ABSTRACT

CHEN, XIAOHONG. Some Advances and Applications of Immersed Interface Methods. (Under the direction of Zhilin Li.)

In this thesis, two different types of Immersed Interface method (IIM), augmented IIM and direct IIM, are proposed for interface problems and followed by applications of these two methods. Both approaches are originally used to solve elliptic interface problems with a piecewise variable coefficient that has a finite jump across a smooth interface and provide not only a second order accurate solution in the entire domain but also second order accurate gradients from each side of the interface. In the augmented IIM, the partial differential equation (PDE) is reformulated and an augmented variable of co-dimension one is introduced and solved through a Schur complement system. Second order convergence proof is completed for both the solution and the gradient. Combined with Alternating Direction Implicit (ADI) method, the augmented IIM is extended to solve multi-scale parabolic PDE problems in which the coefficient is discontinuous, and the domain has many inclusions with different material parameters. Two ADI methods based on augmented approaches are studied. The introduced augmented variable along the interface enables us to get dimension by dimension jump conditions in a scale needed to get accurate discretization in the coordinate directions. With the new ADI methods, not only is the solution second order accurate globally, but also the augmented variable, which is validated through various examples.

Unlike the augmented approach, the direct IIM discretizes the system directly without using the augmented variable and, hence, is easier to implement and computationally more efficient. For elliptic interface problems, the resulting finite difference scheme is the standard five-point central scheme at regular grid points, while it is a compact nine-point scheme at irregular grid points. The computed solution is second order accurate and will be used to recover the gradient from each side of the interface to the same accuracy. While second order convergence for solutions can be proved by enforcing a discrete elliptic maximum principle, the theoretical convergence proof for the gradient is still an open question. An application of the direct IIM is also presented to solve two-phase incompressible Stokes equations with an interface and a piecewise constant viscosity on staggered grids. In this application, the velocity components and the pressure are placed at different grid points. The Marker and Cell (MAC) scheme is used for discretizing the momentum and continuity equations at regular grid points. The computed numerical solutions are second order accurate in the L^∞ norm for both the velocity and pressure, which is demonstrated in numerical tests.

Some Advances and Applications of Immersed Interface Methods

by
Xiaohong 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

2018

APPROVED BY:

Mansoor Haider	Kazufumi Ito
Donald E. K. Martin	Zhilin Li Chair of Advisory Committee

DEDICATION

In memory of my Grandfather Xianjin Yang.

BIOGRAPHY

Xiaohong Chen was born in 1991, Quanzhou, Fujian province, P. R. China and grew up in Lianyungang, Jiangsu province. He attended Nanjing University of Aeronautics and Astronautics from 2009 to 2013 and graduated with a B.E. in Aircraft Design and Engineering.

After undergraduate school, the author joined the graduate program at North Carolina State University in Raleigh, and in 2015 received his Master of Financial Mathematics. After graduation, he entered the Ph.D. program in Applied Mathematics at the same university. Under the advisorship of Zhilin Li, the author started to research and publish in numerical methods for partial differential equations.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Zhilin Li for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank the rest of my thesis committee: Dr. Mansoor Haider, Dr. Kazufumi Ito, and Dr. Donald E. K. Martin, for their encouragement, insightful comments, and hard questions.

Last but not the least, I would like to thank my parents, Rongkai Chen and Qingzhen Wu, and my girlfriend, Xun Ma, for their care and support over the years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In this thesis, two different types of Immersed Interface method (IIM), augmented IIM and direct IIM, are proposed for elliptic interface problems and followed by their applications in solving parabolic interface problems and two-phase Stokes equations.

