ABSTRACT

CHEN, GUANYU. Accurate Gradient Computation for Elliptic Interface Problems with Discontinuous and Variable Coefficients. (Under the direction of Dr. Zhilin Li.)

A new numerical method is proposed for interface problems with piecewise variable coefficients. The main motivation is to get not only a second order accurate solution but also a second order accurate gradient for some types of interface problems. The idea is based on the fast IIM (Immersed Interface Method) developed for interface problems with piecewise constant coefficients, in which, second order convergence of the solution and the gradient is achieved. The key of the new method is to introduce the jump in the normal derivative of the solution as the augmented variable and re-write the interface problem as a Laplacian of the solution with lower order derivative terms near the interface. Thus we can get jump relations for second order derivatives using the augmented variable and the lower order derivative terms. The idea should be applicable for boundary value problems as well. An upwind type discretization is used for the finite difference discretization near or on the interface so that the negative of the discrete coefficient matrix is an M-matrix, which is diagonally dominant and invertible. A multigrid solver is used to solve the linear system of equations and a GMRES iterative method is used to solve for the augmented variable. Numerical examples and convergence proof are also provided to show that the new method maintains second order accuracy of both the solution and the gradient.

Accurate Gradient Computation for Elliptic Interface Problems with Discontinuous and Variable Coefficients

by
Guanyu Chen

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2015

APPROVED BY:

Dr. Alina Chertock	Dr. Ernest Stitzinger
Dr. Xiaobiao Lin	Dr. Zhilin Li Chair of Advisory Committee

DEDICATION

To my wife, Yan Li

To my parents, Jian Chen and Ping Wang

To my grandparents, Changbao Chen and Yufang Jing

BIOGRAPHY

Guanyu Chen was born in a small town in Eastern China called Haian in 1986. He graduated from Haian High School and started his undergraduate studies at Nanjing University in 2004. Several months after he started college, he had the rare opportunity of competing for a full scholarship to study in Hong Kong. After three rounds of selections, he was among the seven students who were awarded the full scholarship that year to go to Hong Kong Baptist University. He earned his Bachelor's degree in Applied and Computational Mathematics with Minor in Finance from Hong Kong Baptist University in 2008. After that, to pursue higher level academic goals, he decided to come to the United States and joined the Applied Mathematics Graduate Program at NC State University in Raleigh. He started his research in the area of numerical mathematics with Dr. Zhilin Li since 2010. He worked as a Research Assistant for the Department of Mathematics at NC State University during regular semesters and had some internship experiences in investment banking and capital management during summers. He received his Master's degree in Financial Mathematics in 2010.

ACKNOWLEDGEMENTS

It is my honor to thank the many people who have helped me through this work.

I would like to express my deepest gratitude to my advisor and mentor, Dr. Zhilin Li, for his guidance, training and support throughout my PhD study. These last five years have been amazing for me. I have been able to grow as a student, and more importantly, as a man. He not only taught me the knowledge needed to be successful in research but also led by example the correct attitude towards challenges on our way. His enthusiasm, meticulousness and great patience helped me continuously to accomplish in my research work. The experience I learned from his guidance will be an invaluable part of my life.

I would like to thank my committee members, Dr. Alina Chertock, Dr. Ernest Stitzinger and Dr. Xiaobiao Lin for reviewing my dissertation and offering valuable suggestions. I would also like to thank Mrs. Denise Seabrooks for her kind help during my six years in the graduate program.

I also want to thank Haifeng Ji and Professor Jinru Chen from Nanjing Normal University for their collaborations and discussions on my research. Thanks also go to my dear friends at NC State University for their help and collaborations on the research projects: Peng Song, Sidong Zhang, Zhaohui Wang, Tom and Mami Wentworth, Paige Shy, Ralph Abbey, Katherine Varga, Ming Jiang, Min Yang, Anna Fregosi, Gadi Elamami and Ranya Ali.

Finally, I want to thank my wife, Yan Li; my parents, Jian Chen and Ping Wang; and my grandparents, Changbao Chen and Yufang Jing for their unconditional love and support. You are the reason I keep moving forward, opening new doors and taking on new challenges.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Interface problems

1.1.1 Applications of interface problems

Interface problems have attracted lots of attentions from many mathematicians and physicists, not only because it is typically challenging to solve them, either analytically or numerically, but also because of their diverse range of real world applications. Interface problems occur in many multi-physics or multi-phase applications in science and engineering, for example, the heat propagating in materials with different conductivities in thermodynamics, a shock wave traveling through materials with different viscosities in fluid dynamics, thin film and crystal growth simulations in material science, or a bubble formation and movement in biological science, and so on [6]. These real world phenomenons can be modeled by different types of partial differential equations (PDEs) with some specified interface conditions. The parameters in the governing PDE models are typically nonsmooth and discontinuous across the interface separating two materials

or two states, and the source terms are often singular. As a result, the solutions to the differential equations are typically nonsmooth, or even discontinuous across the interface. Therefore, many standard numerical methods based on the assumption of smoothness of solutions do not work or work poorly for interface problems, due to the complications arise from the presence of interfaces, discontinuities in the coefficients and the singular source terms.

To develop numerical methods for interface problems, the main challenges lie in how to obtain an approximate solution with certain order of accuracy near or on the interface and how to efficiently solve the linear system of equations involved. Our work in this dissertation is also centered around these two key questions.

1.1.2 Elliptic PDEs with interfaces

In this dissertation, we're interested in elliptic interface problems. An elliptic interface problem can be written in the following form,

$$\nabla \cdot \big(\beta(x,y)\nabla u(x,y)\big) + \sigma(x,y)u(x,y) = f(x,y), \quad (x,y) \in \Omega = \Omega^+ \cup \Omega^-, \tag{1.1}$$

together with the jump conditions across the interface Γ,

$$[u]_\Gamma = w, \qquad [\beta u_{\mathbf{n}}]_\Gamma = v, \tag{1.2}$$

where Ω is a rectangular domain with a prescribed boundary condition on $\partial\,\Omega$ and $\beta \geq \beta_{\min} > 0$. Within this domain, Γ is an interface between two subdomains Ω^+ and Ω^-, across which, the diffusion coefficient β is discontinuous, see Figure 1.1 for an illustration. $u_{\mathbf{n}}$ is the normal derivative on the interface, which is defined as $u_{\mathbf{n}} = \dfrac{\partial u}{\partial \mathbf{n}} = \nabla u \cdot \mathbf{n}$, where

n is the normal direction of the interface Γ pointing outward. Moreover, the coefficient $\beta(x, y)$ is variable in each subdomain,

$$\beta(x,y) = \begin{cases} \beta^+(x,y) & \text{if } (x,y) \in \Omega^+, \\ \beta^-(x,y) & \text{if } (x,y) \in \Omega^-. \end{cases} \tag{1.3}$$

The functions w and v are two jump conditions defined only along the interface Γ. σ and f are piecewise continuous functions, i.e., $\sigma^{\pm}(x, y) \in C$, $f^{\pm}(x, y) \in C$, but may have a finite jump discontinuity across the interface. The interface Γ is assumed to be twice continuously differentiable along the interface, i.e., $\Gamma \in C^2$. The solution is assumed to be twice continuously differentiable in each subdomain, i.e., $u^{\pm}(x, y) \in C^2$. The diffusion coefficient $\beta(x, y)$ is assumed to be continuously differentiable in each subdomain, i.e., $\beta^{\pm}(x, y) \in C^1$.

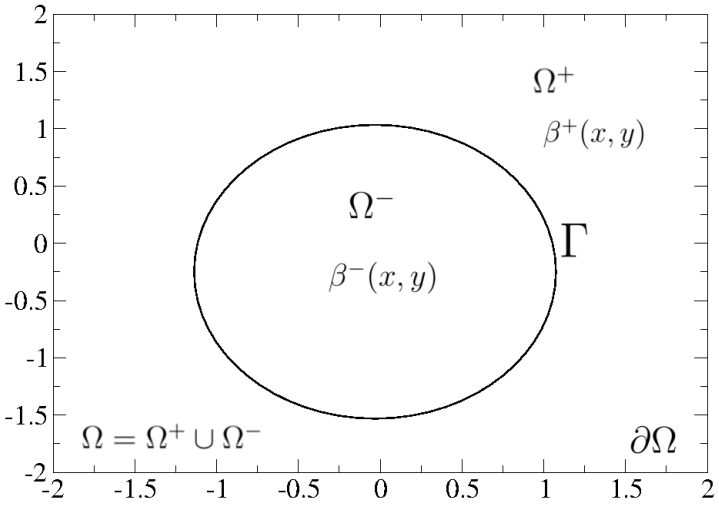

Figure 1.1: A typical rectangular domain $\Omega = \Omega^+ \cup \Omega^-$ with an interface Γ. The coefficients $\beta(\mathbf{x})$ have a finite jump across the interface Γ.

The jump conditions across the interface Γ are defined as:

$$[u]_\Gamma \stackrel{\text{def}}{=} u^+\big(X(s),Y(s)\big) - u^-\big(X(s),Y(s)\big) = w(s), \tag{1.4}$$

$$\begin{aligned}[\beta u_{\mathbf{n}}]_\Gamma \stackrel{\text{def}}{=} &\beta^+\big(X(s),Y(s)\big)u_{\mathbf{n}}^+\big(X(s),Y(s)\big) \\ &-\beta^-\big(X(s),Y(s)\big)u_{\mathbf{n}}^-\big(X(s),Y(s)\big) = v(s),\end{aligned} \tag{1.5}$$

where $\big(X(s),Y(s)\big)$ is the arc-length parametrization of the interface Γ. The $+$ and $-$ signs are assigned to be the limiting values of a function taken from the subdomain Ω^+ and the subdomain Ω^-.

Note that if the first jump condition $[u] = w = 0$, the solution to the interface problem (1.1) is equivalent to the solution of the following equation in the entire domain including the interface ($\Omega^+ \cup \Omega^- \cup \Gamma$), with a singular source on the right hand side,

$$\nabla \cdot \big(\beta(x,y)\nabla u(x,y)\big) + \sigma(x,y)u(x,y) = f(x,y) + \int_\Gamma v(s)\delta(\mathbf{x}-\mathbf{X}(s))\,ds\,. \tag{1.6}$$

where δ is the two-dimensional Dirac-delta function. The second jump condition $[\beta u_{\mathbf{n}}] = v$ can then be derived by integrating the above equation,

$$\begin{aligned}\lim_{\Omega_0\to 0}\iint_{\Omega_0}\big\{\nabla\cdot\big(\beta(\mathbf{x})\nabla u(\mathbf{x})\big) + \sigma(\mathbf{x})u(\mathbf{x}) - f(\mathbf{x}) - \int_\Gamma v(s)\delta(\mathbf{x}-\mathbf{X}(s))\,ds\big\}\,d\mathbf{x} = 0 \\ \Longrightarrow \int_\Gamma \big\{\big(\beta^+(\mathbf{x})\nabla u^+(\mathbf{x})\cdot\mathbf{n}\big) - \big(\beta^-(\mathbf{x})\nabla u^-(\mathbf{x})\cdot\mathbf{n}\big) - v(s)\big\}\,ds = 0 \\ \Longrightarrow [\beta u_{\mathbf{n}}]_\Gamma = v(s)\end{aligned} \tag{1.7}$$

where Ω_0 is a small integration domain that contains the entire interface Γ. The second line in the derivation comes naturally from the divergence theorem and the properties

of delta function. Note that the normal derivative $u_{\mathbf{n}}$ is usually discontinuous across the interface due to the discontinuity in the coefficient β. And if $w \neq 0$, the solution u itself would also be discontinuous across the interface.

1.2 A review of numerical methods for elliptic interface problems

There are many applications in solving elliptic equations with discontinuous coefficients, for example, steady state heat diffusion, multi-phase flow, crystallization process, and bubble simulation, and etc. Moreover, solving one or several elliptic interface problems is also the most expensive step of several well-known efficient methods for Navier-Stokes equations. There are two main concerns in solving (1.1)–(1.3). The first concern is about how to discretize (1.1)–(1.3) accurately. It is difficult to study the consistency and the stability of a numerical scheme because of the discontinuities across the interface. The second concern is about how to solve the resulting linear systems efficiently and robustly. Usually if the jump in the coefficient is large, the resulting linear system is ill-conditioned, and the number of iterations needed in solving such a linear system is large. That's because the number of iterations usually is proportional to the jump in the coefficient.

Quite a few numerical methods in the literatures have addressed how to solve the elliptic equations with discontinuous coefficients, for example, harmonic averaging, smoothing method, and finite element approach, etc. Most of these methods are second order accurate in the L^1 or the L^2 norm, but not in the L^∞ norm, since they usually smooth out the solution near the interface. So in the discussion below, we review a few popular finite difference methods for the elliptic interface problems.

1. **The smoothing method**

The idea of the smoothing method is to replace the original discontinuous coefficient $\beta(x)$ with a smoothing function β_ϵ. We illustrate the idea through a one-dimensional example. Assume the coefficient $\beta(x)$ has a finite jump at $x = \alpha$, i.e., $[\beta]_\alpha = \lim_{x\to\alpha^+} \beta(x) - \lim_{x\to\alpha^-} \beta(x) \neq 0$.

We define a smoothing function β_ϵ as

$$\beta_\epsilon(x) = \bar{\beta}^-(x) + (\bar{\beta}^+(x) - \bar{\beta}^-(x))H_\epsilon(x-a), \tag{1.8}$$

where $\bar{\beta}^+(x)$ and $\bar{\beta}^-(x)$ are continuously differentiable functions given by

$$\bar{\beta}^-(x) = \begin{cases} \beta(x) & \text{if } x \leq \alpha, \\ \beta(\alpha-) + \beta'(\alpha-)(x-\alpha) & \text{if } x > \alpha, \end{cases} \tag{1.9a}$$

$$\bar{\beta}^+(x) = \begin{cases} \beta(\alpha+) + \beta'(\alpha+)(x-\alpha) & \text{if } x < \alpha, \\ \beta(x) & \text{if } x \geq \alpha. \end{cases} \tag{1.9b}$$

$H_\epsilon(x)$ is the smoothed Heaviside function,

$$H_\epsilon(x) = \begin{cases} 0 & \text{if } x \leq -\epsilon, \\ \frac{1}{2}\big(1 + \frac{x}{\epsilon} + \frac{1}{\pi}\sin\frac{\pi x}{\epsilon}\big) & \text{if } |x| \leq \epsilon, \\ 1 & \text{if } x > \epsilon. \end{cases} \tag{1.10}$$

and $\epsilon > 0$ is a small number depending on the mesh size of a numerical scheme [26]. The coefficient $1/\pi$ in front of the sine function is chosen to make $H_\epsilon(x)$ twice continuously differentiable at $x = \pm\epsilon$.

It is easy to implement the smoothing method in 1D, 2D and 3D if the interface is represented by the zero level set of a Lipschitz continuous function. The smoothing method is not very accurate because it smoothens the coefficient β in the interface, as as result, the solution is also smoothened in the interface.

2. **The harmonic averaging method**

The harmonic averaging method [3, 25, 28] is more accurate than the smoothing method for discontinuous coefficients. Consider the one-dimensional model problem $(\beta u_x)_x = f(x)$, which can be discretized as

$$\frac{1}{h^2}\Big(\beta_{i+\frac{1}{2}}(u_{i+1} - u_i) - \beta_{i-\frac{1}{2}}(u_i - u_{i-1})\Big) = f(x_i), \tag{1.11}$$

where $h = x_i - x_{i+1}$ is the uniform grid spacing in the x-direction. For smooth β in (x_{i-1}, x_{i+1}), we can take $\beta_{i+\frac{1}{2}} = \beta(x_{i+\frac{1}{2}})$, where $x_{i+\frac{1}{2}} = x_i + \dfrac{h}{2}$, and the discretization is second-order accurate. But if β is discontinuous in (x_{i-1}, x_{i+1}), then the harmonic average of $\beta(x)$ is

$$\beta_{i+\frac{1}{2}} = \Big(\frac{1}{h}\int_{x_i}^{x_{i+1}} \beta^{-1}(x)\, dx\Big)^{-1}. \tag{1.12}$$

The finite difference scheme (1.11) using the harmonic averaging (1.12) is second order accurate in the infinity norm L^∞ for 1D elliptic interface problems with $[u]_\alpha = [\beta u_x]_\alpha = [f]_\alpha = 0$, mainly due to the cancellation of errors.

The harmonic averaging can also deal with 2D elliptic interface problems by integrating over squares to get the harmonic average of $\beta(x, y)$. However, the method usually does not give second order accurate solutions, because the cancellations of errors are unlikely to occur for arbitrary interfaces. Moreover, it is also nontrivial to compute the integrals accurately near the interface in 2D when β is discontinuous.

3. Peskin's immersed boundary (IB) method

To simulate the blood flow pattern in human's heart, Peskin first developed the immersed boundary (IB) method [21, 22, 23, 20], which used numerical approximation of the δ function for singular sources on the interface.

There are several discrete delta functions in the literatures. The commonly used ones include the hat function

$$\delta_\epsilon(x) = \begin{cases} \dfrac{(\epsilon - |x|)}{\epsilon} & \text{if} \quad |\, x \,| < \epsilon, \\ 0 & \text{if} \quad |\, x \,| \geq \epsilon. \end{cases} \tag{1.13}$$

and Peskin's original discrete cosine delta function

$$\delta_\epsilon(x) = \begin{cases} \dfrac{1}{4\epsilon}\left(1 + \cos\left(\dfrac{\pi x}{2\epsilon}\right)\right) & \text{if} \quad |\, x \,| < 2\epsilon, \\ 0 & \text{if} \quad |\, x \,| \geq 2\epsilon. \end{cases} \tag{1.14}$$

These two discrete delta function are both continuous. The second one, first introduced by Peskin and most often used in the literature, is also continuously differentiable.

It is easy to implement the IB method. In high dimensions, the discrete delta function is the product of one-dimensional discrete delta functions, for example, in 2D, $\delta_\epsilon(x, y) = \delta_\epsilon(x)\delta_\epsilon(y)$. With Peskin's discrete delta function approach, we can discretize the right hand side of (1.6) at a grid point (x_i, y_j) as

$$F_{ij} = f_{ij} + \sum_{k=1}^{m} v(s_k)\delta_h(x_i - X_k)\delta_h(y_j - Y_k)\Delta s, \tag{1.15}$$

where N_b is the number of discrete control points $\{X_k, Y_k\}$ on the interface, and

h is the mesh spacing. Note that, from (1.13) and (1.14), we see that the singular source is distributed to grid points near the interface Γ.

The IB method is difficult to achieve high order accuracy. It is still a smoothing method that smears discontinuities. Using the IB method, we can achieve second order accurate solution in an average norm such as the L^1 norm or the L^2 norm. But it is unlikely to achieve a second-order accurate solution in the point-wise L^∞ norm. The reason is that the discrete right-hand size (1.15) is independent of the derivative of $v(s)$ and the curvature of the interface, which are crucial in the immersed interface method (IIM) [7, 10] which we will discuss later in section 1.3.

4. **Numerical methods based on integral equations**

   A. Mayo and A. Greenbaum [18, 19] have derived an integral equation for elliptic interface problems with piecewise constant coefficients. By solving the integral equation, they can obtain second-order accurate solutions in the L^∞ norm using the techniques developed by Mayo in [16, 18] for solving Poisson and biharmonic equations on irregular domains. The total cost includes solving the integral equation and a regular Poisson equation. They also mentioned the possibility to extend the method to variable coefficients in [18]. While this methods based on integral equations are very effective for homogeneous source terms, they usually require extra efforts for non-homogeneous source terms and different boundary conditions, for which the implementations of these methods are not trial.

So far, most of our discussions focus on numerical methods that smears the discontinuities in either the discontinuous coefficients or the singular source terms near the interface. As a result, the solutions are also smoothened near the interface. So they usually give second order accuracy in an average norm such as the L^1 and L^2 norm, but not

in the point-wise L^∞ norm. However, when dealing with interface problems, we are more interested in errors in the point-wise L^∞ norm instead of an average L^1 or L^2 norm. An average norm cannot correctly reflect errors near the interface, while the point-wise L^∞ norm usually reflects the accuracy of solutions near interfaces that are the main interest for many interface problems.

In the section below, we will introduce the immersed interface method (IIM) [7, 10], in which the discontinuities and the jump conditions are enforced either exactly or approximately. The IIM generally has a second-order accuracy globally, which means the computed solution is second-order accurate in the point-wise L^∞ norm.

1.3 The immersed interface method (IIM) and an augmented strategy

The immersed interface method (IIM) [7, 10] is a different approach developed by R. J. LeVeque and Z. Li for discretizing elliptic problems with irregular interfaces , which can handle both discontinuous coefficients and singular sources. The main idea is to incorporate the jump conditions into the finite difference scheme near the interface using Taylor expansion. This approach has also been applied to 3D elliptic equations [11], parabolic equations [12, 15, 17], hyperbolic wave equations with discontinuous coefficients [9], and the incompressible Stokes flow problems with moving interfaces [8].

1.3.1 The IIM for elliptic interface problems

In the IIM, the standard finite difference scheme is modified only when its finite difference stencil is cut by the interface. In other words, away from the interface, the IIM uses

the standard finite difference methods. while near or on the interface, the IIM modifies the finite different schemes to treat the irregularities. Since the dimension of the interfaces is one dimension lower than that of the solution domain, the modifications do not significantly increase the computational costs.

Here we illustrate the main idea of IIM through a one-dimensional model problem

$$(\beta u_x)_x = f, \quad x \in (0, \alpha) \cup (\alpha, 1), \tag{1.16a}$$

$$[u]_{x=\alpha} = 0, \quad [\beta u_x]_{x=\alpha} = v, \tag{1.16b}$$

with specified boundary conditions of $u(x)$ at $x = 0$ and $x = 1$. The function $\beta(x)$ is discontinuous at $x = \alpha$. While this model problem is quite simple, it illustrates the main ideas of the IIM.

For simplicity, we assume that $f(x)$ is a continuous function and β is piecewise constant with a finite jump at the interface $x = \alpha$. From the jump conditions (1.16b) and the equation (1.16a), we first obtain the following interface relations, which express the limiting values from the "+" side ($x > \alpha$) in terms of those from the "−" side ($x < \alpha$),

$$u^+ = u^-, \quad u_x^+ = \frac{\beta^-}{\beta^+} u_x^- + \frac{v}{\beta^+}, \quad u_{xx}^+ = \frac{\beta^- u_{xx}^-}{\beta^+} \tag{1.17}$$

The algorithm of IIM for (1.16a)–(1.16b) is then outlined below.

