multiresolution in Geometric Modelling: Summer School Lectures Notes.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[62] J. Reddy. *Applied Functional Analysis and Variational Methods in Engineering.* McGraw-Hill Book Company, 1986.

[63] D. F. Richards, M. O. Bloomfield, S. Sen, and T. S. Calea. Extension velocities for level set based surface profile evolution. *Journal of Vacuum Science and Technology A*, 19(4):1630–1635, 2001.

[64] G. I. N. Rozvany. Exact analytical solutions for some popular benchmark problems in topology optimization. *Structural and Multidisciplinary Optimization*, 15(1):42–48, February 1998.

[65] G. I. N. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21(2):90–108, 2001.

[66] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural and Multidisciplinary Optimization*, 4(3–4):250–252, September 1992.

[67] R. Sandboge. Adaptive finite element methods for systems of reaction-diffusion equations. pages 227–235, 1998.

[68] S. A. Sarra. Digital total variation filtering as postprocessing for radial basis function approximation methods. *Comput. Math. Appl.*, 52(6-7):1119–1130, 2006.

[69] S. A. Sarra. Integrated multiquadric radial basis function approximation methods. *Comput. Math. Appl.*, 51(8):1283–1296, 2006.

[70] R. Schaback. Creating surfaces form scattered data using radial basis functions. In M. Daehlen, T. Lyche, and L. Schumaker, editors, *Mathematical Methods in CAGD III*, pages 477–496. Vanderbilt University Press, 1995.

[71] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, 1995.

[72] R. Schaback. Improved error bounds for scattered data interpolation by radial basis functions. *Math. Comput.*, 68(225):201–216, 1999.

[73] R. Schaback and H. Wendland. Using compactly supported radial basis functions to solve partial differential equations, 1999.

[74] R. Schaback and H. Wendland. Characterization and construction of radial basis functions. In *Multivariate Approximation and Applications*, pages 1–24. Cambridge University Press, 2001.

[75] R. Schaback and H. Wendland. Inverse and saturation theorems for radial basis function interpolation. *Math. Comput.*, 71(238):669–681, 2002.

[76] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 2nd ed. edition, 1999.

[77] J. A. Sethian and J. Strain. Crystal growth and dendritic solidification. *J. Comput. Phys.*, 98(2):231–253, 1992.

[78] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163(2):489–528, September 2000.

[79] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *ACM '68: Proceedings of the 1968 23rd ACM national conference*, pages 517–524, New York, NY, USA, 1968. ACM.

[80] O. Sigmund. A 99 line topology optimization code written in MATLAB. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.

[81] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-independencies and local minima. *Structural Optimization*, 16:68–75, 1998.

[82] J. Sokolowski and A. Zochowski. On the topological derivative in shape optimization. *SIAM J. Control Optim.*, 37(4):1251–1272, 1999.

[83] J. Sokolowski and J. P. Zolésio. Introduction to shape optimization. *Journal of Computational Mathematics*, 16, 1992. Springer Verlag.

[84] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20(4):1165–1191, 1999.

[85] S. Ta'asan. Introduction to shape design and control. Online Lectures, 1997.

[86] A. Takezawa, S. Nishiwaki, and M. Kitamura. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 229(7):2697–2718, 2010.

[87] I. Tobor, P. Reuter, and C. Schlick. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. *Journal of WSCG*, 12(1–3), February 2004.

[88] I. Tobor, P. Reuter, and C. Schlick. Multi-scale reconstruction of implicit surfaces with attributes from large unorganized point sets. In *Shape Modeling Applications, 2004. Proceedings*, pages 19–30. IEEE, June 2004.

[89] M. Y. Wang and S. Y. Wang. A moving superimposed finite element method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(11):1892–1922, 2006.

[90] 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.

[91] M. Y. Wang, X. M. Wang, and D. M. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1–2):227–246, January 2003.

[92] S. Wang, K. Tai, and M. Wang. An enhanced genetic algorithm for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(1):18–44, 2006.

[93] S. Y. Wang and M. Y. Wang. Radial basis functions and level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(12):2060–2090, 2006.

[94] P. Wei and M. Y.Wang. Parametric structural shape and topology optimization method with radial basis functions and level-set method. In *Proceedings of IDETC/CIE 2006*, pages 19–30. ASME, September 2006.

[95] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, December 1995.

[96] H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *Journal of Approximation Theory*, 93(2):258–272, May 1998.

[97] X. Wu, M. Y. Wang, and Q. Xia. Implicit fitting and smoothing using radial basis functions with partition of unity. In *Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics CAD-CG*, pages 139–148, Washington DC, 2005. IEEE Computer Society.

[98] Z. Wu. Compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4(1):283–292, December 1995.

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

[100] Y. M. Xie and G. P. Steven. *Evolutionary Structural Optimization.* Springer Verlag, New York, January 1997.

[101] X. H. Xing, M. Y. Wang, and B. F. Y. Lui. Parametric shape and topology optimization with moving knots radial basis functions and level set methods. In *Proceedings of the 7th WCSMO 2007*, pages 1928–1936, 2007.

[102] J. C. Ye, Y. Bresler, and P. Moulin. A self-referencing level-set method for image reconstruction from sparse fourier samples. *Int. J. Comput. Vision*, 50(3):253–270, 2002.

[103] H. K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

Thesis/Assessment Committee

Professor Hui, Kin-chuen (Chair)

Professor Wang, Yu Michael (Thesis Supervisor)

Professor Wang, Changling Charlie (Committee Member)

Professor Tang, Kai (External Examiner)