1.1 The model problems

1.1.1 Elliptic interface problems

We first consider the elliptic interface problem

$$\nabla \cdot (\beta(\mathbf{x}) \nabla u(\mathbf{x})) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \setminus \Gamma, \quad \Omega = \Omega^+ \cup \Omega^-, \tag{1.1}$$

$$[u](\mathbf{X}) = w(\mathbf{X}), \qquad [\beta u_n](\mathbf{X}) = v(\mathbf{X}), \qquad \mathbf{X} \in \Gamma, \tag{1.2}$$

in one and two space dimensions, where for example, $[u] = [u]_\Gamma(\mathbf{X}) = u^+(\mathbf{X}) - u^-(\mathbf{X})$ is the difference of the limiting values of $u(\mathbf{X})$ from Ω^+ and Ω^- sides, respectively, $u_n = \mathbf{n} \cdot \nabla u = \frac{\partial u}{\partial n}$ is the normal derivative of solution $u(\mathbf{X})$, and $\mathbf{n}(\mathbf{X})$ is the unit normal direction at a point $\mathbf{X}$ on the interface pointing to Ω^+ side, see Fig. 1.1 for an illustration. We use $\mathbf{x}$ to represent a point in the domain while $\mathbf{X}$ a point on the interface Γ. Since a finite difference discretization will be used, we assume that $f(\mathbf{x}) \in C(\Omega^\pm)$, $\beta(\mathbf{x}) \in C^1(\Omega^\pm)$, excluding Γ; and $\Gamma \in C^2$, $w \in C^2(\Gamma)$, $v \in C^1(\Gamma)$. All the parameters and $\frac{\partial \beta}{\partial x}$ and $\frac{\partial \beta}{\partial y}$ are assumed to be bounded. For the regularity requirement of the problem, we also assume that $\beta(\mathbf{x}) \geq \beta_0 > 0$ and $f(\mathbf{x}) \in C^\nu(\Omega \backslash \Gamma)$, for a constant $\nu > 0$ so that $u(\mathbf{x}) \in C^{2+\nu}(\Omega^\pm)$, see [17, 2]. For the error analysis, piecewise higher regularity assumptions are needed for the solution.

Many free boundary and moving interface problems can be modelled by differential equations involving not only the solution to the governing equations, but also the gradient of the solution

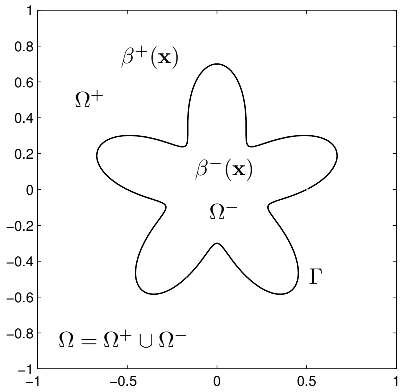

Figure 1.1: A diagram of a rectangular domain $\Omega = \Omega^+ \cup \Omega^-$ with an interface Γ. The coefficient $\beta(\mathbf{x})$ has a finite jump across the interface Γ.

at the free boundary or moving interface from each side. Such examples include the Stefan problem and crystal growth modeling the interface between ice and water in which the velocity of the interface depends on the temperature of the heat equation and its gradient at the interface (called the Stefan condition), [9, 42]; the Hele-Shaw flow [25, 27]; the coupling between a Darcy's system and Stokes or Navier-Stokes equations [32]; and open and traction problems [43, 50]. The most expensive part of simulations from our research on those problems is to solve one or several elliptic interface problems, for example, two generalized Helmholtz and one Poisson equations when we solve the 2D incompressible Navier-Stokes equations involving an interface using the projection method [43].

1.1.2 Parabolic interface problems

The second model problem in this thesis is a two-dimensional parabolic equation

$$\begin{aligned} &u_t = (\beta u_x)_x + (\beta u_y)_y - f, \qquad (x,y) \in \Omega^+ \cup \Omega^- \setminus \Gamma, \\ &u(x,y,0) = u_0(x,y), \qquad \text{Given BC on } \partial\Omega^+, \end{aligned} \tag{1.3}$$

with a fixed interface $\Gamma \in C^2$, where $u_t = \frac{\partial u}{\partial t}$, $u_x = \frac{\partial u}{\partial x}$ and so on, BC stands for a boundary condition along $\partial\Omega^+$ which can be Dirichlet, Neumann, or mixed along different parts of the boundary $\partial\Omega^+$ (see Fig. 1.2 for an illustration). Two more conditions are needed at the interface Γ to close the system, typically the natural jump conditions which state that both the solution and the flux are continuous across the interface Γ. We assume that the coefficient β has a finite jump across the interface Γ corresponding to different materials or states and with a source term along the interface, as in the Peskin's Immersed Boundary model [57],