1. We first generate a uniform Cartesian grid for the finite difference method,

$$x_i = ih, \qquad i = 0, 1, 2, \ldots, N \tag{1.18}$$

where $h = 1/N$. The interface α is usually between grid points, for example, $x_j \leq$

$\alpha \leq x_{j+1}$. Then, the grid points x_j and x_{j+1} are called *irregular grid points*, while the other grid points are called *regular grid points*.

2. We then determine the finite difference scheme for regular grid points.

   At a grid point $x_i, x_i \neq j, j+1$, we use the standard 3-point central finite difference approximation

$$\frac{1}{h^2}\Big(\beta_{i+\frac{1}{2}}(U_{i+1} - U_i) - \beta_{i-\frac{1}{2}}(U_i - U_{i-1})\Big) = f_i \tag{1.19}$$

   with $\beta_{i+\frac{1}{2}} = \beta(x_{i+\frac{1}{2}})$, $f_i = f(x_i)$.

3. Next, we determine the finite difference scheme for the irregular grid points x_j and x_{j+1}.

   We determine the finite difference coefficients using the method of undetermined coefficients,

$$\begin{aligned} \gamma_{j,1}U_{j-1} + \gamma_{j,2}U_j + \gamma_{j,3}U_{j+1} &= f_j + C_j, \\ \gamma_{j+1,1}U_j + \gamma_{j+1,2}U_{j+1} + \gamma_{j+1,3}U_{j+2} &= f_{j+1} + C_{j+1}. \end{aligned} \tag{1.20}$$

   We now illustrate the idea of the IIM about how to determine the finite difference coefficients $\gamma_{j,1}$, $\gamma_{j,2}$ and $\gamma_{j,3}$ in (1.20).

   We want to minimize the magnitude of the local truncation error

$$T_j = \gamma_{j,1}u(x_{j-1}) + \gamma_{j,2}u(x_j) + \gamma_{j,3}u(x_{j+1}) - f(x_j) - C_j\,. \tag{1.21}$$

   The main idea is to use Taylor expansion to expand the solution $u(x_{j-1})$, $u(x_j)$, $u(x_{j+1})$ and $f(x_j)$ at the interface α from each side of the interface.

The Taylor expansion for $u(x_{j+1})$ at α is given by

$$u(x_{j+1}) = u^+(\alpha) + (x_{j+1} - \alpha)u_x^+(\alpha) + \frac{1}{2}(x_{j+1} - \alpha)^2 u_{xx}^+(\alpha) + \mathcal{O}(h^3)\,.$$

and the Taylor expansions of $u(x_{j-1})$ and $u(x_j)$ at α can be written as

$$u(x_l) = u^-(x) + (x_l - \alpha)u_x^-(\alpha) + \frac{1}{2}(x_l - \alpha)^2 u_{xx}^-(\alpha) + \mathcal{O}(h^3), \quad l = j-1,\, j.$$

Then we use the interface relations (1.17) and choose to wirte $u^+(\alpha)$, $u_x^+(\alpha)$, $u_{xx}^+(\alpha)$ in terms of $u^-(\alpha)$, $u_x^-(\alpha)$, $u_{xx}^-(\alpha)$. So we have a new expression for the Taylor expansion of $u(x_{j+1})$:

$$u(x_{j+1}) = u^-(\alpha) + (x_{j+1} - \alpha)\Big(\frac{\beta^-}{\beta^+}u_x^-(\alpha) + \frac{v}{\beta^+}\Big) + \frac{1}{2}(x_{j+1}^2 - \alpha)^2\frac{\beta^-}{\beta^+}u_{xx}^-(\alpha) + \mathcal{O}(h^3)\,.$$

Thus, we now put all these expansions back to the local truncation error (1.21) at $x = x_j$, and collect terms for $u^-(\alpha)$, $u_x^-(\alpha)$, $u_{xx}^-(\alpha)$ to get

$$\begin{aligned}
T_j &= \gamma_{j,1}u(x_{j-1}) + \gamma_{j,2}u(x_j) + \gamma_{j,3}u(x_{j+1}) - f(x_j) - C_j \\
&= (\gamma_{j,1} + \gamma_{j,2} + \gamma_{j,3})u^-(\alpha) + \gamma_{j,3}(x_{j+1} - \alpha)\frac{v}{\beta^+} \\
&\quad + \Big((x_{j-1} - \alpha)\gamma_{j,1} + (x_j - \alpha)\gamma_{j,2} + \frac{\beta^-}{\beta^+}(x_{j+1} - \alpha)\gamma_{j,3}\Big)u_x^-(\alpha) \\
&\quad + \frac{1}{2}\Big((x_{j-1} - \alpha)^2\gamma_{j,1} + (x_j - \alpha)^2\gamma_{j,2} + \frac{\beta^-}{\beta^+}(x_{j+1} - \alpha)^2\gamma_{j,3}\Big)u_{xx}^-(\alpha) \\
&\quad - f(\alpha) - \mathcal{O}(h) - C_j + \mathcal{O}\Big(\max_{1\le l\le 3}|\gamma_{j,l}|h^3\Big)\,.
\end{aligned} \tag{1.22}$$

Finally, by minimizing the magnitude of T_j and using the differential equation at

α from the "$-$" side, we obtain a system of equations for the coefficients $\{\gamma_{j,k}\}$:

$$\begin{cases} \gamma_{j,1} + \gamma_{j,2} + \gamma_{j,3} = 0\,, \\ (x_{j-1} - \alpha)\gamma_{j,1} + (x_j - \alpha)\gamma_{j,2} + \dfrac{\beta^-}{\beta^+}(x_{j+1} - \alpha)\gamma_{j,3} = 0\,, \\ \dfrac{1}{2}(x_{j-1} - \alpha)^2\gamma_{j,1} + \dfrac{1}{2}(x_j - \alpha)^2\gamma_{j,2} + \dfrac{1}{2}\dfrac{\beta^-}{\beta^+}(x_{j+1} - \alpha)^2\gamma_{j,3} = \beta^-\,. \end{cases} \tag{1.23}$$

Once those $\{\gamma_{j,k}\}$'s have been computed, the correction term is set to

$$C_j = \gamma_{j,3}(x_{j+1} - \alpha)\frac{v}{\beta^+}\,, \tag{1.24}$$

which matches the remaining leading terms in the local truncation error T_j above.

Similarly, we can get the $\{\gamma_{j+1,k}\}$'s by considering the local truncation error T_{j+1} at $x = x_{j+1}$ and following the same procedure above.

4. We can solve the linear system of equations (1.19)–(1.20) for all grid points, which is a tridiagonal matrix, to get an approximate solution of $u(x)$ at all grid points.

For 2D and 3D elliptic interface problems, we can use the same IIM algorithm to derive the finite difference scheme for the irregular points on or near the interface.

Since the IIM incorporates the discontinuities and the jump conditions on the interface, it can achieve second-order accuracy in the point-wise L^∞ norm. The second order accuracy of the IIM has been confirmed by many numerical examples and theoretical analysis. Note that, while the solutions have second order accuracy globally at all grid points, the local truncation errors at grid points near the interface are usually $\mathcal{O}(h)$ (see (1.22) for example), which is one order lower than that at regular grid points ($\mathcal{O}(h^2)$).

The finite difference coefficient matrix is usually a block tridiagonal sparse matrix

for 2D and 3D interface problems. So we can use standard iterative methods such as an SOR or the multigrid method [2, 4] to solve the linear system of equations efficiently. But in some numerical examples with large jumps in the coefficients β, the resulting linear system is usually ill-conditioned. So it takes many number of iterations in solving such a system. The immersed interface method may converge very slowly or fail to give accurate answers.

In the next section, we explain an augmented strategy which can be used to solve some interface problems with large jumps in the diffusion coefficient $\beta(x, y)$.

1.3.2 An augmented iterative algorithm for the IIM

The idea of the augmented strategy for interface problems was originally proposed by Z. Li [13] for elliptic interface problems with piecewise constant coefficients.

The augmented strategies have at least two advantages. The first is that the augmented strategies allow us to utilize the existing fast solvers to get a faster algorithm compared to direct discretization. The second is that, for certain types of interface problems, an augmented approach may be the only way to obtain an accurate algorithm. For instance, in the incompressible Stokes equations, the jump conditions for the pressure and the velocity are usually coupled together. The augmented approach enables us to separate the jump conditions so that the idea of the IIM can be applied.

In augmented strategies, some augmented variable g defined only on the interface is introduced. If the augmented variable is known, it is relatively easy to solve the original problem.

There are many ways to introduce an augmented variable. In Z. Li's original paper, he considered the elliptic PDEs with $\sigma(x, y) = 0$. The coefficient $\beta(x, y)$ is assumed to

be constant in each subdomain,

$$\beta(x,y) = \begin{cases} \beta^+ & \text{if } (x,y) \in \Omega^+, \\ \beta^- & \text{if } (x,y) \in \Omega^-. \end{cases} \tag{1.25}$$

With this piecewise constant coefficient, the original PDE in (1.1) with $\sigma(x,y) = 0$ can be preconditioned to a Poisson equation in each subdomain by dividing the constant β^+ in the Ω^+ subdomain and dividing β^- in the Ω^- subdomain from the original problem. So it is natural to introduce the jump in the normal derivative $[u_{\mathbf{n}}]$ as the augmented variable. Thus, we have an equivalent problem:

$$\begin{cases} \nabla^2 u(x,y) = \dfrac{f(x,y)}{\beta^+}, & \text{if } (x,y) \in \Omega^+, \\ \nabla^2 u(x,y) = \dfrac{f(x,y)}{\beta^-}, & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{1.26a}$$

$$[u]_\Gamma = w, \qquad [u_{\mathbf{n}}]_\Gamma = g, \tag{1.26b}$$

with the same boundary condition on $\partial\Omega$ as in the original problem (1.1). The regularity of the new problem is the same as previously mentioned in Section 1.1.2, i.e., f is piecewise continuous, $f^{\pm}(x,y) \in C$; the interface Γ is twice continuously differentiable along the interface, $\Gamma \in C^2$; and the solution u is piecewise twice continuously differentiable, $u^{\pm}(x,y) \in C^2$.

Then we can discretize the corresponding Poisson equation using the standard five-point stencil with some modifications in the right hand side. The discrete form of (1.26a) obtained from the IIM can be written as

$$\nabla_h^2 U_{ij} = \frac{f_{ij}}{\beta_{ij}} + C_{ij}\,, \tag{1.27}$$

where C_{ij} is a correction term, which is zero at a regular grid points, and is non-zero at irregular grid points, and the ∇_h^2 is the discrete Laplacian operator

$$\nabla_h^2 U_{ij} = \frac{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{i,j}}{h^2}\,. \tag{1.28}$$

The correction term C_{ij} at an irregular grid point (x_i, y_j) can be derived from the IIM and is given as

$$\begin{aligned} C_{ij} = {} & a_2 w + a_{12} g' + (a_6 + a_{12}\chi'')w' + a_{10} w'' \\ & + \Big\{ a_4 + (a_8 - a_{10})\chi'' \Big\} g + a_8 \Big\{ \left[\frac{f}{\beta}\right] - w'' \Big\}\,, \end{aligned} \tag{1.29}$$

where w, w', w'', g, and g' are evaluated at a set of control points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_b}$ on the interface. These control points are usually the orthogonal projections of the irregular grid points onto the interface. The $\{a_i\}$'s are given by (3.4) in Chapter 3, which depends on the finite difference coefficients of the 9-point stencil centered at (i, j) and the positions of the 9-point stencil relative to the interface. We will explain this in detail when we generalize the augmented strategies to PDEs with piecewise variable coefficients.

Let $\mathbf{W} = [W_1, W_2, \ldots, W_{N_b}]^T$ and $\mathbf{G} = [G_1, G_2, \ldots, G_{N_b}]^T$ be the discrete values of the jump conditions (1.26b) at the control points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_b}$ on the interface. Let $\mathcal{B}(\mathbf{W}, \mathbf{G})$ be a mapping from $\mathbf{W} = [W_1, W_2, \ldots, W_{N_b}]^T$ and $\mathbf{G} = [G_1, G_2, \ldots, G_{N_b}]^T$ to C_{ij} in (1.29). In discrete form, all the surface derivatives of the jump conditions can be obtained from a linear combination of the values of $\mathbf{W}$ and $\mathbf{G}$ at $\{\mathbf{X}_k\}$. Therefore, $\mathcal{B}(\mathbf{W}, \mathbf{G})$ can be written as a linear function of $\mathbf{W}$ and $\mathbf{G}$

$$\mathcal{B}(\mathbf{W}, \mathbf{G}) = B\mathbf{G} - B_1\mathbf{W} \tag{1.30}$$

where B and B_1 are two matrices with real entries.

Therefore, (1.27) can be re-written as a matrix vector equation, which combines the approximate solution (denoted by $\mathbf{U}$) to the original problem and the augmented variable $\mathbf{G}$ (discrete form of g) together

$$A\mathbf{U} + B\mathbf{G} = \mathbf{F} + B_1\mathbf{W} \stackrel{\text{def}}{=} \mathbf{F}_1 \,. \tag{1.31}$$

where A is the matrix form of the discrete Laplacian operator and $\mathbf{F}$ is the vector formed by $\{\frac{f_{ij}}{\beta_{ij}}\}$. In (1.31), for a given augmented variable $\mathbf{G}$, we can solve for the approximate solution $\mathbf{U}$. In Z. Li's original paper, fast Poisson solvers are utilized in solving for $\mathbf{U}$ to give a fast algorithm.

Equation (1.31) has two unknowns, $\mathbf{U}$ and $\mathbf{G}$. So we need another constraint to form the second equation. We use the flux jump condition $[\beta u_{\mathbf{n}}] = v$ as the constraint. Define the residual vector of the flux jump condition at $\{\mathbf{X}_k\}$ as

$$\mathbf{R}(\mathbf{G}) = [\beta \mathbf{U}_{\mathbf{n}}](\mathbf{G}) - \mathbf{V} = \beta^+ \mathbf{U}_{\mathbf{n}}^+ - \beta^- \mathbf{U}_{\mathbf{n}}^- - \mathbf{V} \,. \tag{1.32}$$

We want to find a $\mathbf{G}^*$ such that $\mathbf{R}(\mathbf{G}^*) = 0$, where the vectors $\mathbf{U}_{\mathbf{n}}^+$ and $\mathbf{U}_{\mathbf{n}}^-$ are the discrete approximations of the normal derivatives $u_{\mathbf{n}}^{\pm}$ at $\{\mathbf{X}_k\}$ from each side of the interface.

For an approximate $\mathbf{G}$, we can obtain the solution $\mathbf{U}$ by solving (1.31). Then, we can interpolate $\{U_{ij}\}$ in a linear manner to get $U_{\mathbf{n}}^{\pm}(\mathbf{X}_k)$ at the control points $\{\mathbf{X}_k\}$, $1 \le k \le N_b$. The interpolation scheme is crucial to the success of the augmented algorithm, and the weighted least squares interpolation is used, which we will explain more in depth in Chapter 3 when we generalize the augmented algorithm to PDEs with piecewise variable

coefficient β. Since the interpolation is linear, we can write

$$\frac{\partial \mathbf{U}^{\pm}(\mathbf{G})}{\partial \mathbf{n}} = E^{\pm}\mathbf{U} + T^{\pm}\mathbf{G} + P^{\pm}\mathbf{V} + Q^{\pm}\mathbf{W}, \tag{1.33}$$

where E^+, E^-, T^+, T^-, P^+, P^-, Q^+, Q^- are some sparse matrices determined from the interpolation scheme. These matrices are used only for theoretical purposes but are not actually constructed in implementation. We need to choose a vector $\mathbf{G}$ such that the flux jump condition $\beta^+\mathbf{U}_{\mathbf{n}}^+ - \beta^-\mathbf{U}_{\mathbf{n}}^- = \mathbf{V}$ is satisfied along the interface Γ. Therefore, we have a second matrix-vector equation

$$E\mathbf{U} + T\mathbf{G} - P\mathbf{V} - Q\mathbf{W} = 0\,. \tag{1.34}$$

If we eliminate $\mathbf{U}$ from the matrix vector equations (1.31) and (1.34), we get the Schur complement system for the augmented variable,

$$(T - EA^{-1}B)\mathbf{G} = P\mathbf{V} + Q\mathbf{W} - EA^{-1}\mathbf{F}_1 \stackrel{\text{def}}{=} \mathbf{F}_2\,. \tag{1.35}$$

This is an $N_b \times N_b$ system for $\mathbf{G}$, a much smaller linear system compared to the one for $\mathbf{U}$. Therefore, we can use the GMRES [24] iterative methods to solve the Schur complement system for the augmented variable. The matrix vector multiplication in GMRES iteration includes two main steps: (1) solving the original problem (1.31) for a given augmented variable $\mathbf{G}$; (2) finding the residual of the constraint (1.32) using the computed approximate solution from the given augmented variable.

The augmented IIM, also called the fast IIM, has been demonstrated to be second order accurate in solutions in the L^∞ norm by numerous numerical experiments. Moreover, this augmented algorithm has shown to be very efficient, because the number of GMRES

iterations is reasonably small and is independent of both the jumps in the coefficients and the mesh size.

1.4 Motivation to generalize the augmented strategy to variable coefficients

Despite of the great success of the augmented IIM for piecewise constant coefficients, this method has never been applied to solve the elliptic interface equations with piecewise variable coefficients.

Generally speaking, to develop an augmented algorithm for variable coefficients, we need to solve four major problems. Firstly, we need to develop a new finite difference scheme at the irregular points since the discrete laplacian operator no long works for variable coefficients. Secondly, as a result of the first problem, we cannot take advantage of the fast Poisson solver [27]. Instead, we need to utilize a multigrid solver [5, 1], which is comparable to a fast Poisson solver using an FFT. Thirdly, the interpolation scheme in Z. Li's original paper only works for piecewise constant coefficients. So we need to develop a generalized weighted least squares interpolation scheme for variable coefficients β to compute the normal derivatives on the interface from a grid function U_{ij}. The accuracy of this interpolation scheme is crucial for the success of the augmented algorithm. Last but not least, we need to propose an efficient new preconditioner to solve the Schur complement system since the original one proposed in Z. Li's paper [13] converges very slowly for interface problems with piecewise variable coefficients.

In this dissertation, we develop a new numerical method for interface problems with piecewise variable and discontinuous coefficients. The main motivation is to get not only

a second order accurate solution, but also a second order accurate gradient for some types of interface problems.

The idea is based on the augmented IIM, also called the fast IIM developed for interface problems with piecewise constant coefficients, in which, second order convergence of the solution and the gradient is achieved. The key of the new method is to introduce the jump in the normal derivative of the solution as the augmented variable and re-write the interface problem as a Laplacian of the solution with lower order derivative terms near the interface. Thus we can get jump relations for second order derivatives using the augmented variable and the lower order derivative terms. The idea should be applicable for boundary value problems with irregular domains as well. An upwind type discretization is used for the finite difference discretization near or on the interface so that the negative of the discrete coefficient matrix is an M-matrix. A multigrid solver DMGD9V [5] is used to solve the linear system of equations and a GMRES iterative method is used to solve for the augmented variable. Numerical experiments and convergence proof are also provided to show that the new method archives second order accuracy not only in the solution, but also in its gradient in the point-wise L^∞ norm.

This dissertation is organized as follows.

Chapter 2 describes how to use the augmented IIM strategy to construct all the components of the linear systems. In Chapter 2, we first precondition the original elliptic interface equation to get an equivalent problem, which we can develop an augmented iterative algorithm to solve. Next we derive for the interface relations from the jump conditions and the PDE. Then we use the augmented IIM idea to discretize the equivalent problem and derive the Schur complement system.

Chapter 3 describes how to utilize iterative algorithms to solve the linear system of equations. In Chapter 3, we first develope a second order least squares interpolation

scheme to approximate the discrete normal derivatives $\mathbf{U}_{\mathbf{n}}^{\pm}$ from the grid function U_{ij}. Next, we use the multigrid solver DMGD9V to solve the linear system of equations for $\mathbf{U}$ and a GMRES iterative method to solve for the augmented variable $\mathbf{G}$. The we propose an efficient preconditioner for the Schur complement system, which accelerates the convergence of the GMRES iterations.

Chapter 4 shows some numerical experiments and analysis to demonstrate that this new method can achieve the second order accuracy not only in the solution itself, but also in its gradient. Moreover, we also demonstrate the efficiency and robustness of the new method by showing that the number of iterations is almost independent of the mesh size and the ratio of the jump in the coefficients $\beta(x, y)$. At the end of this chapter, we also generalize our new method for elliptic interface problems with non-zero $\sigma(x, y)$ and demonstrate that it can achieve second order accuracy in both the solution and the gradient for these more general cases.

Finally, in Chapter 5, we summarize the main ideas of our new method and address the highlights of the new method. We also provide potential extension of this work in the end.

Chapter 2

Elliptic Interface Problems with Variable Coefficients

Consider the elliptic interface problem,

$$\begin{aligned} \nabla \cdot \big(\beta(x,y)\nabla u(x,y)\big) &= f(x,y), \quad (x,y) \in \Omega^+ \cup \Omega^-, \\ [u]_\Gamma = w, &\qquad [\beta u_{\mathbf{n}}]_\Gamma = v, \end{aligned} \tag{2.1}$$

in a rectangular domain Ω with a prescribed boundary condition on $\partial\Omega$ and $\beta \geq \beta_{\min} > 0$. Within this domain, Γ is an interface, across which the coefficient β is discontinuous, as illustrated in Figure 1.1. Moreover, the coefficient $\beta(x,y)$ is variable in each subdomain,

$$\beta(x,y) = \begin{cases} \beta^+(x,y) & \text{if } (x,y) \in \Omega^+, \\ \beta^-(x,y) & \text{if } (x,y) \in \Omega^-. \end{cases} \tag{2.2}$$

The functions w and v are two jump conditions that are defined only along the interface Γ. f is piecewise continuous but may have a finite jump discontinuity across the interface

Γ. The interface Γ may or may not align with an underlining Cartesian grid.

The regularity of this problem is the same as mentioned in Chapter 1, i.e., the interface Γ is assumed to be twice continuously differentiable, $\Gamma \in C^2$; the right hand side f is assumed to be piecewise continuous in each subdomain, $f^{\pm}(x, y) \in C$; the solution u is assumed to be piecewise twice continuously differentiable in each subdomain, $u^{\pm}(x, y) \in C^2$; and the coefficient β is assumed to be piecewise continuously differentiable in each subdomain, $\beta^{\pm}(x, y) \in C^1$.