$$\begin{aligned} u_t = (\beta u_x)_x + (\beta u_y)_y - f(\mathbf{x},t) - \int_\Gamma v(\mathbf{X}(s),t)\,\delta(\mathbf{x}-\mathbf{X}(s))\,ds, \\ \mathbf{x}\in\Omega=\Omega^+\cup\Omega^-, \qquad \mathbf{X}(s)\in\Gamma, \end{aligned} \tag{1.4}$$

where $\mathbf{x}=(x,y)$ is a point in the domain, $\mathbf{X}(s)=(X(s),Y(s))$ is a parametric form of the interface Γ. Note that the equation above is defined in the entire domain including the interface Γ while (1.3) is defined in each sub-domain Ω^+ or Ω^- excluding Γ.

We use the expression (1.3) instead of (1.4) to avoid using a discrete delta function so that we can design accurate numerical schemes. We assume that $f(\mathbf{x},t)$ is piecewise continuous in the domain with a possible finite discontinuity across the interface Γ. Thus the solution of $u(\mathbf{x},t)$ is a piecewise $C^2(\Omega^\pm)$ function in the interior of each sub-domain but coupled together across the interface Γ by two jump conditions (also called internal boundary conditions),

$$[u]_{\mathbf{X}\in\Gamma} = w(s,t), \qquad \left[\beta\frac{\partial u}{\partial n}\right]_{\mathbf{X}\in\Gamma} = v(s,t), \tag{1.5}$$

where for example,

$$\begin{aligned} [\beta u_n]_{\mathbf{X}\in\Gamma} = \left[\beta\frac{\partial u}{\partial n}\right]_{\mathbf{X}\in\Gamma} &= \lim_{\mathbf{x}\to\mathbf{X},\mathbf{x}\in\Omega^+}\beta(\mathbf{x})\frac{\partial u(\mathbf{x})}{\partial n} - \lim_{\mathbf{x}\to\mathbf{X},\mathbf{x}\in\Omega^-}\beta(\mathbf{x})\frac{\partial u(\mathbf{x})}{\partial n} \\ &= \beta^+\frac{\partial u^+}{\partial n}\bigg|_{\mathbf{X}} - \beta^-\frac{\partial u^-}{\partial n}\bigg|_{\mathbf{X}} = [\beta u_n]_{\mathbf{X}}\,, \end{aligned}$$

and so on, $\frac{\partial u}{\partial n}=\nabla u\cdot\mathbf{n}=u_n$ is the directional derivative in the normal direction $\mathbf{n}$ with $|\mathbf{n}|=1$ pointing to the Ω^+ side. We use $\boldsymbol{\tau}$ as the tangential direction following the right-hand side rule. Thus $\frac{\partial u}{\partial\tau}=\nabla u\cdot\boldsymbol{\tau}=u_\tau$ is the tangential derivative of the solution u. When $w\equiv 0$ and $v\equiv 0$, the jump conditions are called *natural jump conditions.* Due to the discontinuity in β, the normal derivative u_n is typically discontinuous even with natural jump conditions.

1.1.3 Two-phase Stokes equations

Last, we consider the following two-phase Stokes equations

$$\nabla p = \nabla\cdot\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)+\mathbf{g}+\mathbf{F}, \quad \mathbf{x}\in\Omega, \tag{1.6}$$

$$\nabla\cdot\mathbf{u}=\mathbf{0}, \qquad \mathbf{x}\in\Omega, \tag{1.7}$$

$$\mathbf{u}\,|_{\partial\Omega}=\mathbf{u}_b, \tag{1.8}$$

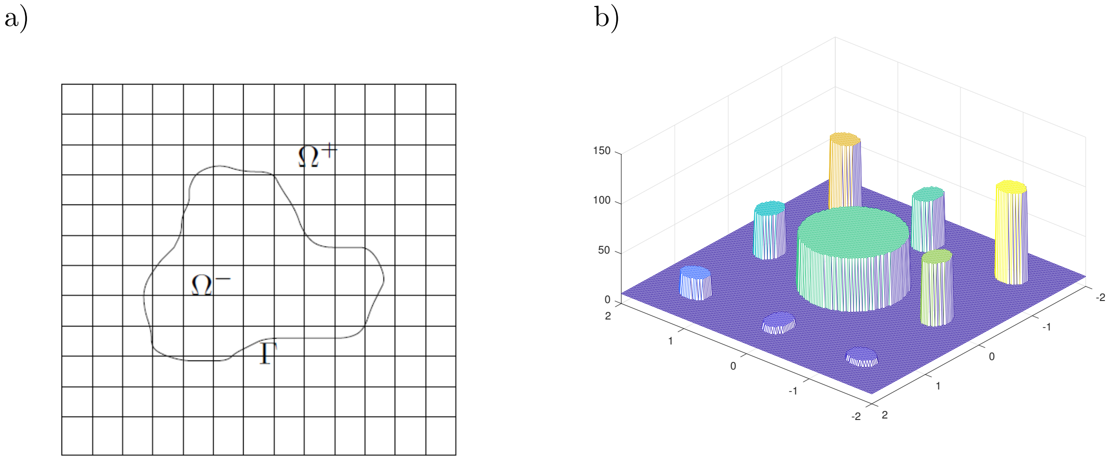

Figure 1.2: (a): A diagram of an arbitrary interface immersed in a uniform grid. (b): A diagram of the multiscale problem with multi-inclusions. The plot is the coefficient $\beta(x) \geq \beta_0 > 0$ which is a piecewise constant.

where $\mathbf{u} = (u, v)^T$ is the fluid velocity, p is the fluid pressure, μ is the fluid viscosity, $\mathbf{u}_b$ is a Dirichlet boundary condition for the velocity, $\mathbf{g} = (g_1, g_2)^T$ is the body force and $\mathbf{F}$ is a singular source defined as

$$\mathbf{F} = \int_\Gamma \mathbf{f}(s)\delta(\mathbf{x} - \mathbf{X}(s))ds. \tag{1.9}$$

Here $\mathbf{X}(s)$ is a parametric form of the interface Γ with s being the parameter, for example, the arc-length, $\delta(\cdot)$ is the Dirac delta function, and $\mathbf{f} = (f_1, f_2)^T$ is the density function of the surface force along the interface. The problem is illustrated in Figure 1.3, where the domain Ω is divided into two subdomains, Ω^+ and Ω^-, by the interface Γ.

For the regularity requirement of the problem, we also assume that $\mu \geq \mu_0 > 0$, $\Gamma \in C^2$, and $\mathbf{f} \in L^2$. The divergence free condition (5.2) combined with divergence theorem leads to the compatibility condition for $\mathbf{u}_b$, i.e.

$$\int_{\partial\Omega} \mathbf{u}_b \cdot \mathbf{n}_b = 0, \tag{1.10}$$

where $\mathbf{n}_b$ is the unit normal vector pointing to the outside of $\partial\Omega$.

1.2 A brief overview of the numerical methods for interface problems

Naturally, finite element methods can be and have been applied to solve the interface problem. It is well known that a second order accurate approximation to the solution of an interface

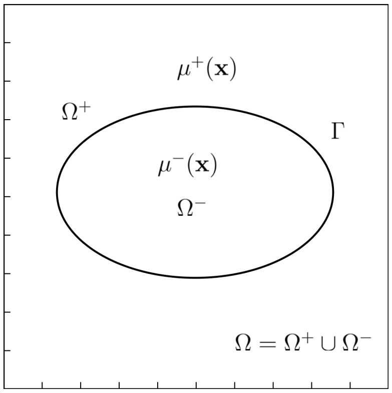

Figure 1.3: A diagram of a rectangular domain $\Omega = \Omega^+ \cup \Omega^-$ with an interface Γ. The coefficient $\mu(\mathbf{x})$ has a finite jump across the interface Γ.