In this chapter, we first precondition (2.1) and (2.2) to get an equivalent problem and derive necessary interface relations from the equivalent problem. Then we use the IIM idea to discretize the equivalent problem and derive the Schur complement system.

2.1 The equivalent problem

The original elliptic interface problem is stated in *Problem* I,

Problem (I).

$$\nabla \cdot \big(\beta(x, y)\nabla u(x, y)\big) = f(x, y), \quad x \in \Omega = \Omega^+ \cup \Omega^-, \tag{2.3a}$$

$$\text{Give BC on } \partial\Omega, \tag{2.3b}$$

with jump conditions along the interface Γ specified as

$$[u]_\Gamma = w(s), \tag{2.4a}$$

$$[\beta u_{\mathbf{n}}]_\Gamma = v(s). \tag{2.4b}$$

By introducing the normal derivative on the interface $[u_{\mathbf{n}}]$ as the augmented variable,

we are interested in solving a new problem as stated in *Problem* II.

Consider the solution set $u_g(x,y)$ of the following problem as a functional of $g(s)$.

Problem (II).

$$\nabla^2 u + \frac{\nabla\beta^+(x,y)}{\beta^+(x,y)} \cdot \nabla u = \frac{f(x,y)}{\beta^+(x,y)}, \quad \text{if } x \in \Omega^+, \tag{2.5a}$$

$$\nabla^2 u + \frac{\nabla\beta^-(x,y)}{\beta^-(x,y)} \cdot \nabla u = \frac{f(x,y)}{\beta^-(x,y)}, \quad \text{if } x \in \Omega^-, \tag{2.5b}$$

$$\text{Give BC on } \partial\Omega, \tag{2.5c}$$

with specified jump conditions along the interface Γ

$$[u]_\Gamma = w(s), \tag{2.6a}$$

$$[u_{\mathbf{n}}]_\Gamma = g(s). \tag{2.6b}$$

Take the exact solution of *Problem* (I) as $u^*(x,y)$, and we define its corresponding normal derivative along the interface as

$$g^*(s) = [u^*_{\mathbf{n}}](s)\,. \tag{2.7}$$

Then $u^*(x,y)$ satisfies *Problem* (II) with $g(s) \equiv g^*$. In other words, if we specify $g(s) \equiv g^*$ in (2.6b) and solve *Problem* (II), the solution $u_{g^*}(x,y)$ we obtained will be exactly the same as the solution to *Problem* (I), i.e., $u_{g^*}(x,y) \equiv u^*(x,y)$. So $u_{g^*}(x,y)$ automatically satisfies the second flux jump condition in *Problem* (I),

$$\left[\beta \frac{\partial u_{g^*}}{\partial n}\right] = v(s). \tag{2.8}$$

Therefore, solving *Problem* (I) is equivalent to finding the corresponding g^* and then $u_{g^*}(x, y)$ in *Problem* (II). Notice that g^* is only defined along the interface, so it is one dimensional lower than $u(x, y)$. *Problem* (II) is an elliptic interface problem which is much easier to solve because the jump condition $[u_{\mathbf{n}}]$ is given instead of $[\beta u_{\mathbf{n}}]$. We can use the IIM idea to construct a second order scheme which also satisfies the maximum principle. The maximum principle guarantees the negative of the coefficient matrix of the finite difference scheme is an M-matrix that is diagonally dominant and invertible. Most iterative methods are guaranteed to converge for M-matrices.

We can write the finite difference scheme for *Problem* (II) in a general form at any grid point (x_i, y_j),

$$\sum_{k}^{n_s} \gamma_k U_{i+i_k, j+j_k} = \frac{f_{ij}}{\beta_{ij}} + C_{ij} \,. \tag{2.9}$$

n_s is the number of grid points involved in the finite difference stencil and U_{ij} is a discrete approximation to the exact solution. The sum over k involves several grid points near (x_i, y_j). So i_k, j_k takes value in the set $\{0, \pm 1, \pm 2, \ldots\}$.

To enforce the negative of the finite difference coefficient matrix to be an M-matrix, we impose the restrictions on the finite difference coefficients $\{\gamma_k\}$ in (2.9),

$$\begin{cases} \gamma_k \geq 0 & \text{if} \quad (i_k, j_k) \neq (0, 0) \\ \gamma_k < 0 & \text{if} \quad (i_k, j_k) = (0, 0) \end{cases} \tag{2.10}$$

In this dissertation, we focus on the case when β is variable and discontinuous as in (2.2). We develop a new finite difference scheme using five-point stencil to discretize the PDE in (2.5a)–(2.5c). Then we use the multigrid solver DMGD9V to solve for the solution u in *Problem* (II). We aim to develop a new numerical method with which we can obtain not only a second order accurate solution, but also a second order accurate

gradient. The key to success is to compute the augmented variable g^* accurately and efficiently. We describe our method to determine g^* in Chapter 3. Once g^* is found, we just need to apply the multigrid solver one more time to get the solution $u^*(x, y)$. Before we explain our method, we first provide some theoretical preparations by deriving the interface relations for *Problem* (II).

2.2 The interface relations

Note that the second jump condition (2.6b) in *Problem* (II) is different from that in (2.4b) of the original *Problem* (I). The interface relations from the original jump conditions (2.4a)–(2.4b) in *Problem* (I) has been derived in the book of Z. Li and K. Ito [14]. Here, we want to derive the interface relations from the new jump conditions (2.6a)–(2.6b).

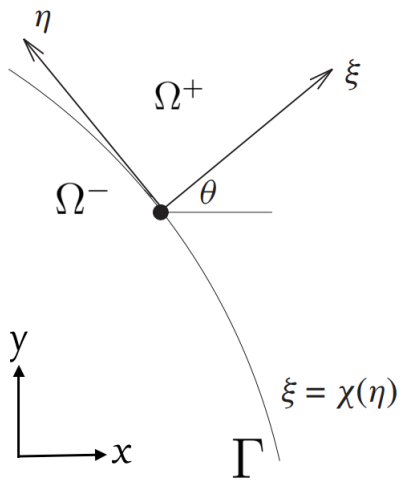

Figure 2.1: A diagram of the local coordinates in the normal and tangential directions, where θ is the angle between the x-axis and the normal direction.

Since the jump condition is given in the normal direction $\mathbf{n}$, it is more convenient to

use the local coordinates in the normal and tangential directions. Given a point (X, Y) on the interface, the local coordinate system in the normal and tangential directions is defined as (see Figure 2.1 for illustrations),

$$\begin{cases} \xi = (x - X)\cos\theta + (y - Y)\sin\theta \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta \end{cases} \tag{2.11}$$

where θ is the angle between the x-axis and the normal direction ($\hat{\xi} = \hat{\mathbf{n}}$), pointing to the Ω^+ subdomain. Under such new coordinates system, the interface can be parameterized by

$$\xi = \chi(\eta) \qquad \text{with} \qquad \chi(0) = 0,\ \chi'(0) = 0. \tag{2.12}$$

where we assume that the interface is twice continuously differentiable, and hence $\chi'(0) = 0$. The curvature of the interface at (X, Y) is $\chi''(0)$.

Consider the jumps across Γ at a fixed point (X, Y) that corresponds to the local coordinates $\xi = \eta = 0$. At this point, the solution of *Problem* (II) will satisfy the following interface relations that represent the limiting values from one side of the interface in terms of the other using the local coordinates (2.11).

Theorem 1. *Let (X, Y) be a point on the interface $\Gamma \in C^2$. Then, from the jump condi-*

tions (2.6a)–(2.6b) *and the PDE* (2.5a)–(2.5b)*, we have the following interface relations:*

$$
\begin{aligned}
[u] &= w, \\
[u_\xi] &= g, \\
[u_\eta] &= w', \\
[u_{\eta\eta}] &= -g\chi'' + w'', \\
[u_{\xi\eta}] &= w'\chi'' + g', \\
[u_{\xi\xi}] &= g\left(\chi'' - \frac{\beta_\xi^+}{\beta^+}\right) - w'' - \left[\frac{\beta_\xi}{\beta}\right] u_\xi^- - \left[\frac{\beta_\eta}{\beta}\right] u_\eta^- - \frac{\beta_\eta^+}{\beta^+} w' + \left[\frac{f}{\beta}\right], \\
&= g\left(\chi'' - \frac{\beta_\xi^-}{\beta^-}\right) - w'' - \left[\frac{\beta_\xi}{\beta}\right] u_\xi^+ - \left[\frac{\beta_\eta}{\beta}\right] u_\eta^+ - \frac{\beta_\eta^-}{\beta^-} w' + \left[\frac{f}{\beta}\right],
\end{aligned}
\tag{2.13}
$$

where w', g' and w'' are the first- and second-order surface derivatives of w and g on the interface, which are all evaluated at $(\xi, \eta) = (0, 0)$.

Proof: In a neighborhood of (X, Y), the interface can be expressed as $\xi = \chi(\eta)$ with $\chi(0) = 0$ and $\chi'(0) = 0$. Then the jump conditions w and g can be written as functions of only η. For simplicity, we still use the notation $[u] = w(\eta)$ and $[u_{\mathbf{n}}] = g(\eta)$ in the local coordinate system. Setting $\eta = 0$ in (2.6a) and (2.6b), we get the first two equalities in (2.13),

$$[u] = w(\eta) \implies [u] = w(0) = w, \tag{2.14}$$

$$[u_{\mathbf{n}}] = g(\eta) \implies [u_\xi] = g(0) = g. \tag{2.15}$$

Differentiating the first jump condition of *Problem* I in (2.6a) with respect to η along the interface, we get

$$[u_\xi]\chi' + [u_\eta] = w'(\eta). \tag{2.16}$$

Setting $\eta = 0$, we get the third equation in (2.13). Differentiating the equation above

again with respect to η, we obtain

$$[u_\xi]\chi'' + \chi' \frac{d}{d\eta}[u_\xi] + [u_{\xi\eta}]\chi' + [u_{\eta\eta}] = w''(\eta). \tag{2.17}$$

Setting $\eta = 0$, we get the fourth equality in (2.13). Notice that in the local coordinates, the second jump condition of *Problem* II in (2.6b) can be written as

$$(u_\xi^+ - u_\eta^+\chi') = (u_\xi^- - u_\eta^-\chi') + g\sqrt{1+\chi'^2}. \tag{2.18}$$

Differentiating (2.18) with respect to η along the interface, we have

$$\begin{aligned} u_{\xi\xi}^+\chi' + u_{\xi\eta}^+ - \frac{d}{d\eta}(u_\eta^+)\chi' - u_\eta^+\chi'' = {} & u_{\xi\xi}^-\chi' + u_{\xi\eta}^- - \frac{d}{d\eta}(u_\eta^-)\chi' - u_\eta^-\chi'' \\ & + g'(s)\sqrt{1+x'^2} + g(s)\frac{1}{2}\frac{2\chi'\chi''}{\sqrt{1+\chi'^2}}\,. \end{aligned} \tag{2.19}$$

Setting $\eta = 0$, we get the fifth equality in (2.13). Since the PDE in (2.5a)–(2.5b) are invariant under the transformation of the coordinates system in (2.11), we have

$$\begin{aligned} \left[u_{\xi\xi} + u_{\eta\eta} + \frac{\beta_\xi}{\beta}u_\xi + \frac{\beta_\eta}{\beta}u_\eta\right] = \left[\frac{f}{\beta}\right] \implies [u_{\xi\xi}] = {} & -(-g\chi'' + w'') - \left(\frac{\beta_\xi^+}{\beta^+}u_\xi^+ - \frac{\beta_\xi^-}{\beta^-}u_\xi^-\right) \\ & - \left(\frac{\beta_\eta^+}{\beta^+}u_\xi^+ - \frac{\beta_\eta^-}{\beta^-}u_\xi^-\right) - \left[\frac{f}{\beta}\right]. \end{aligned} \tag{2.20}$$

Substituting $u_\xi^+ = u_\xi^- + g$ and $u_\eta^+ = u_\eta^- + w'$ in the above equation, we get the first line of the last equality in (2.13). Similarly, if we substitute $u_\xi^- = u_\xi^+ - g$ and $u_\eta^- = u_\eta^+ - w'$ in the above equation, we obtain the second line of the last equation in (2.13). □

These interface relations derived here will be used to derive the finite difference method in the next section.

2.3 The augmented IIM

In Section 2.1, we introduced the jump in the normal derivative of the solution as the augmented variable and re-wrote the original interface problem as a Laplacian of the solution with lower order derivative terms. In Section 2.2, we derived jump relations for the solution, and its first order and second order derivatives in the local coordinates system. In this section, we use the IIM idea to discretize *Problem* (II) and derive the Schur complement system.

We first generate a uniform grid on the rectangular domain $[a, b] \times [c, d]$ where the elliptic interface problem is defined:

$$x_i = a + ih_x,\ y_j = c + jh_y, \qquad 0 \le i \le N,\ 0 \le j \le M \tag{2.21}$$

where $h_x = (b-a)/N$ and $h_y = (d-c)/M$. We assume that $h_x = h_y = h$ for simplicity.

We use the zero level set of a Liptschiz continuous function $\phi(x)$ defined on the entire domain $(\Omega^+ \cup \Omega^- \cup \Gamma)$ to represent the interface. For example, if the interface is the unit circle in two dimension, then the choice of a level set function is $\phi(x, y) = \sqrt{x^2 + y^2} - 1$. The entire domain is then divided into two disjoint parts $\Omega^- = \{(x, y), \phi(x, y) < 0\}$ and $\Omega^+ = \{(x, y), \phi(x, y) > 0\}$ by the interface $\Gamma = \{(x, y), \phi(x, y) = 0\}$.

To distinguish the discrete solution from the continuous solution, we use uppercase letters to indicate the solution of the discrete problem and lowercase letters for the continuous solutions. We also use bold letters for vectors.

We say (x_i, y_j) is a *regular grid point* if the interface does not come between any grid points in the standard five-point stencil centered at (i, j). At these points, we can obtain

an $\mathcal{O}(h^2)$ local truncation error using the standard 5-point formula

$$\begin{aligned}
&\frac{1}{h_x}\left(\beta_{i+\frac{1}{2},j}\frac{\big(U_{i+1,j}-U_{i,j}\big)}{h_x}-\beta_{i-\frac{1}{2},j}\frac{\big(U_{i,j}-U_{i-1,j}\big)}{h_x}\right)\\
&+\frac{1}{h_y}\left(\beta_{i,j+\frac{1}{2}}\frac{\big(U_{i,j+1}-U_{i,j}\big)}{h_y}-\beta_{i,j-\frac{1}{2}}\frac{\big(U_{i,j}-U_{i,j-1}\big)}{h_y}\right)=f_{i,j}\,,\\
\Longrightarrow\quad &\frac{1}{\beta_{\text{sum}}}\left(\beta_{i+\frac{1}{2},j}\frac{U_{i+1,j}}{h^2}+\beta_{i-\frac{1}{2},j}\frac{U_{i-1,j}}{h^2}\right.\\
&\qquad\left.+\beta_{i,j+\frac{1}{2}}\frac{U_{i,j+1}}{h^2}+\beta_{i,j-\frac{1}{2}}\frac{U_{i,j-1}}{h^2}\right)-\frac{U_{i,j}}{h^2}=\frac{f_{i,j}}{\beta_{\text{sum}}}\,,
\end{aligned}\tag{2.22}$$

where we have used the assumption $h_x = h_y = h$ and $\beta_{\text{sum}} = \beta_{i+1/2,j} + \beta_{i-1/2,j} + \beta_{i,j+1/2} + \beta_{i,j-1/2}$. Since $\beta(x,y) \geq \beta_{\min} > 0$, the finite difference coefficients in (2.22) satisfy the sign restrictions in (2.10).

We say (x_i, y_j) is an *irregular grid point* if the grid points in the standard five-point stencil centered at (i,j) are from both sides of the interface, as illustrated in Figure 2.2. We also wish to determine formula of the form (2.22) for the irregular points. Since these irregular points are adjacent to the interface Γ and form a lower dimensional set, it turns out to be sufficient to require an $\mathcal{O}(h)$ local truncation error at these points. To derive the finite difference scheme for the irregular points, we first write the PDE at the irregular point (x_i, y_j) as follows

$$u_{xx}(x_i,y_j)+u_{yy}(x_i,y_j)+\frac{\beta_x(x_i,y_j)}{\beta(x_i,y_j)}u_x(x_i,y_j)+\frac{\beta_y(x_i,y_j)}{\beta(x_i,y_j)}u_y(x_i,y_j)=\frac{f(x_i,y_j)}{\beta(x_i,y_j)}\tag{2.23}$$

Note that if (x_i, y_j) happens to be on the interface, then $\beta(x_i,y_j)$, $\beta_x(x_i,y_j)$, $\beta_y(x_i,y_j)$ and $f(x_i,y_j)$ are defined as the limiting value from a pre-chosen side of the interface., say the "$-$" side. The discrete forms for the operators $u_{xx}(x_i,y_j)$, $u_{yy}(x_i,y_j)$, $u_x(x_i,y_j)$

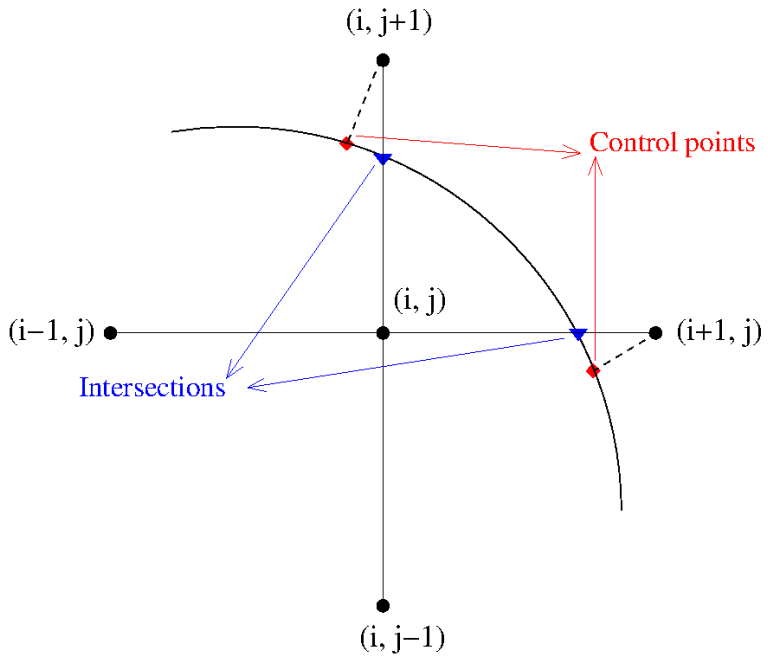

Figure 2.2: The geometry at an irregular grid point (x_i, y_j). The red diamonds are the control points, which are the orthogonal projections of the grid points (x_i, y_{j+1}) and (x_{i+1}, y_j) on the interface. The blue triangles are the intersection points, where the interface intersects with the grid lines involved in the 5-point stencil.

and $u_y(x_i, y_j)$ are discussed in detail below.

2.3.1 Discretization of u_{xx} and u_{yy} at the irregular grid points

Consider the irregular grid point (i, j), which could belong to either the Ω^- domain (Figure 2.3) or the Ω^+ domain (Figure 2.4). Here, we first discuss the case when (i, j) is in the Ω^- subdomain. In this case, at least one of its four nearest neighboring grid points ($(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$) must belong to the other subdomain Ω^+.

To discretize $u_{xx}(x_i, y_j)$, we need the two nearest neighboring points from the left side $(i-1, j)$ and the right side $(i+1, j)$. Their positions with respect to the interface could have three possibilities as listed below.

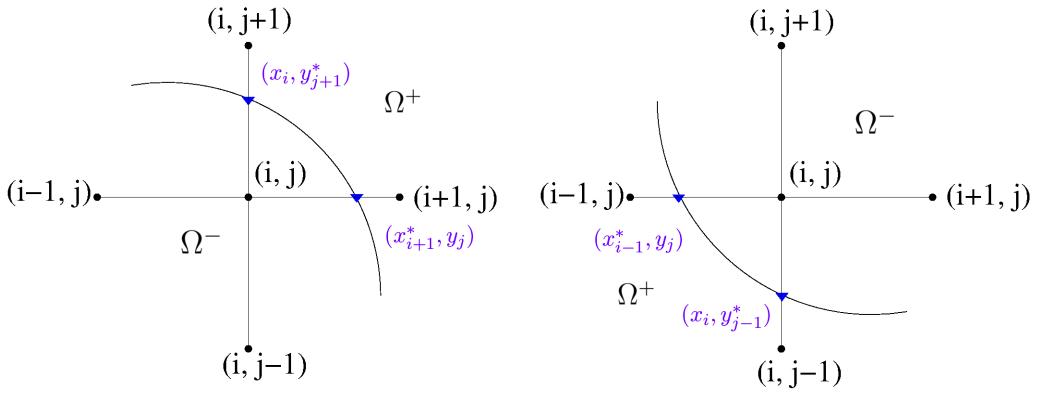

Figure 2.3: The irregular grid point (x_i, y_j) in the Ω^- subdomain. At least one of its four nearest neighbor grid points must belong to the other subdomain Ω^+. The intersections are labeled by the little blue triangles, with their coordinates listed inside the parentheses.