problem with $w \equiv 0$ and $v \equiv 0$ can be generated by the Galerkin finite element method with the standard linear basis functions if the triangulation is aligned with the interface, that is, a body fitted mesh is used, see for example, [1, 5, 12, 69]. Some kind of posterior techniques or at least quadratic elements are needed in order to get second order accurate gradients from each side of the interface. The cost in the mesh generation coupled with unstructured linear solver is uncompetitive with the algorithm proposed in this thesis in our opinion. There are also quite a few finite element methods using Cartesian meshes. The immersed finite element (IFEM) was developed for 1D and 2D interface problems in [36] and [40], respectively. Since then, many IFEM methods and analyses have appeared in the literature, see for example, [14, 21], with applications in [48, 70]. The IFEM distinguishes from other FE methods in terms of degree of the freedom and structure of the coefficient matrix, for example, the extended finite element method (XFEM) in which enrichment functions are added near the interface [61]; unfitted finite element method based on the Nitsche's method in [20]. Related work in this direction can be found in [7, 30, 18, 23] and others. Note that, the methods developed in [24, 26] use a Petrov-Galerkin finite element discretization in which the non-conforming IFE space and the standard linear finite element space are used as the trial and test functional spaces, respectively. A partially penalty IFE method has been proposed in [49]. Another type of method is based on discontinuous Galerkin [68, 53] or weak Galerkin[65] methods with some penalties. In those methods, some parameters are chosen to achieve the optimal convergence. In general, discontinuous or weak Galerkin methods are flexible because there are more choices of the degree of freedom, which in turn implies these types of methods may be computationally more expensive. Those methods are usually better suited for hyperbolic problems and conservation laws.

Finite difference methods have also played a very important role in scientific computing and solving engineering problems. Advantages of finite difference methods based on Cartesian meshes include simplicity, ease to programming, and ability to utilize many existing fast solvers. Note that error estimates from finite element methods are based on integral forms which may not exactly predict the actual errors near the interface compared with that from finite difference methods that are based on the point-wise (L^∞) norm.

Many new finite difference methods based on a Cartesian mesh have been developed for interface problems, see for example, the matched interface and interface method (MIB) [73], the virtual node method [22, 3]. The MBI methods share the same basic idea with IIM to construct the fictitious grid values of the solution near the interface. The IIM achieved this by adding an additional correction term to the original stencil and the correction term is determined either by direct interpolation or the augmented variable. In contrast, the MBI methods assume the fictitious grid values are known and use them to interpolate the limiting values at the interface. The real fictitious grid values are determined when the limiting values satisfy all the jump conditions. This procedure does not involves deriving jump relations and strongly depends on the shape of the interface. And, as a result, the finite difference stencil at the grid point near the interface is not confined and well structured. The virtual node method adapts ideas from finite element methods. It tries to construct the basis functions at the grid points near the interface or irregular boundary to discretize the energy minimization form. The resulted coefficient matrix or the stiffness matrix is symmetric positive semi-definite. However, it only produces a first-order accurate gradient.

Most numerical methods for interface problems based on structured meshes are between first and second order accurate for the solution but the accuracy for the gradient is usually one order lower. Note that the gradient recovering techniques for example, [64, 71], are not suitable for structured meshes because of the arbitraries of the interface and the underlying mesh. The mixed finite element approach and a few other methods that can find accurate solutions and the gradient simultaneously in the entire domain often lead to a saddle problem and are computational expensive, which are not ideal choices if we are only interested in the accurate gradients near the interface or boundary.

For Poisson equations with singular source along an interface, it has been proved in [2] that both the computed solution and gradient are second order accurate by a factor of $\log h$ in the infinity norm. In [44], an augmented immersed interface method (AIIM) is proposed to solve the elliptic interface problems with piecewise constant coefficients. For more general elliptic interface problems, an augmented IIM [46] that provides both a second order accurate solution and the gradient on the interface has been developed and analyzed. In [45], a maximum principle preserving scheme is proposed for variable discontinuous coefficients. A quadratic optimization is used to determine the finite difference coefficients at grid points near the interface so that the coefficient matrix is diagonally dominant. Based on this maximum principle preserving scheme, a new direct IIM [10] without using augmented variables was proposed which produces both second order accurate solution and the gradient.

For heat equations with a constant heat conductivity but singular sources, a second order ADI method was developed In [41]. Since only the source term has a singularity, it is possible to modify only right-hand sides in the ADI method. The method is further generalized to discontinuous coefficient cases with a special, but less common flux jump condition $[u_n]$ (not $[\beta u_n]$) in [47]. The ADI approach is also applied to a non-linear convection-diffusion equation with a continuous coefficient in [51]. In [72], a matched interface and boundary (MIB) based ADI method is proposed for heat equations with a discontinuous coefficient, which is a step forward in this direction. However, the method is only first order accurate in time and does not have the same structure as that of the classical ADI method. In each directional sweep, one needs to solve a penta-diagonal system of equations if the grid line involves points from each side of the interface which makes the ADI method less attractive. The stability and error analysis are also open questions. A similar splitting technique for elliptic interface problems can be found in [13] in which the second order version would destroy the ADI structure and compromise the stability of the algorithm. Some other related research for heat and parabolic interface problems can be found in [4, 28]. It is challenging to develop ADI schemes for problems with a discontinuous coefficient because one cannot in general decouple the flux jump condition in coordinate directions.

The immersed interface method is first extended to solve Stokes flow with constant viscosity in [34]. Later on, an augmented IIM was developed for incompressible 2D Stokes flow with discontinuous viscosity in [39] and similar problems with a moving immersed boundary in [31]. All these methods decompose the incompressible Stokes equations into three separate Poisson equations with the same interface and the pressure is computed first to recover the velocity. Since the standard uniform grid is adapted, it is also challenging to design a numerical boundary condition for the pressure using a uniform grid. An alternative way to solve incompressible Stokes equations is to discretize momentum and continuity equations together as a whole system. In [63], an augmented IIM is combined with the Marker and Cell (MAC) scheme and an efficient algorithm is designed to solve two-phase incompressible Stokes equations. In [11], a direct IIM based on MAC scheme was proposed to solve the same problem. Both methods produce second order solution for both velocity and pressure.

1.3 Comparisons among finite difference based methods for interface problems

In this section, we use a simple one-dimensional elliptic interface problem to illustrate the differences among some finite difference based methods. Consider the following problem with a piecewise constant coefficient

$$\begin{aligned} &\beta u_{xx}(x) = f(x), \\ &[u] = w, \quad [\beta u_x] = v. \end{aligned}$$

The key to solve this problem is to derive finite difference schemes at irregular grid points, and more specifically schemes for u_{xx} term at x_i and x_{i+1}. Let us assume α is the interface (a point) located between the two irregular grid points x_i and x_{i+1}.

- The classical IIM [35] is based on Taylor expansion, and the method of undetermined coefficient. This approach assumes the finite difference scheme at x_i has the form

  $$u_{xx}(x_i) \approx c_1 u_{i-1} + c_2 u_i + c_3 u_{i+1} + c_4, \tag{1.11}$$

  where $c_1 - c_4$ are undetermined coefficients. Unlike the standard central difference stencil which apply Taylor expansion about the center point, the IIM expands u_{i-1}, u_i and u_{i+1} about the control point α, or the interface in the 1D case. This is because Taylor expansion requires the continuity and the solution in our problem has a jump across the interface. The standard method of undetermined coefficients then regards u, u_x and u_{xx} as a basis, and sets the coefficients of these basis equal zero to determine the scheme. In our case, we have two sets of basis to use, namely $\{u^-, u_x^-, u_{xx}^-\}$ and $\{u^+, u_x^+, u_{xx}^+\}$, which are related by jump relations. One can then choose either set of basis and apply the method of undetermined coefficients to derive the scheme.

  The classical IIM is a very general and well-studied approach. The finite difference stencil at a irregular grid point is confined involving three grid values in 1D and six to nine grid values in 2D problem. However, the original IIM approach does not generate second order accurate gradient.

- The MIB method [73] introduces a useful concept called fictitious value which is a hypothetical grid value extended from the opposite subdomain. For example, we know $u_i \in \Omega^-$ and $u_{i+1} \in \Omega^+$. Now let $\hat{u}_i$ and $\hat{u}_{i+1}$ be the fictitious values of u_i and u_{i+1} respectively. Then $\hat{u}_i$ is a hypothetical value of the solution at x_i in Ω^+ assuming we can extend the solution from Ω^+ smoothly to x_i. In the same way, $\hat{u}_{i+1}$ is a hypothetical value of the solution at x_{i+1} in Ω^-.