- When $(i, j), (i-1, j) \in \Omega^-$, while $(i+1, j) \in \Omega^+$ (see the left graph in Figure 2.3):

  Define an auxiliary function in the range $x \in [x_{i-1}, x_{i+1}], y = y_j$

$$\widetilde{u}(x) = \begin{cases} 0 & x_{i-1} \le x \le x^*_{i+1}, \\ [u]_R + [u_x]_R(x - x^*_{i+1}) + [u_{xx}]_R \dfrac{(x - x^*_{i+1})^2}{2} & x^*_{i+1} \le x \le x_{i+1}, \end{cases}$$

  where x^*_{i+1} is x-coordinate of the intersection point (Figure 2.3) and $[u]_R$, $[u_x]_R$ and $[u_{xx}]_R$ are the jumps at the intersection point (x^*_{i+1}, y_j) on the right hand side of (i, j). Then, we define another function $q(x) = u(x, y_j) - \widetilde{u}(x)$, which satisfies $[q] = [q_x] = [q_{xx}] = 0$. That says $q(x)$ is twice continuously differentiable. Since

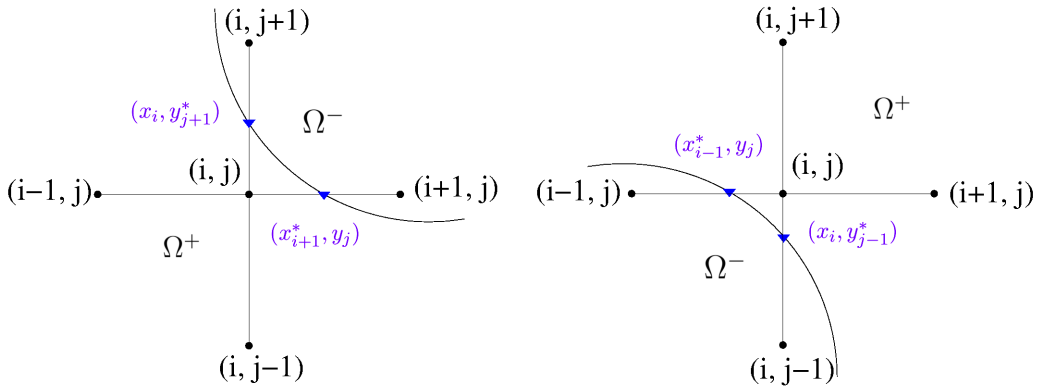

Figure 2.4: The irregular grid point (x_i, y_j) in Ω^+ subdomain. At least one of its four nearest neighbor grid points must belong to the other subdomain Ω^-. The intersections are labeled by the little blue triangles, with their coordinates listed inside the parentheses.

$u(x, y_j) = q(x)$ in the neighborhood of x_i, we have

$$\begin{aligned} u_{xx}(x_i, y_j) = q_{xx}(x_i) &= \frac{Q_{i+1} - 2Q_i + Q_{i-1}}{h_x^2} + \mathcal{O}(h_x^2) \\ &= \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} \\ &\quad - \frac{[u]_R + [u_x]_R d_R + [u_{xx}]_R \dfrac{d_R^2}{2}}{h_x^2} + \mathcal{O}(h_x^2)\,, \end{aligned} \tag{2.24}$$

where $d_R = x_{i+1} - x^*_{i+1}$.

- When $(i, j), (i+1, j) \in \Omega^-$, while $(i-1, j) \in \Omega^+$ (see the right graph in Figure 2.3):

  Define an auxiliary function in the range $x \in [x_{i-1}, x_{i+1}], y = y_j$

$$\widetilde{u}(x) = \begin{cases} [u]_L + [u_x]_L (x - x^*_{i-1}) + [u_{xx}]_L \dfrac{(x - x^*_{i-1})^2}{2} & x_{i-1} \le x \le x^*_{i-1}, \\ 0 & x^*_{i-1} \le x \le x_{i+1}, \end{cases}$$

  where x^*_{i-1} is x-coordinate of the intersection point (Figure 2.3) and $[u]_L$, $[u_x]_L$ and

$[u_{xx}]_L$ are the jump relations at the intersection point (x^*_{i-1}, y_j) on the left hand side of (i,j). Then, we define another function $q(x) = u(x,y_j) - \widetilde{u}(x)$, which satisfies $[q] = [q_x] = [q_{xx}] = 0$. That says $q(x)$ is twice continuously differentiable. Since $u(x,y_j) = q(x)$ in the neighborhood of x_i, we have

$$\begin{aligned} u_{xx}(x_i,y_j) = q_{xx}(x_i) &= \frac{Q_{i+1} - 2Q_i + Q_{i-1}}{h_x^2} + \mathcal{O}(h_x^2) \\ &= \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} \\ &\quad - \frac{[u]_L + [u_x]_L d_L + [u_{xx}]_L \dfrac{d_L^2}{2}}{h_x^2} + \mathcal{O}(h_x^2)\,, \end{aligned} \tag{2.25}$$

where $d_L = x_{i-1} - x^*_{i-1}$.

- When $(i,j) \in \Omega^-$, while $(i-1,j), (i+1,j) \in \Omega^+$:

  Define an auxiliary function in the range $x \in [x_{i-1}, x_{i+1}], y = y_j$

$$\widetilde{u}(x) = \begin{cases} [u]_L + [u_x]_L (x - x^*_{i-1}) + [u_{xx}]_L \dfrac{(x - x^*_{i-1})^2}{2} & x_{i-1} \le x \le x^*_{i-1}, \\ 0 & x^*_{i-1} \le x \le x^*_{i+1}, \\ [u]_R + [u_x]_R (x - x^*_{i+1}) + [u_{xx}]_R \dfrac{(x - x^*_{i+1})^2}{2} & x^*_{i+1} \le x \le x_{i+1}, \end{cases}$$

  where x^*_{i-1} and x^*_{i+1} are x-coordinates of the intersection points (Figure 2.3) and $[u]_{L/R}$, $[u_x]_{L/R}$ and $[u_{xx}]_{L/R}$ are the jumps at the intersection point $(x^*_{i-1}, y_j)/(x^*_{i+1}, y_j)$ on the left/right hand side of (i,j). Then, we define another function $q(x) = u(x,y_j) - \widetilde{u}(x)$, which satisfies $[q] = [q_x] = [q_{xx}] = 0$. That says $q(x)$ is twice continuously differentiable. Since $u(x,y_j) = q(x)$ in the neighborhood of x_i, we

have

$$
\begin{aligned}
u_{xx}(x_i, y_j) = q_{xx}(x_i) &= \frac{Q_{i+1} - 2Q_i + Q_{i-1}}{h_x^2} + \mathcal{O}(h_x^2) \\
&= \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} \\
&\quad - \frac{[u]_L + [u_x]_L d_L + [u_{xx}]_L \dfrac{d_L^2}{2}}{h_x^2} \\
&\quad - \frac{[u]_R + [u_x]_R d_R + [u_{xx}]_R \dfrac{d_R^2}{2}}{h_x^2} + \mathcal{O}(h_x^2)\,,
\end{aligned} \tag{2.26}
$$

where $d_L = x_{i-1} - x_{i-1}^*$ and $d_R = x_{i+1} - x_{i+1}^*$.

Similarly, to discretize $u_{yy}(x_i, y_j)$, we need the two nearest neighbor points from the up side $(i, j+1)$ and the down side $(i, j-1)$. Their positions with respect to the interface could also have three possibilities as listed below.

- When $(i, j), (i, j-1) \in \Omega^-$, while $(i, j+1) \in \Omega^+$ (see the left graph in Figure 2.3):

  Define an auxiliary function in the range $y \in [y_{j-1}, y_{j+1}], x = x_i$

$$
\widetilde{u}(y) = \begin{cases} 0 & y_{j-1} \le x \le y_{j+1}^*, \\ [u]_U + [u_y]_U (y - y_{j+1}^*) + [u_{yy}]_U \dfrac{(y - y_{j+1}^*)^2}{2} & y_{j+1}^* \le y \le y_{j+1}, \end{cases}
$$

  where y_{j+1}^* is y-coordinate of the intersection point (Figure 2.3) and $[u]_U$, $[u_y]_U$ and $[u_{yy}]_U$ are the jump relations at the intersection point (x_i, y_{j+1}^*) on the up side of (i, j). Then, we define another function $q(y) = u(x_i, y) - \widetilde{u}(y)$, which satisfies $[q] = [q_y] = [q_{yy}] = 0$. That says $q(y)$ is twice continuously differentiable. Since

$u(x_i, y) = q(y)$ in the neighborhood of y_j, we have

$$\begin{aligned} u_{yy}(x_i, y_j) = q_{yy}(y_j) &= \frac{Q_{j+1} - 2Q_j + Q_{j-1}}{h_y^2} + \mathcal{O}(h_y^2) \\ &= \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} \\ &\quad - \frac{[u]_U + [u_y]_U d_U + [u_{yy}]_U \dfrac{d_U^2}{2}}{h_y^2} + \mathcal{O}(h_y^2)\,, \end{aligned} \tag{2.27}$$

where $d_U = y_{j+1} - y_{j+1}^*$.

- When $(i, j), (i, j+1) \in \Omega^-$, while $(i, j-1) \in \Omega^+$ (see the right graph in Figure 2.3):

  Define an auxiliary function in the range $y \in [y_{j-1}, y_{j+1}], x = x_i$

$$\widetilde{u}(y) = \begin{cases} [u]_D + [u_y]_D (y - y_{j-1}^*) + [u_{yy}]_D \dfrac{(y - y_{j-1}^*)^2}{2} & y_{j-1} \le y \le y_{j-1}^*, \\ 0 & y_{j-1}^* \le y \le y_{j+1}, \end{cases}$$

  where y_{j-1}^* is y-coordinate of the intersection point (Figure 2.3) and $[u]_D$, $[u_y]_D$ and $[u_{yy}]_D$ are the jump relations at the intersection point (x_i, y_{j-1}^*) on the down side of (i, j). Then, we define another function $q(y) = u(x_i, y) - \widetilde{u}(y)$, which satisfies $[q] = [q_y] = [q_{yy}] = 0$. That says $q(y)$ is twice continuously differentiable. Since $u(x_i, y) = q(y)$ in the neighborhood of y_j, we have

$$\begin{aligned} u_{yy}(x_i, y_j) = q_{yy}(y_j) &= \frac{Q_{j+1} - 2Q_j + Q_{j-1}}{h_y^2} + \mathcal{O}(h_y^2) \\ &= \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} \\ &\quad - \frac{[u]_D + [u_y]_D d_D + [u_{yy}]_D \dfrac{d_D^2}{2}}{h_y^2} + \mathcal{O}(h_y^2)\,, \end{aligned} \tag{2.28}$$

where $d_D = y_{j-1} - y^*_{j-1}$.

- When $(i,j) \in \Omega^-$, while $(i,j-1),(i,j+1) \in \Omega^+$:

  Define an auxiliary function in the range $y \in [y_{j-1}, y_{j+1}], x = x_i$

$$\widetilde{u}(y) = \begin{cases} [u]_D + [u_y]_D(y - y^*_{j-1}) + [u_{yy}]_D \dfrac{(y - y^*_{j-1})^2}{2} & y_{j-1} \le y \le y^*_{j-1}, \\ 0 & y^*_{j-1} \le y \le y^*_{j+1}, \\ [u]_U + [u_y]_U(y - y^*_{j+1}) + [u_{yy}]_U \dfrac{(y - y^*_{j+1})^2}{2} & y^*_{j+1} \le y \le y_{j+1}, \end{cases}$$

  where y^*_{j-1} and y^*_{j+1} are y-coordinates of the intersection points (Figure 2.3) and $[u]_{D/U}$, $[u_y]_{D/U}$ and $[u_{yy}]_{D/U}$ are the jumps at the intersection point $(x_i, y^*_{j-1})/(x_i, y^*_{j+1})$ on the down/up side of (i,j). Then, we define another function $q(y) = u(x_i, y) - \widetilde{u}(y)$, which satisfies $[q] = [q_y] = [q_{yy}] = 0$. That says $q(y)$ is twice continuously differentiable. Since $u(x_i, y) = q(y)$ in the neighborhood of y_j, we have

$$\begin{aligned} u_{yy}(x_i, y_j) = q_{yy}(y_j) &= \frac{Q_{j+1} - 2Q_j + Q_{j-1}}{h_y^2} + \mathcal{O}(h_y^2) \\ &= \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} \\ &\quad - \frac{[u]_D + [u_y]_D d_D + [u_{yy}]_D \dfrac{d_D^2}{2}}{h_y^2} \\ &\quad - \frac{[u]_U + [u_y]_U d_U + [u_{yy}]_U \dfrac{d_U^2}{2}}{h_y^2} + \mathcal{O}(h_y^2)\,, \end{aligned} \tag{2.29}$$

  where $d_D = y_{j-1} - y^*_{j-1}$ and $d_U = y_{j+1} - y^*_{j+1}$.

So far, we have obtained the discretization scheme for the differential operators $u_{xx}(x_i, y_j)$ and $u_{yy}(x_i, y_j)$ at an irregular point in the Ω^- domain. If the irregular point

(x_i, y_j) lies in the Ω^+ domain, we can follow exactly the same procedures above to derive the corresponding discrete forms. Here, for the case when the irregular grid point $(i,j) \in \Omega^+$ (see Figure 2.4), the only difference we need to make is to define $q = u + \widetilde{u}$, instead of $q = u - \widetilde{u}$ as used in the above analysis for $(i,j) \in \Omega^-$. This thus will change the "$-$" before the correction terms in (2.24)–(2.29) to the "$+$" sign in the discrete forms. We summarize the results in the following two equations. The discretization of $u_{xx}(x_i, y_j)$ for $(i,j) \in \Omega^+$ is

$$u_{xx}(x_i, y_j) = \begin{cases} \nabla^2_{h_x} + \dfrac{[u]_R + [u_x]_R d_R + [u_{xx}]_R \dfrac{d_R^2}{2}}{h_x^2} + \mathcal{O}(h_x^2) & \text{only } (i+1,j) \in \Omega^-, \\ \nabla^2_{h_x} + \dfrac{[u]_L + [u_x]_L d_L + [u_{xx}]_L \dfrac{d_L^2}{2}}{h_x^2} + \mathcal{O}(h_x^2) & \text{only } (i-1,j) \in \Omega^-, \\ \nabla^2_{h_x} + \dfrac{[u]_R + [u_x]_R d_R + [u_{xx}]_R \dfrac{d_R^2}{2}}{h_x^2} & \\ \quad + \dfrac{[u]_L + [u_x]_L d_L + [u_{xx}]_L \dfrac{d_L^2}{2}}{h_x^2} + \mathcal{O}(h_x^2) & \text{both } (i \pm 1, j) \in \Omega^-, \end{cases} \tag{2.30}$$

where $\nabla^2_{h_x} = \dfrac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2}$ and $[u]_{L/R}$, $[u_x]_{L/R}$, $[u_{xx}]_{L/R}$, $d_{L/R}$ are previously

defined in (2.24)–(2.26). The discretization of $u_{yy}(x_i, y_j)$ for $(i, j) \in \Omega^+$ is

$$u_{yy}(x_i, y_j) = \begin{cases} \nabla^2_{h_y} + \dfrac{[u]_U + [u_y]_U d_U + [u_{yy}]_U \dfrac{d_U^2}{2}}{h_y^2} + \mathcal{O}(h_y^2) & \text{only } (i, j+1) \in \Omega^-, \\ \nabla^2_{h_y} + \dfrac{[u]_D + [u_y]_D d_D + [u_{yy}]_D \dfrac{d_D^2}{2}}{h_y^2} + \mathcal{O}(h_y^2) & \text{only } (i, j-1) \in \Omega^-, \\ \nabla^2_{h_y} + \dfrac{[u]_U + [u_y]_U d_U + [u_{yy}]_U \dfrac{d_U^2}{2}}{h_y^2} & \\ \quad + \dfrac{[u]_D + [u_y]_D d_D + [u_{yy}]_D \dfrac{d_D^2}{2}}{h_y^2} + \mathcal{O}(h_y^2) & \text{both } (i, j \pm 1) \in \Omega^-, \end{cases} \tag{2.31}$$

where $\nabla^2_{h_y} = \dfrac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2}$ and $[u]_{U/D}$, $[u_x]_{U/D}$, $[u_{xx}]_{U/D}$, $d_{U/D}$ are previously defined in (2.27)–(2.29).

2.3.2 Jumps at the intersections

In the discrete forms of $u_{xx}(x_i, y_j)$ and $u_{yy}(x_i, y_j)$ in (2.24)–(2.31), we need to evaluate the jumps $[u]$, $[u_x]$, $[u_{xx}]$, $[u_y]$ and $[u_{yy}]$ at the intersection grid points. From the transformation of coordinates system in (2.11), we can derive the relations between the differential

operators in the local coordinate system and Cartesian coordinate system, which satisfies

$$\begin{aligned}
\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix}, \\
\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} &= \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}, \\
\begin{pmatrix} \frac{\partial^2}{\partial x^2} \\ \frac{\partial^2}{\partial y^2} \end{pmatrix} &= \begin{pmatrix} \cos^2\theta & -2\sin\theta\cos\theta & \sin^2\theta \\ \sin^2\theta & 2\sin\theta\cos\theta & \cos^2\theta \end{pmatrix} \begin{pmatrix} \frac{\partial^2}{\partial \xi^2} \\ \frac{\partial^2}{\partial \xi \partial \eta} \\ \frac{\partial^2}{\partial \eta^2} \end{pmatrix},
\end{aligned} \tag{2.32}$$

where the differential operators could act on any functions, for example, the solution $u(x, y)$.

Here we first consider the case where the center grid point $(i, j) \in \Omega^-$. At a specified intersection point (X_I, Y_J), from the interface relations in (2.13) and the differential

operator relations in (2.32), we have

$$
\begin{aligned}
[u] &= w, \\
[u_x] &= \cos\theta[u_\xi] - \sin\theta[u_\eta] = g\cos\theta - w'\sin\theta\,, \\
[u_y] &= \sin\theta[u_\xi] + \cos\theta[u_\eta] = g\sin\theta + w'\cos\theta\,, \\
[u_{xx}] &= \cos^2\theta[u_{\xi\xi}] - 2\sin\theta\cos\theta[u_{\xi\eta}] + \sin^2\theta[u_{\eta\eta}] \\
&= -2\sin\theta\cos\theta(w'\chi'' + g') + \sin^2\theta(-g\chi'' + w'') \\
&\quad + \cos^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - w'' - \frac{\beta_\eta^+}{\beta^+}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^- + \sin\theta u_y^-) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^- + \cos\theta u_y^-)\Big\}\,, \\
[u_{yy}] &= \sin^2\theta[u_{\xi\xi}] + 2\sin\theta\cos\theta[u_{\xi\eta}] + \cos^2\theta[u_{\eta\eta}] \\
&= 2\sin\theta\cos\theta(w'\chi'' + g') + \cos^2\theta(-g\chi'' + w'') \\
&\quad + \sin^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - w'' - \frac{\beta_\eta^+}{\beta^+}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^- + \sin\theta u_y^-) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^- + \cos\theta u_y^-)\Big\}\,,
\end{aligned}
\tag{2.33}
$$

where u_x^- and u_y^- can be further approximated by $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$ at the center grid point (i, j) using Taylor expansion. That says, since the distance between the intersection point and the center point (i, j) is less than or equal to h_x or h_y (see Figure 2.3), we can write

$$
\begin{aligned}
u_x^-|_{\text{intersection}} &= u_x(x_i, y_j) + \mathcal{O}(h_x), \\
u_y^-|_{\text{intersection}} &= u_y(x_i, y_j) + \mathcal{O}(h_y)\,.
\end{aligned}
\tag{2.34}
$$

Similarly, in the case $(i, j) \in \Omega^+$, we need to choose the alternative formula for the jump $[u_{\xi\xi}]$, see the second line of the last equality in (2.13). So the jumps $[u_{xx}]$, $[u_{yy}]$

become

$$
\begin{aligned}
[u_{xx}] &= \cos^2\theta[u_{\xi\xi}] - 2\sin\theta\cos\theta[u_{\xi\eta}] + \sin^2\theta[u_{\eta\eta}] \\
&= -2\sin\theta\cos\theta(w'\chi'' + g') + \sin^2\theta(-g\chi'' + w'') \\
&\quad + \cos^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^-}{\beta^-}) - w'' - \frac{\beta_\eta^-}{\beta^-}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^+ + \sin\theta u_y^+) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^+ + \cos\theta u_y^+)\Big\}, \\
[u_{yy}] &= \sin^2\theta[u_{\xi\xi}] + 2\sin\theta\cos\theta[u_{\xi\eta}] + \cos^2\theta[u_{\eta\eta}] \\
&= 2\sin\theta\cos\theta(w'\chi'' + g') + \cos^2\theta(-g\chi'' + w'') \\
&\quad + \sin^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^-}{\beta^-}) - w'' - \frac{\beta_\eta^-}{\beta^-}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^+ + \sin\theta u_y^+) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^+ + \cos\theta u_y^+)\Big\},
\end{aligned}
\tag{2.35}
$$

where u_x^+ and u_y^+ can be further approximated by $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$ at the center grid point (i, j) using Taylor expansion. Since the distance between the intersection point and the center point (i, j) is less than or equal to h_x or h_y (see Figure 2.4), we can write

$$
\begin{aligned}
u_x^+|_{\text{intersection}} &= u_x(x_i, y_j) + \mathcal{O}(h_x), \\
u_y^+|_{\text{intersection}} &= u_y(x_i, y_j) + \mathcal{O}(h_y)\,.
\end{aligned}
\tag{2.36}
$$

Substituting (2.33) and (2.35) to the discrete forms of u_{xx} and u_{yy} in (2.24)–(2.29), and collecting terms $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$, we have

$$
\begin{aligned}
u_{xx}(x_i, y_j) + u_{yy}(x_i, y_j) = \nabla_h^2 u(x_i, y_j) + k_1 u_x(x_i, y_j) + k_2 u_y(x_i, y_j) \\
+ C_1 + \mathcal{O}(\max\{h_x, h_y\})\,,
\end{aligned}
\tag{2.37}
$$

where $\nabla_h^2 u(x_i, y_j) = \dfrac{U_{i+1,j} + U_{i-1,j} - 2U_{i,j}}{h_x^2} + \dfrac{U_{i,j+1} + U_{i,j-1} - 2U_{i,j}}{h_y^2}$ and k_1, k_2, C_1 are constant correction terms. Notice that k_1 and k_2 are independent of the jump conditions g, w and their surface derivatives g', w' and w'' at the intersections (see (2.33) and (2.35) for details), while the constant C_1 is a function of g, w and their surface derivatives at the intersection points. Therefore, the PDE at the irregular grid point (x_i, y_j) in (2.23) becomes

$$\begin{aligned} \nabla_h^2 u(x_i, y_j) + \Big(\frac{\beta_x(x_i, y_j)}{\beta(x_i, y_j)} + k_1\Big) u_x(x_i, y_j) \\ + \Big(\frac{\beta_y(x_i, y_j)}{\beta(x_i, y_j)} + k_2\Big) u_y(x_i, y_j) \;=\; \frac{f(x_i, y_j)}{\beta(x_i, y_j)} - C_1 + \mathcal{O}(\max\{h_x, h_y\}) \;, \end{aligned} \tag{2.38}$$

where $\nabla_h^2 u(x_i, y_j) = \dfrac{U_{i+1,j} + U_{i-1,j} - 2U_{i,j}}{h_x^2} + \dfrac{U_{i,j+1} + U_{i,j-1} - 2U_{i,j}}{h_y^2}$ is the discrete Laplacian operator. We still need to derive the discrete forms for $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$.

2.3.3 Discretization of u_x and u_y at the irregular grid points

We use the two point stencil for the first order derivatives $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$. We first consider the discretization of $u_x(x_i, y_j)$. If the coefficient $(\frac{\beta_x}{\beta} + k_1)$ before $u_x(x_i, y_j)$ in (2.38) is positive , we use (i, j) and the grid point on the right $(i+1, j)$ to approximate $u_x(x_i, y_j)$. Otherwise, we use (i, j) and the grid point on the left $(i-1, j)$ to approximate $u_x(x_i, y_j)$. The stencil points are chosen in this way to satisfy the sign restrictions of the finite difference coefficients in (2.10). So, we have

$$u_x(x_i, y_j) = \begin{cases} \frac{U_{i+1,j} - U_{i,j}}{h_x} + C_R + \mathcal{O}(h_x) & \text{if } (\frac{\beta_x}{\beta} + k_1) > 0 \,, \\ \frac{U_{i,j} - U_{i-1,j}}{h_x} + C_L + \mathcal{O}(h_x) & \text{if } (\frac{\beta_x}{\beta} + k_1) \le 0 \,, \end{cases} \tag{2.39}$$

where both C_R and C_L are constants that are determined by the positions of the two point stencil relative to the interface Γ . $C_R = 0$ when (i, j) and $(i + 1, j)$ are from the same side of the interface. Similar, $C_L = 0$ if (i, j) and $(i - 1, j)$ are from the same side of the interface.

However, when the two grid points of the stencil are on different sides of the interface, we need to derive the nonzero constants C_R and C_L.

For C_R, the two grid points involved are (i, j) and $(i + 1, j)$, and their positions with respect to the interface could have two possibilities as listed below.

- When $(i, j) \in \Omega^-$, while $(i+1, j) \in \Omega^+$ (see the left graph in Figure 2.3): We define an auxiliary function in the range $x \in [x_{i-1}, x_{i+1}], y = y_j$

$$\widetilde{u}(x) = \begin{cases} 0 & x_{i-1} \le x \le x^*_{i+1}, \\ [u]_R + [u_x]_R(x - x^*_{i+1}) & x^*_{i+1} \le x \le x_{i+1}, \end{cases} \tag{2.40}$$

  where x^*_{i+1} is x-coordinate of the intersection point (Figure 2.3) and $[u]_R$ and $[u_x]_R$ are the jumps at the intersection point (x^*_{i+1}, y_j) on the right hand side of (i, j). Then, we define another function $q(x) = u(x, y_j) - \widetilde{u}(x)$, which satisfies $[q] = [q_x] = 0$. That says $q(x)$ is continuously differentiable. Since $u(x, y_j) = q(x)$ in the neighborhood of x_i, we have

$$\begin{aligned} u_x(x_i, y_j) = q_x(x_i) &= \frac{Q_{i+1} - Q_i}{h_x} + \mathcal{O}(h_x) \\ &= \frac{U_{i+1,j} - U_{i,j}}{h_x} - \frac{[u]_R + [u_x]_R d_R}{h_x} + \mathcal{O}(h_x) \\ \Longrightarrow C_R &= -\frac{[u]_R + [u_x]_R d_R}{h_x} \end{aligned} \tag{2.41}$$

  where $d_R = x_{i+1} - x^*_{i+1}$.

- When $(i,j) \in \Omega^+$, while $(i+1,j) \in \Omega^-$ (see the left graph in Figure 2.4): We define the same auxiliary function $\widetilde{u}(x)$ in (2.40). Then, we define another function $q(x) = u(x,y_j) + \widetilde{u}(x)$. Notice here we use "+" instead of "−" sign between $u(x,y_j)$ and $\widetilde{u}(x)$ since (i,j) is in the Ω^+ domain now. The $q(x)$ function satisfies $[q] = [q_x] = 0$. That says $q(x)$ is continuously differentiable. Since $u(x,y_j) = q(x)$ in the neighborhood of x_i, we have

$$\begin{aligned} u_x(x_i,y_j) = q_x(x_i) &= \frac{Q_{i+1} - Q_i}{h_x} + \mathcal{O}(h_x) \\ &= \frac{U_{i+1,j} - U_{i,j}}{h_x} + \frac{[u]_R + [u_x]_R d_R}{h_x} + \mathcal{O}(h_x) \\ \Longrightarrow C_R &= \frac{[u]_R + [u_x]_R d_R}{h_x} \end{aligned} \tag{2.42}$$

  where $d_R = x_{i+1} - x^*_{i+1}$.

For C_L, the two grid points involved are (i,j) and $(i-1,j)$, and their positions with respect to the interface could also have two possibilities as listed below.

- When $(i,j) \in \Omega^-$, while $(i-1,j) \in \Omega^+$ (see the right graph in Figure 2.3): We define an auxiliary function in the range $x \in [x_{i-1}, x_{i+1}], y = y_j$

$$\widetilde{u}(x) = \begin{cases} [u]_L + [u_x]_L (x - x^*_{i-1}) & x_{i-1} \le x \le x^*_{i-1}, \\ 0 & x^*_{i-1} \le x \le x_{i+1}, \end{cases} \tag{2.43}$$

  where x^*_{i-1} is x-coordinate of the intersection point (Figure 2.3) and $[u]_L$ and $[u_x]_L$ are the jump relations at the intersection point (x^*_{i-1}, y_j) on the left hand side of (i,j). Then, we define another function $q(x) = u(x,y_j) - \widetilde{u}(x)$, which satisfies $[q] = [q_x] = 0$. That says $q(x)$ is continuously differentiable. Since $u(x,y_j) = q(x)$

in the neighborhood of x_i, we have

$$\begin{aligned} u_x(x_i, y_j) = q_x(x_i) &= \frac{Q_i - Q_{i-1}}{h_x} + \mathcal{O}(h_x) \\ &= \frac{U_{i,j} - U_{i-1,j}}{h_x} - \frac{[u]_L + [u_x]_L d_L}{h_x} + \mathcal{O}(h_x) \\ \Longrightarrow C_L &= -\frac{[u]_L + [u_x]_L d_L}{h_x} \end{aligned} \tag{2.44}$$

where $d_L = x_{i-1} - x^*_{i-1}$.

- When $(i, j) \in \Omega^+$, while $(i-1, j) \in \Omega^-$ (see the right graph in Figure 2.4): We define the same auxiliary function $\widetilde{u}(x)$ in (2.43). Then, we define another function $q(x) = u(x, y_j) + \widetilde{u}(x)$. Notice here we use "+" instead of "−" sign between $u(x, y_j)$ and $\widetilde{u}(x)$ since (i, j) is in the Ω^+ domain now. The $q(x)$ function satisfies $[q] = [q_x] = 0$. That says $q(x)$ is continuously differentiable. Since $u(x, y_j) = q(x)$ in the neighborhood of x_i, we have

$$\begin{aligned} u_x(x_i, y_j) = q_x(x_i) &= \frac{Q_i - Q_{i-1}}{h_x} + \mathcal{O}(h_x) \\ &= \frac{U_{i,j} - U_{i-1,j}}{h_x} + \frac{[u]_L + [u_x]_L d_L}{h_x} + \mathcal{O}(h_x) \\ \Longrightarrow C_L &= \frac{[u]_L + [u_x]_L d_L}{h_x} \end{aligned} \tag{2.45}$$

where $d_L = x_{i-1} - x^*_{i+1}$.

Next, to derive the discrete form for $u_y(x_i, y_j)$, we use (i, j) and the grid point on the up side $(i, j+1)$ to approximate $u_y(x_i, y_j)$ if the coefficient $(\frac{\beta_y}{\beta} + k_2)$ before $u_y(x_i, y_j)$ in (2.38) is positive. Otherwise, we use (i, j) and the grid point on the down side $(i, j-1)$ to approximate $u_y(x_i, y_j)$. Note that, the stencil points are chosen in this way to satisfy

the sign restrictions of the finite difference coefficients in (2.10). So, we have

$$u_y(x_i, y_j) = \begin{cases} \dfrac{U_{i,j+1} - U_{i,j}}{h_y} + C_U + \mathcal{O}(h_y) & \text{if } (\dfrac{\beta_y}{\beta} + k_2) > 0\,, \\ \dfrac{U_{i,j} - U_{i,j-1}}{h_y} + C_D + \mathcal{O}(h_y) & \text{if } (\dfrac{\beta_y}{\beta} + k_2) \leq 0\,, \end{cases} \tag{2.46}$$

where both C_U and C_D are constants that are determined by the positions of the two point stencil relative to the interface Γ . $C_U = 0$ when (i, j) and $(i, j + 1)$ are from the same side of the interface. Similar, $C_D = 0$ if (i, j) and $(i, j - 1)$ are from the same side of the interface.

However, when the two grid points of the stencil are on different sides of the interface, we need to derive the nonzero constants C_U and C_D. For C_U, the two grid points involved are (i, j) and $(i, j + 1)$, and their positions with respect to the interface could have two possibilities. We then follow the previous procedure in deriving C_L and C_R to obtain C_U and C_D. The idea is to find a function q that is continuously differentiable and equals $u(x_i, y_j)$ in the neighborhood of the irregular grid point (i, j). Here, we skip the derivations and summarize the results below

$$C_U = \begin{cases} -\dfrac{[u]_U + [u_y]_U d_U}{h_y} & \text{if } (i,j) \in \Omega^-, (i,j+1) \in \Omega^+\,, \\ \dfrac{[u]_U + [u_y]_U d_U}{h_y} & \text{if } (i,j) \in \Omega^+, (i,j+1) \in \Omega^-\,, \end{cases} \tag{2.47}$$

where $d_U = y_{j+1} - y^*_{j+1}$ and $[u]_U$, $[u_y]_U$ are jumps at the intersection point (x_i, y^*_{j+1}).

$$C_D = \begin{cases} -\dfrac{[u]_D + [u_y]_D d_D}{h_y} & \text{if } (i,j) \in \Omega^-, (i,j-1) \in \Omega^+\,, \\ \dfrac{[u]_D + [u_y]_D d_D}{h_y} & \text{if } (i,j) \in \Omega^+, (i,j-1) \in \Omega^-\,, \end{cases} \tag{2.48}$$

where $d_D = y_{j-1} - y^*_{j-1}$ and $[u]_D$, $[u_y]_D$ are jumps at the intersection point (x_i, y^*_{j-1}).

Now, if we substitute the discrete forms of $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$ in (2.39) and (2.46) to the PDE in (2.38), we get a discretization scheme of the PDE at the irregular grid (x_i, y_j). Combining with the discretized form at the regular points in (2.22), we can write a general matrix vector form of the PDE for all grid points as

$$A\mathbf{U} + \mathcal{B}(\mathbf{W}, \mathbf{G}) = \mathbf{F} \tag{2.49}$$

where $\mathbf{U}$ is the vector formed by $\{U_{ij}\}$ and $\mathbf{F}$ is the vector formed by $\{\frac{f_{ij}}{\beta_{\text{sum}}}\}$ at regular grid points and $\{\frac{f_{ij}}{\beta_{ij}}\}$ at irregular grid points.

$$\mathbf{W} = [W_1, W_2, \ldots, W_{N_b}]^T$$

and

$$\mathbf{G} = [G_1, G_2, \ldots, G_{N_b}]^T$$

are the discrete values of the jump conditions (2.6a) and (2.6b) at a set of control points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_b}$ on the interface. These control points are the orthogonal projections of the irregular grid points onto the interface Γ, see Figure 2.2 for an illustration. $\mathcal{B}(\mathbf{W}, \mathbf{G})$ is a mapping from $\mathbf{W} = [W_1, W_2, \ldots, W_{N_b}]^T$ and $\mathbf{G} = [G_1, G_2, \ldots, G_{N_b}]^T$ to the constant correction terms C_1 in (2.38) and $C_{L/R/U/D}$ in (2.39) and (2.46). We know that $\mathcal{B}(\mathbf{W}, \mathbf{G})$ depends on $w(s)$ and $g(s)$ at the intersection points, which can be interpolated from the values of $w(s)$ and $g(s)$ at the control points. Moreover, $\mathcal{B}(\mathbf{W}, \mathbf{G})$ also depends on the first and second derivatives of $w(s)$, and the first derivative of $g(s)$ at the intersection points, where the differentiation is carried along the interface. At this stage we do not know whether such an interpolation and differentiation are linear or not. However, in

the discrete case, all the interpolations are linear combinations of the values on those control points. Moreover, all the derivatives at the intersections are also obtained by linear combinations of the values at these intersection points. Therefore, $\mathcal{B}(\mathbf{W}, \mathbf{G})$ is indeed a linear function of $\mathbf{W}$ and $\mathbf{G}$, and can be written as

$$\mathcal{B}(\mathbf{W}, \mathbf{G}) = B\mathbf{G} - B_1\mathbf{W}, \tag{2.50}$$

where B and B_1 are two matrices with real entries. Thus, (2.49) becomes

$$\begin{aligned} A\mathbf{U} + B\mathbf{G} &= \mathbf{F} + B_1\mathbf{W} \\ &\stackrel{\text{def}}{=} \bar{\mathbf{F}}\,. \end{aligned} \tag{2.51}$$

Note that A is the coefficient matrix resulting from the finite difference equations for all grid points, which strictly satisfies the sign restriction in (2.10). Thus $-A$ is an M-matrix, and is diagonally dominant and invertible. So for any given $\mathbf{G}$, there exists a unique solution $\mathbf{U}$ from (2.51).

2.3.4 The Schur complement system

The solution $\mathbf{U}$ of the equation above certainly depends on $\mathbf{G}$ and we are interested in finding $\mathbf{G}^*$ which satisfies the discrete form of the second jump condition of *Problem* I in (2.4b)

$$[\beta \odot \mathbf{U}_n](\mathbf{G}^*) - \mathbf{V} = \beta^+ \odot \mathbf{U}_n^+(\mathbf{G}^*) - \beta^- \odot \mathbf{U}_n^-(\mathbf{G}^*) - \mathbf{V} \equiv 0\,, \tag{2.52}$$

where the components of the vectors $\mathbf{U}_n^+$ and $\mathbf{U}_n^-$ are discrete approximations of the normal derivative at control points from each side of the interface. Note the $\odot$ notation

means element-wise multiplication of the two vectors. We use element-wise multiplication since $\beta^{\pm}$ may differ among the control points. In the next chapter, we will discuss how to use the known jump $\mathbf{G}$, to interpolate $\mathbf{U}_{ij}$ to get $\mathbf{U}_n^+$ and $\mathbf{U}_n^-$ in detail. As we will see in the next chapter, $\mathbf{U}_n^+$ and $\mathbf{U}_n^-$ depend on $\mathbf{U}$, $\mathbf{G}$ and $\mathbf{V}$ linearly, so we have

$$\begin{aligned} \beta^+ \odot \mathbf{U}_n^+ - \beta^- \odot \mathbf{U}_n^- - \mathbf{V} &= E\mathbf{U} + D\mathbf{G} + \bar{P}\mathbf{V} - \mathbf{V} \\ &= E\mathbf{U} + D\mathbf{G} - P\mathbf{V}\,, \end{aligned} \tag{2.53}$$

where E, D and $\bar{P}$ are some matrices and $P = I - \bar{P}$. Combining (2.51) and (2.53) to obtain the linear system of equations for $\mathbf{U}$ and $\mathbf{G}$:

$$\begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{F}} \\ P\mathbf{V} \end{bmatrix} \tag{2.54}$$

The solution $\mathbf{U}$ and $\mathbf{G}$ are the discrete forms of $u_{g^*}(x, y)$ and g^*, which are the solutions of *Problem* (II). They also satisfy the flux jump condition in (2.8).

The next question is how to solve (2.54) efficiently. The multigrid approach and the GMRES method are two attractive choices. We have decided to solve for $\mathbf{G}$ in (2.54) first, and then to find the solution $\mathbf{U}$ by using the multigrid solver. Eliminating $\mathbf{U}$ from (2.54) gives a linear system for $\mathbf{G}$

$$\begin{aligned} (D - EA^{-1}B)\mathbf{G} &= P\mathbf{V} - EA^{-1}\bar{\mathbf{F}} \\ &\stackrel{\text{def}}{=} \bar{\mathbf{V}}\,. \end{aligned} \tag{2.55}$$

This is an $N_b \times N_b$ system for $\mathbf{G}$, a much smaller linear system compared to the one for $\mathbf{U}$. Since we cannot guarantee the coefficient matrix of this Schur complement to be symmetric positive definite, the GMRES iterative method is preferred. In practice, the

matrices A, B, E, D, P, and the vectors $\bar{\mathbf{V}}$, $\bar{\mathbf{F}}$ are used only for theoretical purpose, and are not constructed explicitly in practice. Thus the iterative method such as GMRES is preferred, since it requires only matrix-vector multiplication. The way we compute the left-hand side of (2.53) will give different E and D, and would affect the condition number of (2.55) substantially. We will address these aspects in the next Chapter.

Chapter 3

Development of the Fast Algorithm

3.1 Generalized weighted least squares interpolation scheme

When we apply the GMRES method to solve the Schur complement system of (2.55) for the augmented variable $\mathbf{G}^*$, we need to compute the matrix-vector multiplication on the left hand side, which is equivalent to computing $\mathbf{U}_n^+$ and $\mathbf{U}_n^-$ with the knowledge of $\mathbf{U}_{ij}$. This turns out to be a crucial step that could affect the accuracy and efficiency of the augmented method significantly. Our approach is based on a weighted least squares formulation. The idea described here can also be, and has been, applied to the case where we want to approximate some quantities on the interface from a grid function. For example, interpolating $\mathbf{U}_{ij}$ to the interface to get $\mathbf{U}^+(X,Y)$ or $\mathbf{U}^-(X,Y)$, where (X,Y) are the control points on the interface.

Previously, Z. Li has developed a weighted least square interpolation scheme for interface problems with piecewise constant coefficients. Here, we want to derive an interpola-

tion scheme that works for the more generalized case where the coefficients β is piecewise variable.

Our interpolation scheme for approximating $\mathbf{U}_{\mathbf{n}}^{-}$ can be written as

$$\frac{\partial \mathbf{U}^{-}}{\partial \mathbf{n}}(\mathbf{X}) = \sum_{k=0}^{k_s-1} \gamma_k U_{i+i_k, j+j_k} - Q\,, \tag{3.1}$$

where k_s is the number of grid points involved in the interpolation scheme, (x_i, y_j) is the irregular grid point whose orthogonal projection onto the interface generates the control point $\mathbf{X} = (X, Y)$, and Q is a correction term. Below we discuss how to determine the interpolation coefficients $\{\gamma_k\}$ and the correction term Q using the information from both sides of the interface. Note that $\{\gamma_k\}$ and Q depend on the position of the control point $\mathbf{X}$, but for simplicity of notation, we omit the dependency.

The interpolation coefficients $\{\gamma_k\}$ are determined by minimizing the interpolation error of (3.1) when $U_{i+i_k, j+j_k}$ is substituted with the exact solution $u(x_{i+i_k}, y_{j+j_k})$. Using the local coordinates system (2.11) centered at the control point $\mathbf{X}$, and denoting the local coordinates of (x_{i+i_k}, y_{j+j_k}) as (ξ_k, η_k), we have the following form from the Taylor expansion at $\mathbf{X} = (X, Y)$ or $(0, 0)$ in the local coordinates:

$$\begin{aligned} u(x_{i+i_k}, y_{j+j_k}) = u(\xi_k, \eta_k) \;=\; & u^{\pm} + \xi_k u_{\xi}^{\pm} + \eta u_{\eta}^{\pm} \\ & + \frac{1}{2}\xi_k^2 u_{\xi\xi}^{\pm} + \xi_k \eta_k u_{\xi\eta}^{\pm} + \frac{1}{2}\eta_k^2 u_{\eta\eta}^{\pm} + \mathcal{O}(h^3) \end{aligned} \tag{3.2}$$

where the "+" or "−" sign is chosen depending on whether (ξ_k, η_k) lies on the Ω^+ subdomain or the Ω^- subdomain, and $u^{\pm}, u_{\xi}^{\pm}, \ldots, u_{\eta\eta}^{\pm}$ are evaluated at the local coordinates $(0, 0)$ or $\mathbf{X} = (X, Y)$ in the original Cartesian coordinates system. Note that we should have used something like $u(X, Y) = \bar{u}(0, 0)$ to distinguish the two coordinate systems;

however, we omit the bars and use the same notation $u(X,Y) = u(0,0)$ for simplicity.