Now we have a set of grid values u_{i-1}, u_i, and $\hat{u}_{i+1}$ in Ω^- subdomain and another set of grid values $\hat{u}_i$, u_{i+1}, and u_{i+2} in Ω^+ subdomain. Since the function value is continuous in each set or subdomain, we can use standard method of undetermined coefficients to determine the finite difference scheme for limiting values at the interface. For example, we will know

$$\begin{aligned} u^- &\approx c_1^- u_{i-1} + c_2^- u_i + c_3^- \hat{u}_{i+1}, \\ u^+ &\approx c_1^+ \hat{u}_i + c_2^+ u_{i+1} + c_3^+ u_{i+2}, \\ u_x^- &\approx d_1^- u_{i-1} + d_2^- u_i + d_3^- \hat{u}_{i+1}, \\ u_x^+ &\approx d_1^+ \hat{u}_i + d_2^+ u_{i+1} + d_3^+ u_{i+2}, \end{aligned}$$

where all the coefficients $\{c_i^\pm\}$ and $\{d_i^\pm\}$ are known. These limiting values have to satisfy two given jump conditions, i.e.,

$$\begin{aligned} u^+ - u^- &= w, \\ \beta^+ u_x^+ - \beta^- u_x^- &= v. \end{aligned} \tag{1.12}$$

Here we have two equations (1.12) and two unknowns, $\hat{u}_i$ and $\hat{u}_{i+1}$, and hence can determine the schemes for these two fictitious values.

Unlike IIM, the MIB method does not derive the jump relations (more specifically jump relations for second and higher order derivatives), and this is also true for high-order dimensional problems. As a result, some interface information like curvature is not incorporated into the algorithm and it seems unable to produce second-order accurate gradient.

Another feature of MIB is that the form of its stencil at irregular grid point highly depends on the shape of the interface. This can also be observed in 1D problem that the two fictitious values $\hat{u}_i$ and $\hat{u}_{i+1}$ are coupled together, see (1.12), and their schemes involve four grid values instead of three in IIM. If we further assume another interface γ located between x_{i+1} and x_{i+2}, the scheme at irregular grid points will involve total six grid values. But in IIM, the scheme uses three grid values for all cases. This phenomenon becomes more common in two dimensional problems. The stencil is not confined and may involve quite a lot of grid values. As a result, the coefficient matrix is not well structured.

The advantage of MIB is that it can deal with nonsmooth interface with sharp points. This has not been done for any IIM approach.

- The new direct IIM [10] is based on classical IIM. The main idea is that the resulted coefficient matrix for interface problem should be close to the coefficient matrix generated by the same problem without the interface. To achieve this, we rewrite the finite difference scheme at irregular grid points as the standard central difference scheme plus a perturbation term, i.e.,

$$u_{xx}(x_i) \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + C, \tag{1.13}$$

where C is the correction term and has the formula

$$C = \frac{1}{h^2}\left([u] + (x_{i+1} - \alpha)[u_x] + \frac{(x_{i+1} - \alpha)^2}{2}[u_{xx}]\right). \tag{1.14}$$

The direct IIM assumes the interpolation scheme for the correction term C has the form

$$C = c_1 u_i + c_2 u_{i+1} + c_3, \tag{1.15}$$

where the coefficients $c_1 - c_3$ are determined by the method of undetermined coefficients using jump relations as illustrated for the classical IIM. One small difference here is that the direct IIM takes advantages of an additional jump relation from the PDE itself (the PDE provides two jump relations instead of one), which is the key to construct the interpolation scheme (1.15) and prove the convergence.

It will be more obvious to see why we prefer the formulation (1.13) over (1.11) in two or higher dimensional problems. In two-dimensional problem, usually the scheme at the irregular grid point involves nine gird values. The IIM determines these coefficients by calling SVD (singular value decomposition) solver which generates the minimum norm solution. If we use the formulation (1.11), the weights for the involved nine grid values have the same level of the magnitude. Notice that in (1.15), the coefficients $c1$ and c_2 are $O(\frac{1}{h^2})$ in general but actually far smaller than $\frac{1}{h^2}$ in magnitude in the numerical tests. So we can consider that the formulation (1.13) puts more weights on the five grid values used in the standard central difference stencil. In this case, the correction term C can be regarded as a small perturbation and the eigenvalues of the coefficient matrix still remain in the stability region as demonstrated in our numerical tests. This is also the main reason, as we believed, for gettiing second order accurate gradient.