We carry out this expansion for all the grid points involved in the interpolation scheme and plug (3.2) into (3.1). After collecting and arranging terms, we can write

$$\begin{aligned}\frac{\partial u^-}{\partial \mathbf{n}}(\mathbf{X}) \;\approx\; & a_1 u^- + a_2 u^+ + a_3 u^-_\xi + a_4 u^+_\xi + a_5 u^-_\eta + a_6 u^+_\eta \\ & + a_7 u^-_{\xi\xi} + a_8 u^+_{\xi\xi} + a_9 u^-_{\eta\eta} + a_{10} u^+_{\eta\eta} + a_{11} u^-_{\xi\eta} + a_{12} u^+_{\xi\eta} \\ & - Q + \mathcal{O}(h^3 \max|\gamma_{ij}|)\,,\end{aligned} \tag{3.3}$$

where the $\{a_i\}$'s are given by

$$\begin{aligned} a_1 &= \sum_{k\in\Omega^-} \gamma_k, & a_2 &= \sum_{k\in\Omega^+} \gamma_k, \\ a_3 &= \sum_{k\in\Omega^-} \xi_k\gamma_k, & a_4 &= \sum_{k\in\Omega^+} \xi_k\gamma_k, \\ a_5 &= \sum_{k\in\Omega^-} \eta_k\gamma_k, & a_6 &= \sum_{k\in\Omega^+} \eta_k\gamma_k, \\ a_7 &= \tfrac{1}{2}\sum_{k\in\Omega^-} \xi_k^2\gamma_k, & a_8 &= \tfrac{1}{2}\sum_{k\in\Omega^+} \xi_k^2\gamma_k, \\ a_9 &= \tfrac{1}{2}\sum_{k\in\Omega^-} \eta_k^2\gamma_k, & a_{10} &= \tfrac{1}{2}\sum_{k\in\Omega^+} \eta_k^2\gamma_k, \\ a_{11} &= \sum_{k\in\Omega^-} \eta_k\xi_k\gamma_k, & a_{12} &= \sum_{k\in\Omega^+} \xi_k\eta_k\gamma_k. \end{aligned} \tag{3.4}$$

From *Theorem* 1, we have the interface relations in (2.13). Therefore, we can express all the quantities from the "+" side in (3.3) in terms of those from the "−" side. Thus,

we have

$$\begin{aligned}\frac{\partial u^-}{\partial \mathbf{n}}(\mathbf{X}) \;\approx\; & (a_1+a_2)u^- + \Big(a_3+a_4-a_8\Big[\frac{\beta_\xi}{\beta}\Big]\Big)u^-_\xi \\ & + \Big(a_5+a_6-a_8\Big[\frac{\beta_\eta}{\beta}\Big]\Big)u^-_\eta + (a_7+a_8)u^-_{\xi\xi} + (a_9+a_{10})u^-_{\eta\eta} \\ & + (a_{11}+a_{12})u^-_{\xi\eta} + \Big\{a_2 w + a_4 g + a_6 w' \\ & + a_8\Big(g(\chi'' - \frac{\beta^+_\xi}{\beta^+}) - \frac{\beta^+_\eta}{\beta^+}w' - w'' + \Big[\frac{f}{\beta}\Big]\Big) + a_{10}(-g\chi'' + w'') \\ & + a_{12}(w'\chi'' + g')\Big\} \\ & - Q + \mathcal{O}(h^3 \max|\gamma_{ij}|)\,.\end{aligned} \tag{3.5}$$

To minimize the interpolation error, we should set the following linear system of equations for the coefficients $\{\gamma_k\}$ by matching the terms of $u^-, u^-_\xi, \ldots, u^-_{\xi\eta}$. Since $u^-_{\mathbf{n}} = u^-_\xi$, we have

$$\begin{aligned} a_1 + a_2 &= 0, \\ a_3 + a_4 - a_8\Big[\frac{\beta_\xi}{\beta}\Big] &= 1, \\ a_5 + a_6 - a_8\Big[\frac{\beta_\eta}{\beta}\Big] &= 0, \\ a_7 + a_8 &= 0, \\ a_9 + a_{10} &= 0, \\ a_{11} + a_{12} &= 0. \end{aligned} \tag{3.6}$$

If the linear system (3.6) has a solution $\{\gamma^*_k\}$, since each γ^*_k has roughly the same magnitude $\mathcal{O}(\frac{1}{h})$, then we can obtain a second-order interpolation scheme for the normal derivative $u^-_{\mathbf{n}}$ by choosing an appropriate correction term Q. From (3.4) and (3.6), we can see that the system of equations for the $\{\gamma_k\}$'s is independent of the augmented variable $[u_{\mathbf{n}}]$ which means that we can calculate $\{\gamma_k\}$ outside of the GMRES iteration.

If more than six different grid points ($k_s > 6$) in a neighborhood of $\mathbf{X}$ are used in the interpolation, we will have an underdetermined system of linear equations that has an infinite number of solutions. Usually $9 - 16$ closest grid point to $\mathbf{X} = (X, Y)$ are chosen as the interpolation stencil. In our work, we use 12 closest grid points to interpolate the normal derivatives $U_{\mathbf{n}}^{\pm}(\mathbf{X})$. For stability, we solve (3.6) using SVD. Given a system of $Mx = b$, let the singular value decomposition of M be $M = U\Sigma V^H$, where U and V are two unitary matrices, $\Sigma = \text{diag}(D, \mathbf{0})$ with D being invertible. Then the SVD solution of $Mx = b$ is $x^* = V\Sigma^+ U^H b$, where $\Sigma^+ = \text{diag}(D^{-1}, \mathbf{0})$. The SVD solution has the smallest 2-norm among all possible solutions,

$$\sum_{k=0}^{k_s-1} (\gamma_k^*)^2 = \min \sum_{k=1}^{k_s-1} \gamma_k^2 \qquad \text{subject to} \quad (3.6).$$

For such a solution, the magnitude of γ_k^* is well under control, which is crucial to the stability of the entire algorithm.

Once the $\{\gamma_k\}$'s are computed, we will have the $\{a_i\}$'s, and the correction term Q is then determined from the following:

$$\begin{aligned} Q \;=\; \Big\{ & a_2 w + a_4 g + a_6 w' + a_8 \big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - \frac{\beta_\eta^+}{\beta^+} w' - w'' + \left[\frac{f}{\beta}\right] \big) \\ & + a_{10}(-g\chi'' + w'') + a_{12}(w'\chi'' + g') \Big\}. \end{aligned} \tag{3.7}$$

Since we represent the interface by the zero level set of a Liptschiz continuous function $\phi(\mathbf{x})$, we can use the least squares interpolation to approximate the surface derivatives g', w' and w'' using the values of g and w at the control points. The control points we use are the orthogonal projections of irregular grid points.

Thus, we are able to compute $U_{\mathbf{n}}^{-}(\mathbf{X})$ to second-order accuracy. We can derive a

formula for $U_{\mathbf{n}}^{+}(\mathbf{X})$ in exactly the same way. However, with the relation $U_{\mathbf{n}}^{+}(\mathbf{X}) = U_{\mathbf{n}}^{-}(\mathbf{X}) + g(\mathbf{X})$, we can write down a second-order interpolation scheme for $U_{\mathbf{n}}^{+}(\mathbf{X})$ immediately

$$U_{\mathbf{n}}^{+}(\mathbf{X}) \approx \sum_{k=0}^{k_s-1} \gamma_k U_{i+i_k,j+j_k} - Q + g, \tag{3.8}$$

where $\{\gamma_k\}$'s are the coefficients of the interpolation scheme for $U_{\mathbf{n}}^{-}(\mathbf{X})$.

3.2 The iterative procedures

A key process in our algorithm is to solve the Schur complement system (2.55) using the GMRES method, with an initial guess

$$\mathbf{G}^{(0)} = \left\{ G_1^{(0)}, G_2^{(0)}, \ldots, G_{N_b}^{(0)} \right\}. \tag{3.9}$$

We need to derive the right-hand side of the Schur complement system, and explain how to compute the matrix-vector multiplication of the system without explicitly forming the coefficient matrix. The right-hand side needs to be computed just once, which is described below.

3.2.1 Right-hand side of the Schur complement system

If we take $\mathbf{G} = \mathbf{0}$, and apply one step of the immersed interface method to solve *Problem* II to get $\mathbf{U}(\mathbf{0})$, then

$$\mathbf{U}(\mathbf{0}) = A^{-1}\bar{\mathbf{F}}. \tag{3.10}$$

With the knowledge of $\mathbf{U}(\mathbf{0})$ and $\mathbf{G} = \mathbf{0}$, we can compute the normal derivatives on each side of the interface to get $\mathbf{U}_{\mathbf{n}}^{\pm}(\mathbf{0})$ using the approach described in the previous section.

Thus the right-hand side of the Schur complex is

$$\begin{aligned} \bar{\mathbf{V}} &= P\mathbf{V} - EA^{-1}\bar{\mathbf{F}} \\ &= P\mathbf{V} - E\mathbf{U}(\mathbf{0}) \\ &= -\left(\beta^{+} \odot \mathbf{U}_{\mathbf{n}}^{+}(\mathbf{0}) - \beta^{-} \odot \mathbf{U}_{\mathbf{n}}^{-}(\mathbf{0}) - \mathbf{V}\right) \end{aligned} \tag{3.11}$$

where the last equality is obtained from (2.53) with $\mathbf{G} = \mathbf{0}$. The $\odot$ notation means element-wise multiplication of two vectors since $\beta^{\pm}$ may not be the same for all the control points. Now we are able to compute the right-hand side of the Schur complement system.

3.2.2 Matrix-vector multiplication of the Schur complement system

Consider the matrix-vector multiplication

$$(D - EA^{-1}B)\mathbf{Q} \tag{3.12}$$

on the left hand side of the Schur complement system, where $\mathbf{Q}$ is an arbitrary vector of dimension N_b. This involves essentially two steps:

1. A multigrid solver for computing

$$\mathbf{U}(\mathbf{Q}) = A^{-1}(\bar{\mathbf{F}} - B\mathbf{Q}) \tag{3.13}$$

   which is the solution to *Problem* II with $\mathbf{G} = \mathbf{Q}$ (see (2.51));

2. The weighted least squares interpolation to compute $\mathbf{U}_{\mathbf{n}}^{+}(\mathbf{Q})$ and $\mathbf{U}_{\mathbf{n}}^{-}(\mathbf{Q})$. The resid-

ual vector in the flux jump condition is

$$\mathbf{R}(\mathbf{Q}) = \mathbf{V} - \Big(\beta^+ \odot \mathbf{U}_{\mathbf{n}}^+(\mathbf{Q}) - \beta^- \odot \mathbf{U}_{\mathbf{n}}^-(\mathbf{Q})\Big), \tag{3.14}$$

which is the same residual vector of the second equation in (2.54) from our definition (see also (2.53)). In other words

$$P\mathbf{V} - (D\mathbf{Q} + E\mathbf{U}(\mathbf{Q})) = \mathbf{R}(\mathbf{Q}) \tag{3.15}$$

The matrix-vector multiplication (3.12) can then be computed as

$$\begin{aligned}
(D - EA^{-1}B)\mathbf{Q} &= D\mathbf{Q} - EA^{-1}B\mathbf{Q} \\
&= D\mathbf{Q} + E\mathbf{U}(\mathbf{Q}) - EA^{-1}\bar{\mathbf{F}} \qquad \text{from (3.13)} \\
&= D\mathbf{Q} + E\mathbf{U}(\mathbf{Q}) - P\mathbf{V} + \bar{\mathbf{V}} \qquad \text{from (2.55)} \\
&= -\mathbf{R}(\mathbf{Q}) + \bar{\mathbf{V}} \qquad \text{from (3.15).}
\end{aligned} \tag{3.16}$$

It is worth to point out that once our algorithm is successfully terminated, which means the residual vector is close to the zero vector, we not only have an approximation $\mathbf{Q}$ to the solution $\mathbf{G}^*$, an approximation $\mathbf{U}(\mathbf{Q})$ to the solution $\mathbf{U}^*$, but also approximations $\mathbf{U}_{\mathbf{n}}^{\pm}(\mathbf{Q})$ to the normal derivatives from each side of the interface. Moreover, we can achieve a second order accuracy in the L^∞ norm for both the solution $\mathbf{U}$ and its normal derivatves $\mathbf{U}_{\mathbf{n}}^{\pm}$.

3.2.3 The multigrid solver

For a given $\mathbf{G}$, we want to solve for $\mathbf{U}$ in (2.51). In this dissertation, we choose to use the multigrid solver DMGD9V developed by De Zeeuw to solve the linear equation in (2.51). Since the negative of the resulting finite difference coefficient matrix is an M-matrix, the multigrid solver is guaranteed to converge.

3.3 A new preconditioner for the Schur complement system

With the procedures described in the previous sections, we are able to solve *Problem* I to second order accuracy. In each iteration, we need to use the multigrid method to solve for an updated $\mathbf{U}$, which is equivalent to compute A^{-1} in the Schur complement system. Note we never compute the inverse of the matrix A, instead we do a multigrid solve. The number of iterations of the GMRES method depends on the condition number of the Schur complement. If we make use of both (3.1) and (3.8) to compute $\mathbf{U}_{\mathbf{n}}^{\pm}$, the condition number is proportional to $1/h$. Therefore, the number of iterations will grow as we increase the number of grid points.

Previously, Z. Li has proposed an efficient preconditioner for the Schur complement system, which works very well for piecewise constant coefficients. But that preconditioner gives slow convergence for our current problems with variable coefficients. Below we propose a new preconditioner which will improve the condition number of the Schur complement system.

We first write the augmented equation in (2.52) for each control point explicitly,

$$\beta^+(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^+(\mathbf{X}_i) - \beta^-(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^-(\mathbf{X}_i) = \mathbf{V}(\mathbf{X}_i), \quad i = 1, \ldots, N_b. \tag{3.17}$$

In the GMRES iterations, we also write the corresponding residual equation in (3.14) for each control point explicitly,

$$\mathbf{R}(\mathbf{X}_i) = \mathbf{V}(\mathbf{X}_i) - \Big(\beta^+(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^+(\mathbf{X}_i) - \beta^-(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^-(\mathbf{X}_i)\Big), \ \ i = 1, \ldots, N_b. \tag{3.18}$$

The new idea of preconditioning the Schur complement system is to rescale the augmented equation to the following

$$\frac{\beta^+(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^+(\mathbf{X}_i) - \beta^-(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^-(\mathbf{X}_i)}{\overline{\beta(\mathbf{X}_i)}} = \frac{\mathbf{V}(\mathbf{X}_i)}{\overline{\beta(\mathbf{X}_i)}}, \quad i = 1, \ldots, N_b, \tag{3.19}$$

where $\overline{\beta(\mathbf{X}_i)} = \frac{\beta^+(\mathbf{X}_i)+\beta^-(\mathbf{X}_i)}{2}$. Then the resulting residual in the GMRES iterations is also rescaled as

$$\mathbf{R}^{\text{rescale}}(\mathbf{X}_i) = \frac{\mathbf{V}(\mathbf{X}_i) - \Big(\beta^+(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^+(\mathbf{X}_i) - \beta^-(\mathbf{X}_i)\mathbf{U}_{\mathbf{n}}^-(\mathbf{X}_i)\Big)}{\overline{\beta(\mathbf{X}_i)}}, \quad i = 1, \ldots, N_b. \tag{3.20}$$

With this rescaling, the number of iterations for solving the Schur complement system is independent of the mesh size h, and almost independent of the jump $[\beta]$ in the coefficient as well. Although we have not been able to prove this claim analytically, the algorithm works very successfully in our numerical experiments, see the next Chapter for more analysis.

Chapter 4

Numerical Experiments and Analysis

We have performed a number of numerical experiments to confirm that the second order accuracy of both the solution and and its gradient can be obtained by our new method. Here we show several examples with numerical results and convergence analysis. We also include the CPU time for each computation to show how fast will our new method run.

4.1 Accuracy study from two typical experiments

Example 4.1.1. In this example, the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the rectangular domain $[-1, 1] \times [-1, 1]$. The coefficient is given by

$$\beta(x, y) = \begin{cases} e^{10x} & \text{if } (x, y) \in \Omega^-, \\ \sin(x + y) + 2 & \text{if } (x, y) \in \Omega^+. \end{cases} \tag{4.1}$$

The jump conditions $[u]$ and $[\beta u_{\mathbf{n}}]$, as well as the source term f are determined from the exact solution

$$u(x,y) = \begin{cases} \sin(x+y) & \text{if } (x,y) \in \Omega^-, \\ \log(x^2+y^2) & \text{if } (x,y) \in \Omega^+. \end{cases} \tag{4.2}$$

We present a grid refinement analysis in Table 4.1. For all the examples, we use the pointwise L^∞ norm to measure the errors in the computed solution U_{ij} and their normal derivatives $U^{\pm}_{\mathbf{n}}$ on the interface. The L^∞ norm captures the maximum error over all grid points.

The relative error of the computed solution $U_{i,j}$ is defined as

$$E(U)_N = \frac{\max\limits_{i,j} |u(x_i,y_j) - U_{ij}|}{\max\limits_{i,j} |u(x_i,y_j)|}. \tag{4.3}$$

The order of convergence is defined as

$$\text{order} = \left| \frac{\log(E(U)_{N_1}/E(U)_{N_2})}{\log(N_1/N_2)} \right|. \tag{4.4}$$

Similarly, the relative error of the computed normal derivatives $U^{\pm}_{\mathbf{n}}$ are defined as

$$E(U^+_{\mathbf{n}})_N = \frac{\max\limits_{1\le p\le N_b} |u^+_{\mathbf{n}}(\mathbf{X}_p) - U^+_{\mathbf{n}\,p}|}{\max\limits_{1\le p\le N_b} |u^+_{\mathbf{n}}(\mathbf{X}_p)|}, \tag{4.5}$$

$$E(U^-_{\mathbf{n}})_N = \frac{\max\limits_{1\le p\le N_b} |u^-_{\mathbf{n}}(\mathbf{X}_p) - U^-_{\mathbf{n}\,p}|}{\max\limits_{1\le p\le N_b} |u^-_{\mathbf{n}}(\mathbf{X}_p)|}, \tag{4.6}$$

where $\mathbf{X}_p$, $1 \le p \le N_b$ is the control points on the interface.

In all tables of this chapter, N_b is the number of control points that are used to

represent the interface; N_{coarse} and N_{finest} are the number of the coarsest and finest grid lines for the multigrid solver DMGD9V. For all the numerical examples in this chapter, we use a GMRES convergence tolerance of 10^{-6}. The convergence tolerance of the multigrid solver may differ from problems, so we will list it specifically for each example.

Table 4.1: Numerical results and convergence analysis for Example 4.1.1, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
66	96	0.28805E-01		0.88682E-01		0.12769E-01		8	0.160
130	184	0.98473E-02	1.58	0.32375E-01	1.49	0.46012E-02	1.51	8	0.533
258	368	0.25642E-02	1.96	0.88674E-02	1.89	0.13434E-02	1.80	8	2.272
514	728	0.66291E-03	1.96	0.23339E-02	1.94	0.35159E-03	1.94	8	11.284
1026	1452	0.16604E-03	2.00	0.58702E-03	2.00	0.88848E-04	1.99	8	38.851
2050	2900	0.42837E-04	1.96	0.15218E-03	1.95	0.22854E-04	1.96	8	174.056

We can see from Table 4.1 that second order convergence is achieved for both the solution and its normal derivatives. The convergence tolerance of the multigrid solver is set to 10^{-6}. The number of GMRES iterations is 8 for all grid sizes, which indicates that it is independent of grid sizes. Figure 4.1 shows the plot of the computed solution and the distribution of the relative error.

Example 4.1.2. In this example, the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the rectangular domain $[-1, 1] \times [-1, 1]$. The coefficient is given by

$$\beta(x, y) = \begin{cases} \cos(x + y) + 2 & \text{if } (x, y) \in \Omega^-, \\ \sin(x + y) + 2 & \text{if } (x, y) \in \Omega^+. \end{cases} \tag{4.7}$$

The jump conditions $[u]$ and $[\beta u_{\mathbf{n}}]$, as well as the source term f are determined from the

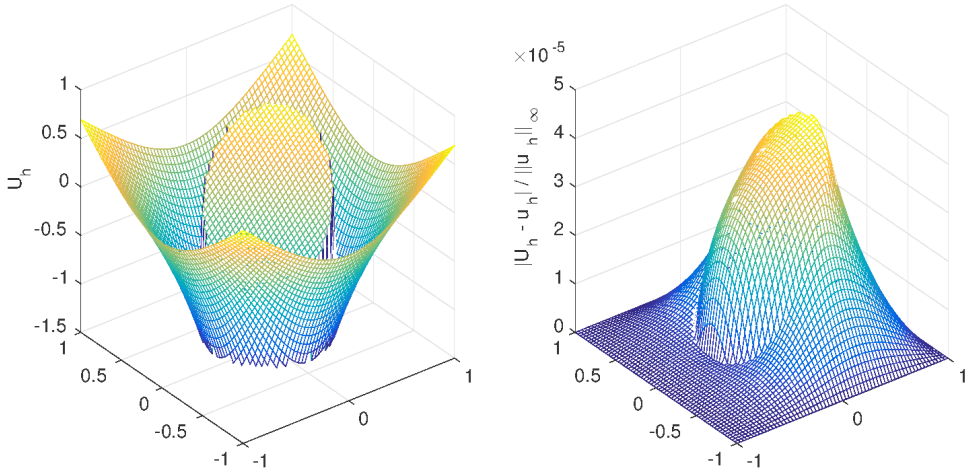

Figure 4.1: The computed solution and the distribution of the relative error for Example 4.1.1.

exact solution

$$u(x,y) = \begin{cases} \sin(x+y) & \text{if } (x,y) \in \Omega^-, \\ \log(x^2+y^2) & \text{if } (x,y) \in \Omega^+. \end{cases} \tag{4.8}$$

This example only differs from Example 4.1.1 in the coefficient $\beta(x,y)$ when $(x,y) \in \Omega^-$. But this will change the jump conditions $[u]$ and $[\beta u_{\mathbf{n}}]$, as well as the source term f. Table 4.2 shows a grid refinement analysis for Example 4.1.2. Again, we confirm second order accuracy for both the computed solution U_{ij} and their normal derivatives $U^{\pm}_{\mathbf{n}}$ on the interface. Figure 4.2 shows the plot of the computed solution U_{ij} on the left, which is the same as the one in Figure 4.1 since the exact solutions of both examples are identical. The plot on the right of Figure 4.2 shows the distribution of the relative error for Example 4.1.2, which is quite different from the one in Figure 4.1 since the jump conditions and the source term are different.

Table 4.2: Numerical results and convergence analysis for Example 4.1.2, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
66	96	0.51511E-03		0.30699E-02		0.11532E-02		4	0.095
130	188	0.95949E-04	2.48	0.92539E-03	1.77	0.31292E-03	1.92	4	0.305
258	368	0.24926E-04	1.97	0.23168E-03	2.02	0.85169E-04	1.90	4	1.266
514	732	0.58605E-05	2.10	0.59349E-04	1.98	0.23245E-04	1.88	4	5.442
1026	1456	0.22833E-05	1.36	0.17030E-04	1.81	0.60299E-05	1.95	4	25.246
2050	2908	0.35940E-06	2.67	0.38769E-05	2.14	0.15288E-05	1.98	4	232.066

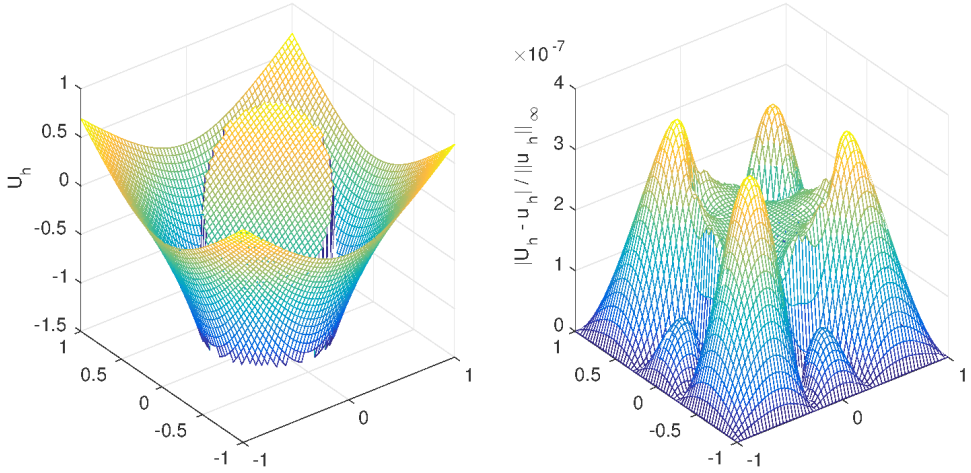

Figure 4.2: The computed solution and the distribution of the relative error of Example 4.1.2.

4.2 The number of GMRES iterations versus the mesh size h

Example 4.2.1. In this example, the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the rectangular domain $[-1, 1] \times [-1, 1]$. The differential equations are

$$(\beta u_x)_x + (\beta u_y)_y = f(\mathbf{x}) + C \int_\Gamma \delta(\mathbf{x} - \mathbf{X}(s))\, ds, \tag{4.9}$$

with $f(\mathbf{x}) = 8(x^2+y^2)+4$ and

$$\beta(x,y) = \begin{cases} x^2+y^2+1 & \text{if } (x,y) \in \Omega^-, \\ b & \text{if } (x,y) \in \Omega^+. \end{cases} \tag{4.10}$$

The exact solution is in the following form

$$u(x,y) = \begin{cases} r^2 & \text{if } (x,y) \in \Omega^-, \\ 1 - \dfrac{1}{8b} - \dfrac{1}{b} + \dfrac{\frac{r^4}{2}+r^2}{b} + \dfrac{C\log(2r)}{b} & \text{if } (x,y) \in \Omega^+, \end{cases}$$

where $r = \sqrt{x^2+y^2}$ and b, C are constants.

The convergence analysis of Example 4.2.1 is listed in Table 4.3 and Table 4.4 for two different setups. We can see that second order accuracy is achieved for both the solution and the normal derivatives in both the large ratio case ($b = 0.05, \beta^-/\beta^+ = 25$) and the small ratio case ($b = 3.5, \beta^-/\beta^+ = 0.36$). Figure 4.3 shows the plots of the computed solution and the corresponding error distribution for the small ratio case, while Figure 4.4 shows that for the large ratio case.

Table 4.3: Numerical results and convergence analysis for Example 4.2.1, $N_{\text{coarse}} = 5$, $C = 0.1, b = 0.05$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
66	96	0.11806E-02		0.10858E-01		0.93667E-02		6	0.103
130	188	0.29244E-03	2.06	0.29057E-02	1.94	0.25065E-02	1.94	6	0.342
258	368	0.71380E-04	2.06	0.70487E-03	2.07	0.60806E-03	2.07	5	1.258
514	732	0.16640E-04	2.11	0.17465E-03	2.02	0.15052E-03	2.03	5	5.540
1026	1456	0.41334E-05	2.01	0.44847E-04	1.97	0.38020E-04	1.99	4	20.863
2050	2908	0.10796E-05	1.94	0.11771E-04	1.93	0.98363E-05	1.95	4	201.511

Table 4.4: Numerical results and convergence analysis for Example 4.2.1, $N_{\text{coarse}} = 5$, $C = 0.1, b = 3.5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
66	96	0.32171E-03		0.76023E-03		0.65468E-03		3	0.065
130	184	0.82564E-04	2.01	0.21392E-03	1.87	0.18403E-03	1.87	3	0.237
258	368	0.20135E-04	2.06	0.57280E-04	1.92	0.49334E-04	1.92	3	1.027
514	728	0.50484E-05	2.01	0.14198E-04	2.02	0.13109E-04	1.92	2	4.271
1026	1452	0.12626E-05	2.01	0.34733E-05	2.04	0.33715E-05	1.96	2	13.788
2050	2900	0.31725E-06	2.00	0.93660E-06	1.89	0.10136E-05	1.74	2	64.238

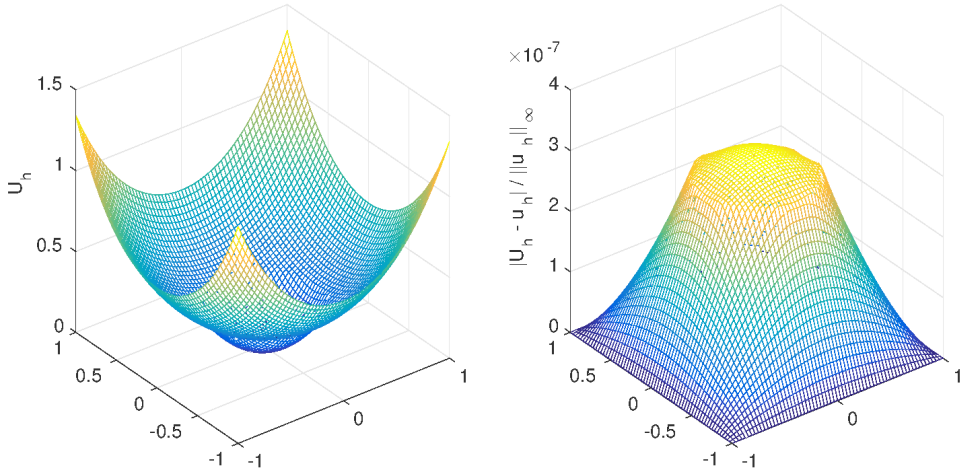

Figure 4.3: The computed solution and the distribution of the relative error of Example 4.2.1, with $C = 0.1, b = 3.5$.

Note that for all the examples we have studied so far, the number of GMRES iterations is almost independent of the mesh size. Figure 4.5 shows the number of GMRES iterations versus the number of grid lines N in the x-direction ($N = M$) for all the examples we have discussed so far.

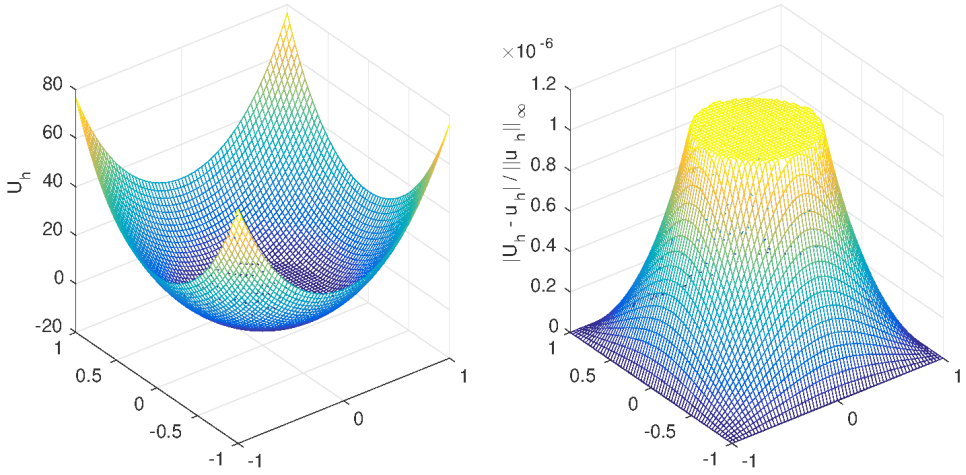

Figure 4.4: The computed solution and the distribution of the relative error of Example 4.2.1, with $C = 0.1, b = 0.05$.

4.3 The number of GMRES iterations versus the jump ratio $\rho = \beta^-/\beta^+$

Figure 4.6 shows the number of GMRES iterations versus the jump ratio $\rho = \beta^-/\beta^+$ in the log-log scale for Example 4.2.1 with fixed mesh size $M = N = 130$. $\rho = 1$ corresponds to the case with continuous coefficient β. When ρ deviates from the 1, we have a larger jump in the coefficient β. As ρ deviates from 1, the number of iterations increase proportionally to $|\log(\rho)|$ when ρ is near the unity. When ρ is far away from the unity, the number of iterations reaches a saturation point and remains constant.

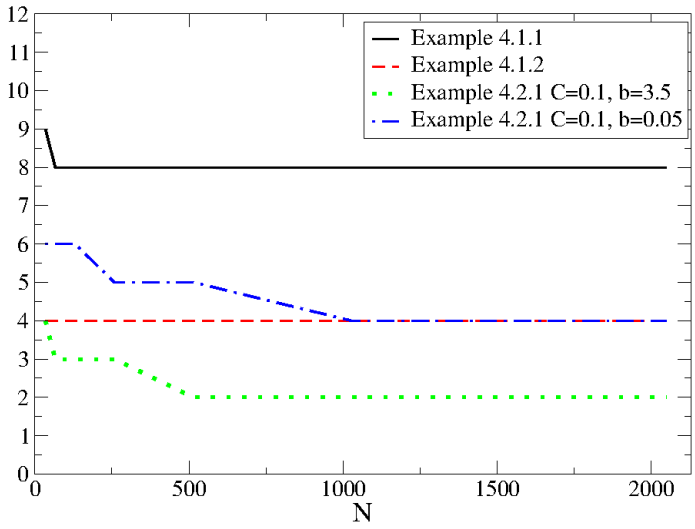

Figure 4.5: The number of GMRES iterations versus the number of grid lines N in the x-direction.

4.4 Applicable to problems with piecewise constant coefficient

In this dissertation, we have developed a new numerical method using the augmented IIM approach and utilizing iterative solvers such as the multigrid solver and the GMRES method. It is designed to work for interface problems with piecewise variable coefficients. But it is also applicable to interface problems with piecewise constant coefficients. So the new method we proposed is essentially a generalization from the fast algorithm developed by Z. Li for interface problems with piecewise constant coefficients. Hence our new method should also work for piecewise constant coefficients. Remember that we have also developed a new preconditioner for the Schur complement in Chapter 3, which is quite different from the original one proposed by Z. Li. Therefore, it is helpful to perform some numerical experiments to interface problems with piecewise constant coefficients and study the convergences. Here we present one example to show that our new method

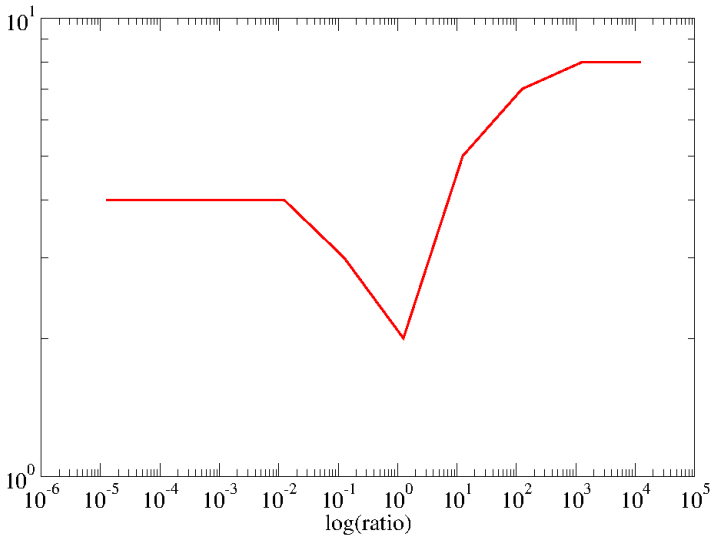

Figure 4.6: The number of GMRES iterations versus the ratio of jumps β^-/β^+ in the log-log scale for Example 4.2.1 with a fixed mesh size $M = 130$ and $N = 130$.

indeed works well in the cases where the coefficients are piecewise constant.

Example 4.4.1. The interface is given by

$$\begin{cases} X = \big(r_0 + \lambda \sin(w\theta)\big) \cos\theta \\ Y = \big(r_0 + \lambda \sin(w\theta)\big) \sin\theta \end{cases} \qquad 0 \le \theta \le 2\pi \tag{4.11}$$

in the rectangular domain $[-1, 1]\times[-1, 1]$. In this example, we use $r_0 = 0.5, \lambda = 0.2, w = 5$ so that the interface is a five-pointed star centered at the origin.

The coefficient is piecewise constant

$$\beta(x, y) = \begin{cases} \beta^- & \text{if } (x, y) \in \Omega^-, \\ \beta^+ & \text{if } (x, y) \in \Omega^+, \end{cases} \tag{4.12}$$

where we set $\beta^- = 1.0$ and $\beta^+ = 3.0$. The exact solution is in the following form

$$u(x,y) = \begin{cases} \dfrac{r^2}{\beta^-} & \text{if } (x,y) \in \Omega^-, \\ \dfrac{r^4 + C_0 \log(2r)}{\beta^+} & \text{if } (x,y) \in \Omega^+, \end{cases}$$

where $r = \sqrt{x^2 + y^2}$ and C_0 is a constant. We set $C_0 = -0.1$ in this example.

Table 4.5: Numerical results and convergence analysis for Example 4.4.1, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
130	312	0.36754E-02		0.23305E+00		0.26544E+00		7	0.576
258	618	0.10946E-02	1.77	0.55982E-01	2.08	0.63760E-01	2.08	7	2.175
514	1230	0.17091E-03	2.69	0.15400E-01	1.87	0.17541E-01	1.87	7	13.775
1026	2452	0.30145E-04	2.51	0.42371E-02	1.87	0.48265E-02	1.87	7	41.462
2050	4898	0.92522E-05	1.71	0.10589E-02	2.00	0.12065E-02	2.00	7	276.882

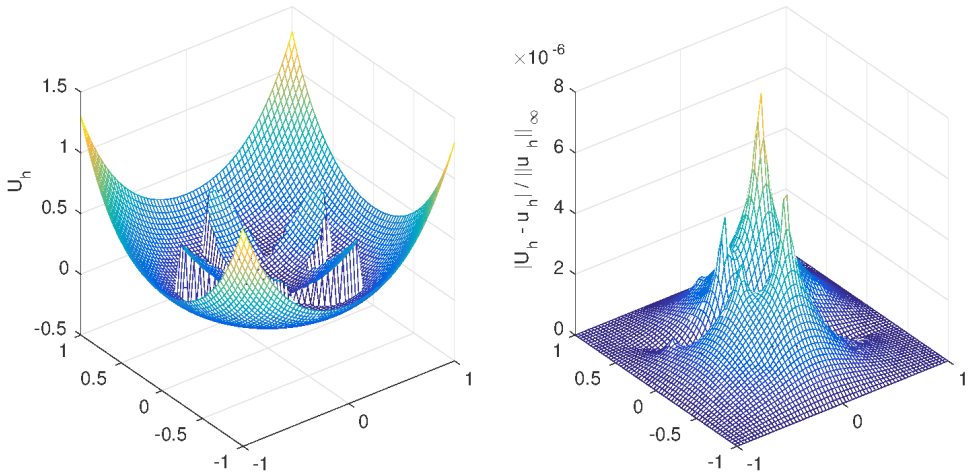

Figure 4.7: The computed solution and the distribution of the relative error for Example 4.4.1.

The convergence analysis of Example 4.4.1 is listed in Table 4.5. The convergence tolerance of the multigrid solver is set to 10^{-6} in this example. We can see that second order convergence is achieved for both the solution and its normal derivatives.

The number of GMRES iterations is reasonably small and almost independent of the mesh grid size. Figure 4.7 shows the plots of the computed solution and the corresponding error distribution, where we can see the interface is a five-pointed star centered at the origin. From this example, we can see that our new method also works well for interface problems with piecewise constant coefficients.

4.5 Generalization of the new method to problems with non-zero $\sigma(x, y)$

So far, we have demonstrated the idea of our new numerical method for elliptic interface problems with $\sigma(x, y) = 0$. Here, we would like to modify this new numerical method a little bit to solve the generalized interface problem with non-zero $\sigma(x, y)$ in the following form

$$\nabla \cdot \big(\beta(x, y)\nabla u(x, y)\big) - \sigma(x, y)u(x, y) = f(x, y), \quad (x, y) \in \Omega = \Omega^+ \cup \Omega^-, \tag{4.13}$$

together with the jump conditions across the interface Γ ,

$$[u]_\Gamma = w, \qquad [\beta u_{\mathbf{n}}]_\Gamma = v, \tag{4.14}$$

where $\sigma(x, y) \neq 0$ and $\sigma(x, y)$ is piecewise continuous function, but may have a finite discontinuity across the interface, i.e., $\sigma^{\pm}(x, y) \in C$. The regulations of other terms

are the same as previously described. Here, we assume that $\sigma(x, y) \geq 0$ to make the negative of the finite difference coefficient matrix an M-matrix. But our numerical method described below should also work for $\sigma(x, y) < 0$ with modest magnitude. However, different approach may be needed to solve problems with large negative σ.

4.5.1 Modifications in the numerical method

Here, we briefly illustrate how to modify our current numerical method to solve elliptic interface PDEs with non-zero $\sigma(x, y)$.

Since the last equality of the interface relations in Theorem 1 is derived from the PDE itself, we need to add two extra terms to the last equality in (2.13) when $\sigma(x, y) \neq 0$. So we have a new interface relation for $u_{\xi\xi}$,

$$\begin{aligned}
[u_{\xi\xi}] &= g\left(\chi'' - \frac{\beta_\xi^+}{\beta^+}\right) - w'' - \left[\frac{\beta_\xi}{\beta}\right] u_\xi^- - \left[\frac{\beta_\eta}{\beta}\right] u_\eta^- - \frac{\beta_\eta^+}{\beta^+} w' + \left[\frac{f}{\beta}\right] + \left[\frac{\sigma}{\beta}\right] u^- + \frac{\sigma^+}{\beta^+} w, \\
&= g\left(\chi'' - \frac{\beta_\xi^-}{\beta^-}\right) - w'' - \left[\frac{\beta_\xi}{\beta}\right] u_\xi^+ - \left[\frac{\beta_\eta}{\beta}\right] u_\eta^+ - \frac{\beta_\eta^-}{\beta^-} w' + \left[\frac{f}{\beta}\right] + \left[\frac{\sigma}{\beta}\right] u^+ + \frac{\sigma^-}{\beta^-} w,
\end{aligned}$$

where two σ-related terms are added. Note that we have two expressions for the jump of $u_{\xi\xi}$. The first one will be used when the irregular grid point is inside Ω^- domain, while the second one will be used when the irregular point is inside the Ω^+ domain.

As a result of the new form of $[u_{\xi\xi}]$, the jumps $[u_{xx}]$ and $[u_{yy}]$ at the intersections points should also be modified. When the centered irregular grid point (i, j) is in the Ω^- domain, we need to add two extra σ-related terms to $[u_{xx}]$ and $[u_{yy}]$ in (2.33). So the new

forms become

$$
\begin{aligned}
[u_{xx}] &= \cos^2\theta[u_{\xi\xi}] - 2\sin\theta\cos\theta[u_{\xi\eta}] + \sin^2\theta[u_{\eta\eta}] \\
&= -2\sin\theta\cos\theta(w'\chi'' + g') + \sin^2\theta(-g\chi'' + w'') \\
&\quad + \cos^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - w'' - \frac{\beta_\eta^+}{\beta^+}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^- + \sin\theta u_y^-) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^- + \cos\theta u_y^-) \\
&\quad + \left[\frac{\sigma}{\beta}\right]u^- + \frac{\sigma^+}{\beta^+}w\Big\}, \\
[u_{yy}] &= \sin^2\theta[u_{\xi\xi}] + 2\sin\theta\cos\theta[u_{\xi\eta}] + \cos^2\theta[u_{\eta\eta}] \\
&= 2\sin\theta\cos\theta(w'\chi'' + g') + \cos^2\theta(-g\chi'' + w'') \\
&\quad + \sin^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - w'' - \frac{\beta_\eta^+}{\beta^+}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^- + \sin\theta u_y^-) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^- + \cos\theta u_y^-) \\
&\quad + \left[\frac{\sigma}{\beta}\right]u^- + \frac{\sigma^+}{\beta^+}w\Big\},
\end{aligned}
\tag{4.15}
$$

where u^- at the intersections can be approximated by $u(x_i, y_j)$ at the center grid point (i, j) using Taylor expansion. Since the distance between the intersection point and the center point (i, j) is less than or equal to h_x or h_y, we have

$$
u^-|_{\text{intersection}} = u(x_i, y_j) + \mathcal{O}(\max\{h_x, h_y\})\,. \tag{4.16}
$$

However, when the centered grid point (i, j) is in the Ω^+ domain, we add two extra σ-related terms to the alternative expressions of $[u_{xx}]$ and $[u_{yy}]$ in (2.35). So the new

forms become

$$
\begin{aligned}
[u_{xx}] &= \cos^2\theta[u_{\xi\xi}] - 2\sin\theta\cos\theta[u_{\xi\eta}] + \sin^2\theta[u_{\eta\eta}] \\
&= -2\sin\theta\cos\theta(w'\chi'' + g') + \sin^2\theta(-g\chi'' + w'') \\
&\quad + \cos^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^-}{\beta^-}) - w'' - \frac{\beta_\eta^-}{\beta^-}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^+ + \sin\theta u_y^+) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^+ + \cos\theta u_y^+) \\
&\quad + \left[\frac{\sigma}{\beta}\right]u^+ + \frac{\sigma^-}{\beta^-}w\Big\}, \\
[u_{yy}] &= \sin^2\theta[u_{\xi\xi}] + 2\sin\theta\cos\theta[u_{\xi\eta}] + \cos^2\theta[u_{\eta\eta}] \\
&= 2\sin\theta\cos\theta(w'\chi'' + g') + \cos^2\theta(-g\chi'' + w'') \\
&\quad + \sin^2\theta\Big\{\big(g(\chi'' - \frac{\beta_\xi^-}{\beta^-}) - w'' - \frac{\beta_\eta^-}{\beta^-}w' + \left[\frac{f}{\beta}\right]\big) \\
&\quad - \left[\frac{\beta_\xi}{\beta}\right](\cos\theta u_x^+ + \sin\theta u_y^+) - \left[\frac{\beta_\eta}{\beta}\right](-\sin\theta u_x^+ + \cos\theta u_y^+) \\
&\quad + \left[\frac{\sigma}{\beta}\right]u^+ + \frac{\sigma^-}{\beta^-}w\Big\},
\end{aligned}
\tag{4.17}
$$

where u^+ at the intersections can be approximated by $u(x_i, y_j)$ at the center grid point (i, j) using Taylor expansion,

$$
u^+|_{\text{intersection}} = u(x_i, y_j) + \mathcal{O}(\max\{h_x, h_y\})\,. \tag{4.18}
$$

Another place affected by the new form of $[u_{\xi\xi}]$ is the least squares interpolation scheme for the normal derivatives $U_{\mathbf{n}}^{\pm}$. We also need to add two extra σ-related terms to

the expression of $u^-_{\mathbf{n}}$ in (3.5), so it becomes

$$\begin{aligned}\frac{\partial u^-}{\partial \mathbf{n}}(\mathbf{X}) \;\approx\; & \big(a_1+a_2+a_8\Big[\frac{\sigma}{\beta}\Big]\big)u^- + \big(a_3+a_4-a_8\Big[\frac{\beta_\xi}{\beta}\Big]\big)u^-_\xi \\ & + \big(a_5+a_6-a_8\Big[\frac{\beta_\eta}{\beta}\Big]\big)u^-_\eta + (a_7+a_8)u^-_{\xi\xi} + (a_9+a_{10})u^-_{\eta\eta} \\ & + (a_{11}+a_{12})u^-_{\xi\eta} + \Big\{a_2 w + a_4 g + a_6 w' \\ & + a_8\big(g(\chi'' - \frac{\beta^+_\xi}{\beta^+}) - \frac{\beta^+_\eta}{\beta^+}w' - w'' + \Big[\frac{f}{\beta}\Big] + \frac{\sigma^+}{\beta^+}w\big) + a_{10}(-g\chi'' + w'') \\ & + a_{12}(w'\chi'' + g')\Big\} \\ & - Q + \mathcal{O}(h^3 \max|\gamma_{ij}|)\,.\end{aligned} \tag{4.19}$$

Consequently, to minimize the interpolation error, we need to add one extra σ-related term to the linear systems of equations for the coefficients $\{\gamma_k\}$ in (3.6). So we have

$$\begin{aligned} a_1 + a_2 + a_8\Big[\frac{\sigma}{\beta}\Big] &= 0, \\ a_3 + a_4 - a_8\Big[\frac{\beta_\xi}{\beta}\Big] &= 1, \\ a_5 + a_6 - a_8\Big[\frac{\beta_\eta}{\beta}\Big] &= 0, \\ a_7 + a_8 &= 0, \\ a_9 + a_{10} &= 0, \\ a_{11} + a_{12} &= 0. \end{aligned} \tag{4.20}$$

We also need to add one extra σ-related term to the correction term Q in (3.7), so its

new form looks as

$$\begin{aligned} Q \;=\; \Big\{ & a_2 w + a_4 g + a_6 w' + a_8 \big(g(\chi'' - \frac{\beta_\xi^+}{\beta^+}) - \frac{\beta_\eta^+}{\beta^+} w' - w'' + \left[\frac{f}{\beta}\right] + \frac{\sigma^+}{\beta^+} w \big) \\ & + a_{10}(-g\chi'' + w'') + a_{12}(w'\chi'' + g') \Big\}. \end{aligned} \tag{4.21}$$

Basically, these are all the modifications we need to make to solve problems with non-zero $\sigma(x, y)$. In the following subsection, we demonstrate through three typical numerical experiments that for problems with non-zero discontinuous $\sigma(x, y)$, we can also achieve second order accuracy not only in the solution itself, but also in its gradient.

4.5.2 Three numerical examples with $\sigma(x, y) \neq 0$

Example 4.5.1. In this example, the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the rectangular domain $[-1, 1] \times [-1, 1]$. The differential equations are

$$(\beta u_x)_x + (\beta u_y)_y - \sigma u = f(\mathbf{x}), \quad (x, y) \in \Omega^+ \cup \Omega^-, \tag{4.22}$$

with

$$\sigma(x, y) = \begin{cases} \sqrt{x^2 + 4y^2} & \text{if } (x, y) \in \Omega^-, \\ \log(x^2 + y^2 + 1) & \text{if } (x, y) \in \Omega^+. \end{cases} \tag{4.23}$$

and

$$\beta(x, y) = \begin{cases} e^x & \text{if } (x, y) \in \Omega^-, \\ x^2 + y^2 + 1 & \text{if } (x, y) \in \Omega^+. \end{cases} \tag{4.24}$$

The jump conditions $[u]$ and $[\beta u_{\mathbf{n}}]$, as well as the source term f are determined from the exact solution

$$u(x,y) = \begin{cases} x^2 - y^2 & \text{if } (x,y) \in \Omega^-, \\ \sin(x)\cos(y) & \text{if } (x,y) \in \Omega^+. \end{cases}$$

We present a grid refinement analysis in Table 4.6. We see that for this case with $\sigma(x,y) \neq 0$, we can also achieve second order accuracy not only in the solution itself, but also in its normal derivatives. The convergence tolerance of the multigrid solver is set to 10^{-6}. The number of GMRES iterations is 4 for all grid sizes in Table 4.6, which indicates it is independent of grid sizes. Figure 4.8 shows the plot of the computed solution and the distribution of the relative error.

Table 4.6: Numerical results and convergence analysis for Example 4.5.1, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^+)$	*order*	$E(U_{\mathbf{n}}^-)$	*order*	Iter	CPU(s)
66	96	0.85969E-03		0.95542E-02		0.59623E-02		4	0.077
130	184	0.18786E-03	2.24	0.25599E-02	1.94	0.15968E-02	1.94	4	0.318
258	368	0.55591E-04	1.78	0.74684E-03	1.80	0.49691E-03	1.70	4	1.272
514	728	0.12783E-04	2.13	0.18721E-03	2.01	0.12500E-03	2.00	4	6.473
1026	1452	0.26051E-05	2.30	0.46393E-04	2.02	0.31318E-04	2.00	4	23.586
2050	2900	0.74611E-06	1.81	0.11647E-04	2.00	0.81641E-05	1.94	4	107.544

Example 4.5.2. In this example, the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the rectangular domain $[-1,1] \times [-1,1]$. The differential equations are

$$(\beta u_x)_x + (\beta u_y)_y - \sigma u = f(\mathbf{x}), \quad (x,y) \in \Omega^+ \cup \Omega^-, \tag{4.25}$$

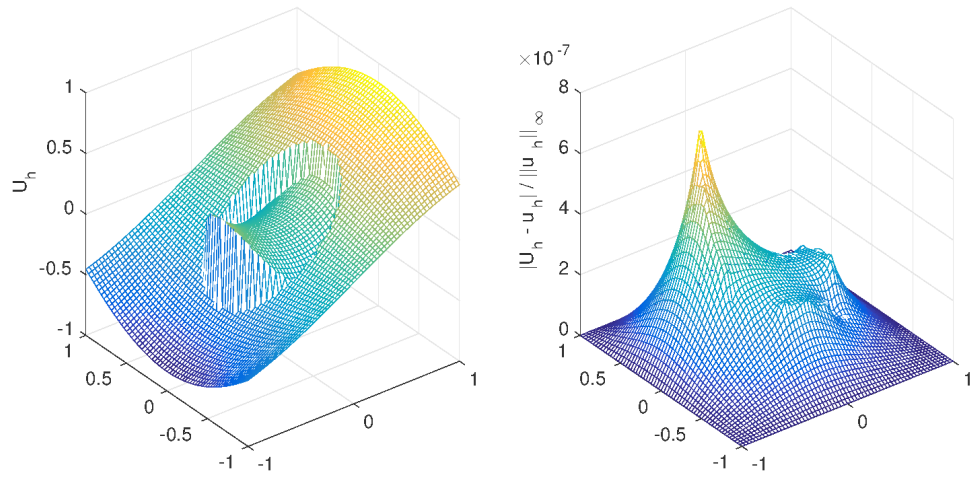

Figure 4.8: The computed solution and the distribution of the relative error for Example 4.5.1.

with

$$\sigma(x,y) = \begin{cases} \cos(xy) + 2 & \text{if } (x,y) \in \Omega^-, \\ x^2 + y^2 + 1 & \text{if } (x,y) \in \Omega^+. \end{cases} \tag{4.26}$$

and

$$\beta(x,y) = \begin{cases} \sin(2x - y) + 3 & \text{if } (x,y) \in \Omega^-, \\ e^{x+2y} & \text{if } (x,y) \in \Omega^+. \end{cases} \tag{4.27}$$

The jump conditions $[u]$ and $[\beta u_{\mathbf{n}}]$, as well as the source term f are determined from the exact solution

$$u(x,y) = \begin{cases} -x^3 + 2y^3 & \text{if } (x,y) \in \Omega^-, \\ y^2 - 2x^2 & \text{if } (x,y) \in \Omega^+. \end{cases}$$

We present a grid refinement analysis in Table 4.7. We see that for this case with $\sigma(x,y) \neq 0$, we can also achieve second order accuracy not only in the solution itself, but

also in its normal derivatives. The convergence tolerance of the multigrid solver is set to 10^{-6}. The number of GMRES iterations is 5 for all grid sizes in Table 4.7, which indicates it is independent of grid sizes. Figure 4.9 shows the plot of the computed solution and the distribution of the relative error.

Table 4.7: Numerical results and convergence analysis for Example 4.5.2, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^{+})$	*order*	$E(U_{\mathbf{n}}^{-})$	*order*	Iter	CPU(s)
66	96	0.43204E-03		0.19545E-01		0.40125E-02		5	0.083
130	184	0.85013E-04	2.40	0.54798E-02	1.88	0.11234E-02	1.88	5	0.341
258	368	0.25923E-04	1.73	0.15025E-02	1.89	0.28031E-03	2.03	5	1.548
514	728	0.57062E-05	2.12	0.37870E-03	2.00	0.74210E-04	1.93	5	7.114
1026	1452	0.11383E-05	2.33	0.93660E-04	2.02	0.17908E-04	2.06	5	25.655
2050	2900	0.33716E-06	1.76	0.25189E-04	1.90	0.47339E-05	1.92	5	122.564

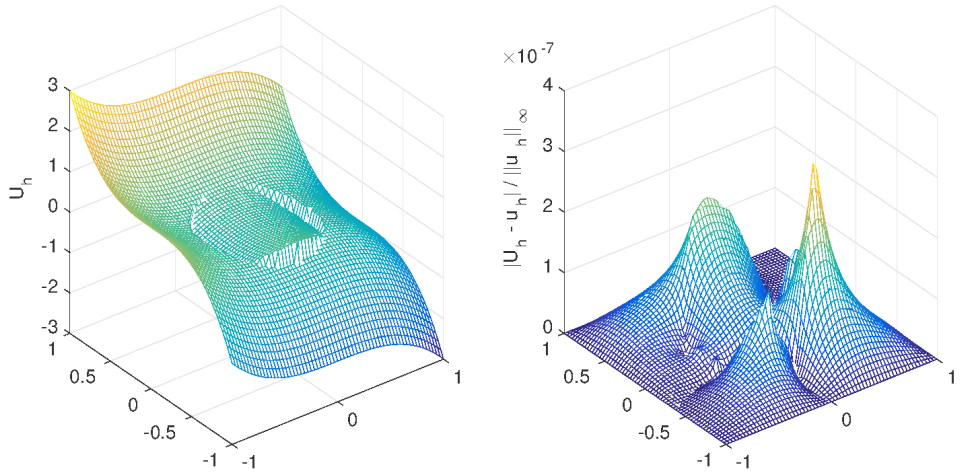

Figure 4.9: The computed solution and the distribution of the relative error for Example 4.5.2.

Example 4.5.3. The differential equations are

$$(\beta u_x)_x + (\beta u_y)_y - \sigma u = f(\mathbf{x}), \quad (x, y) \in \Omega^+ \cup \Omega^-, \tag{4.28}$$

where $\sigma(x, y)$, $\beta(x, y)$ and the exact solution $u(x, y)$ are the same as in Example 4.5.2.

The only difference from Example 4.5.2 is the interface, which is given by

$$\begin{cases} X = \big(r_0 + \lambda \sin(w\theta)\big) \cos\theta \\ Y = \big(r_0 + \lambda \sin(w\theta)\big) \sin\theta \end{cases} \qquad 0 \le \theta \le 2\pi \tag{4.29}$$

in the rectangular domain $[-1, 1] \times [-1, 1]$. We use $r_0 = 0.5, \lambda = 0.1, w = 5$ in this example, and the interface is a five-pointed star centered at the origin.

We present a grid refinement analysis in Table 4.8. We see that for this case with discontinuous $\sigma(x, y)$ across a five-pointed star interface, we can also achieve second order accuracy not only in the solution itself, but also in its normal derivatives. The convergence tolerance of the multigrid solver is set to 10^{-6}. The number of GMRES iterations is 8 for all grid sizes in Table 4.8, which indicates it is independent of grid sizes. Figure 4.10 shows the plot of the computed solution and the distribution of the relative error, where we can see the interface is a five-pointed star centered at the origin. The computed solution in each subdomain looks as that in Figure 4.9 since the exact solutions of the two examples are the same. However, the distribution of the relative error is different from that in Figure 4.9 since the interfaces have different shapes in the two examples.

Table 4.8: Numerical results and convergence analysis for Example 4.5.3, $N_{\text{coarse}} = 5$.

N_{finest}	N_b	$E(U)$	*order*	$E(U_{\mathbf{n}}^{+})$	*order*	$E(U_{\mathbf{n}}^{-})$	*order*	Iter	CPU(s)
130	226	0.45456E-03		0.46124E-01		0.79684E-02		8	0.570
258	446	0.13159E-03	1.81	0.12016E-01	1.96	0.23219E-02	1.80	8	2.900
514	888	0.29301E-04	2.18	0.29090E-02	2.06	0.52542E-03	2.16	8	9.580
1026	1768	0.61645E-05	2.26	0.80398E-03	1.86	0.14383E-03	1.87	8	38.344
2050	3530	0.18536E-05	1.74	0.22244E-03	1.86	0.34908E-04	2.05	8	328.153

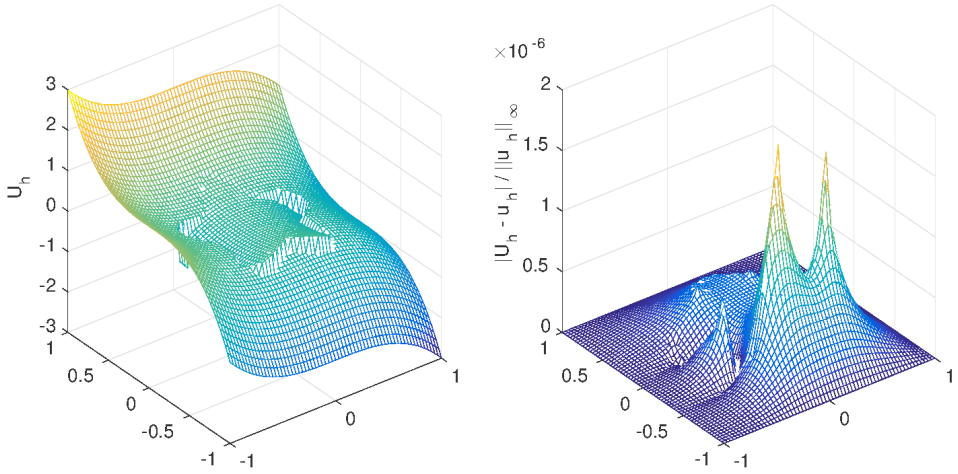

Figure 4.10: The computed solution and the distribution of the relative error for Example 4.5.3.

Chapter 5

Conclusion and Future Work

This dissertation presents a new numerical method for interface problems with piecewise variable coefficients. The new method gives second order accuracy not only for the solution itself, but also for its gradient. The key of this new method lies in introducing the jump of the normal derivative of the solution as an augmented variable, and rewriting the interface problem as a Laplacian of solution with lower order derivative terms near the interface. Thus we can get jump relations for second order derivatives using the augmented variable and the lower order derivative terms. An upwind type discretization is used for the finite difference discretization near or on the interface so that the negative of the discrete coefficient matrix is a M-matrix. A multigrid solver is used to solve the linear system of equations and a GMRES iterative method is used to solve for the augmented variable. Numerical examples and convergence proof are also provided to demonstrate that this new method maintains the second order accuracy of both the solution and the gradient.

Three important highlights of this novel numerical method are summarized below.

First, we can achieve second order accuracy in both the the solution and its gradient

on the interface at the same time. For general interface problems, a second order accurate solution does not necessarily guarantee second order accuracy in its gradient. In fact, people usually get first order accurate gradient on the interface even though the solution itself is second order accurate. The success of this new numerical method lies in the least squares interpolation scheme from a Cartesian grid to an interface, which is second order accurate and has a smooth error distribution. This scheme is a two-sided interpolation that utilizes grid points from both sides of the interface. So it is also very efficient for the GMRES method because the number of GMRES iterations using a two-sided interpolation is often much smaller than using a one-sided interpolation. Using such a least squares interpolation idea, Z. Li has demonstrated the second order accuracy in both solution and its gradient for interface problems with piecewise constant coefficients. In this dissertation, we generalize the discussion to problems with piecewise variable coefficients, and demonstrate for the first time that we can achieve the second order accuracy for both the solution and its gradient.

Second, this new numerical method is fast, efficient and robust. To solve the linear system of equations resulting from discretization, we use a multigrid solver. For all of our numerical experiments, the multigrid solver is highly competitive and about twenty times faster than the other candidate iterative solvers such as Krylov subspace methods. To solve for the augmented variable, we use a GMRES iterative method with a new efficient preconditioner. The original preconditioner in Z. Li's paper works very well for interface problems with piecewise constant coefficients, but it gives slow convergence in all our numerical experiments of interface problems with piecewise variable coefficients. So we propose a new efficient preconditioner that rescales the diagonal element of the Schur complement system. This new preconditioner was tested to be very successful and the number of GMRES iterations is less than 10 for all our examples. Our new

numerical method is also very robust in the sense that the number of GMRES iterations is independent of the size of the grid points. Also, the number of GMRES iterations is reasonably small even for very large ratio of jumps β^-/β^+.

Last but not least, the idea described in this dissertation can be conveniently applied to other related problems. For example, the boundary value problems on an irregular domain. For Dirichlet or Neumann boundary value problems on irregular domains, many challenges exist in direct discretization at the intersections between the boundary and the mesh lines. However, if we embed the irregular domain Ω into a rectangle $R \supset \Omega$ and treat the irregular boundary as the interface, we get an interface problem that could be easily solved using our new method. Furthermore, even though we focus our discussions on elliptic interface problems in this thesis, our new numerical method is readily applicable to hyperbolic and parabolic interface problems. The key of the new method is to introduce the jump in the normal derivative of the solution as the augmented variable and rewrite the interface problem as a Laplacian of the solution with lower order derivative terms near the interface. The same idea would directly work for hyperbolic and parabolic cases. The only difference we need to make is to rewrite the interface problem as another operator of the solution depending on the specific problem, rather than a Laplacian operator of the solution, plus lower order derivative terms on the interface.

In the end, we provide some potential extensions of this work.

First, by performing a series of numerical experiments, we have proved that our new numerical method gives second order accuracy in both the solution itself and its gradient at the same time. The next potential project following this dissertation is to provide a theoretical proof of the convergence and stability of this new method.

Another possible direction is to extend the current method to three dimensional cases. 3D interface problems are usually more challenging to tackle since they require highly

efficient and fast algorithms due to their large size of mesh points. We believe that our new method could be more advantageous in solving 3D interface problems.

REFERENCES

[1] L. M. Adams. A multigrid algorithm for immersed interface problems. In *Proceedings of Copper Mountain Multigrain Conference*, pages 1–14, NASA Conference Publication 3339, 1995.

[2] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM Journal on Scientific and Statistical Computing*, 2(4):430–454, 1981.

[3] J. B. Bell, C. N. Dawson, and G. R. Shubin. An unsplit, higher order godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74(1):1–24, 1988.

[4] F. Brakhagen and T. W. Fogwell. Multigrid methods in modelling porous media flow. In *1st European Conference on the Mathematics of Oil Recovery*, 1989.

[5] P. M. De Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *Journal of Computational and Applied Mathematics*, 33(1):1–27, 1990.

[6] K. Ito and Z. Li. Solving a nonlinear problem in magneto-rheological fluids using the immersed interface method. *Journal of Scientific Computing*, 19(1-3):253–266, 2003.

[7] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.

[8] R. J. LeVeque and Z. Li. Immersed interface methods for stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18(3):709–735, 1997.

[9] R. J. LeVeque and C. Zhang. Finite difference methods for wave equations with discontinuous coefficients. In *Engineering Mechanics*, pages 1038–1041. ASCE, 1995.

[10] Z. Li. *The immersed interface method: a numerical approach for partial differential equations with interfaces*. PhD thesis, University of Washington, 1994.

[11] Z. Li. A note on immersed interface method for three-dimensional elliptic equations. *Computers & Mathematics with Applications*, 31(3):9–17, 1996.

[12] Z. Li. Immersed interface methods for moving interface problems. *Numerical Algorithms*, 14(4):269–293, 1997.

[13] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM Journal on Numerical Analysis*, 35(1):230–254, 1998.

[14] Z. Li and K. Ito. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, volume 33. SIAM Frontiers in Applied Mathematics, 2006.

[15] Z. Li and A. Mayo. ADI method for heat equations with discontinuities along an arbitrary interface. In *Proceedings of Symposia in Applied Mathematics*, volume 48, pages 311–315, 1994.

[16] A. Mayo. The fast solution of Poisson's and the biharmonic equations on irregular regions. *SIAM Journal on Numerical Analysis*, 21(2):285–299, 1984.

[17] A. Mayo. On the rapid evaluation of heat potentials on general regions. Technical report, IBM Technical Report 14305, 1991.

[18] A. Mayo. The rapid evaluation of volume integrals of potential theory on general regions. *Journal of Computational Physics*, 100(2):236–245, 1992.

[19] A. Mayo and A. Greenbaum. Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions. *SIAM Journal on Scientific and Statistical Computing*, 13(1):101–118, 1992.

[20] D. M. McQueen and C. S. Peskin. A three-dimensional computational method for blood flow in the heart. II. contractile fibers. *Journal of Computational Physics*, 82(2):289–297, 1989.

[21] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[22] C. S. Peskin. Lectures on mathematical aspects of physiology. *Lectures in Applied Mathematics*, 19(69):107, 1981.

[23] C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81(2):372–405, 1989.

[24] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[25] G. R. Shubin and J. B. Bell. An analysis of the grid orientation effect in numerical simulation of miscible displacement. *Computer Methods in Applied Mechanics and Engineering*, 47(1):47–71, 1984.

[26] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[27] P. N. Swarztrauber. Fast Poisson Solvers. *Studies in Numerical Analysis*, 24:319–370, Mathematical Association of America, 1984.

[28] A. N. Tikhonov and A. A. Samarskii. Homogeneous difference schemes. *USSR Computational Mathematics and Mathematical Physics*, 1(1):5–67, 1962.