Another benefit from using formulation (1.13) is that it is more suitable for convergence analysis especially for the augmented approaches.

- The augmented IIM [46] uses the formulation (1.13) to derive the finite difference scheme at the irregular grid points. The main idea of AIIM is to introduce an augmented variable, $[u_x] = q$, and solve a new model problem consist of the original PDE, the solution jump condition, and the jump condition for the augmented variable. The benefit of using augmented variable to solve a new model problem is that the formulations for the jump conditions become very simple and concise, especially for high dimensional problems, see (2.32)-(2.33). The solution computed from the new model is also the solution for the original model only if the flux jump condition is satisfied, i.e.,

$$\begin{aligned} v &= [\beta u_x] \\ &= \beta^+ u_x^+ - \beta^- u_x^- \\ &= [\beta] u_x^+ + \beta^- [u_x] \\ &= [\beta] u_x^- + \beta^- q. \end{aligned}$$

Combined with the system for the new model, we have

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ Q \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}.$$

Eliminating U gives the Schur complement equation for Q

$$(D - CA^{-1}B)Q = F_2 - CA^{-1}F_1, \quad \text{or} \quad SQ = F. \tag{1.16}$$

We solve for Q first and then use the computed augmented variable to recover the solution.

Compared with direct IIM, one can consider the augmented IIM derive the interpolation scheme for the correction term C (1.14) implicitly. This approach produces second order accurate gradient and have complete proof for the convergence for both the solution and the gradient. The disadvantage is that it involves solving an additional Schur complement system (1.16).

1.4 Outline of the thesis

The rest of the thesis is organized as follows

- In Chapter 2, we proposed a new augmented immersed interface method for general elliptic interface problems with variable coefficients that have finite jumps across a general interface, and non-homogeneous jump conditions. In Section 2.1, we develop the algorithm in one-dimension since it is easy to understand and explain followed by the convergence proof. In Section 2.3, we extend the algorithm to two dimensions followed again by the convergence analysis in Section 2.4. In Section 2.5, we present some two dimensional numerical examples.

- In Chapter 3, we proposed a new direct immersed interface method for general elliptic interface problems with a variable coefficient that have finite jumps across a general interface, and non-homogeneous jump conditions. In Section 3.1, we develop the algorithm in one-dimension since it is easy to understand and explain followed by the convergence proof. In Section 3.3, we extend the algorithm to two dimensions and prove its convergence. The stability and convergence of our method for general problems will be shown numerically in Section 3.4 by presenting some two dimensional numerical examples.

- In Chapter 4, we proposed two new ADI methods using some augmented variables to solve the heat equation with a interface. In Section 4.1, we explain the idea using augmented strategies and derive needed jump relations in coordinate directions. In Section 4.2, we develop two ADI methods using different augmented variables in detail. In Section 4.3, we explain how to discretize the augmented variable using the Schur complement system and other implementation details. The Schur complement matrix just needs to be generated initially for a fixed interface. In Section 4.4, we present numerical examples to validate our analysis and algorithms along with an application.

- A new direct immersed interface method for solving two-phase incompressible Stokes equations with an interface and a piecewise constant viscosity is proposed in Chapter 5. In Section 5.1, we derive the finite difference scheme, set up the linear system solver, and describe how to implement the algorithm. In Section 5.2, we will present some numerical examples and perform grid refinement and eigenvalue analysis to show the second-order convergence of our method. We conclude in the last section.

- Conclusions and future work are presented in Chapter 6.

Chapter 2

An Augmented Method for Accurate Solution and Gradient Computations for Elliptic Interface Problems

In this chapter, we develop an efficient augmented IIM to solve the elliptic interface problem

$$\nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \setminus \Gamma, \quad \Omega = \Omega^+ \cup \Omega^-, \tag{2.1}$$

$$[u](\mathbf{X}) = w(\mathbf{X}), \qquad [\beta u_n](\mathbf{X}) = v(\mathbf{X}), \qquad \mathbf{X} \in \Gamma, \tag{2.2}$$

in one and two space dimensions. where for example, $[u] = [u]_\Gamma(\mathbf{X}) = u^+(\mathbf{X}) - u^-(\mathbf{X})$ is the difference of the limiting values of $u(\mathbf{X})$ from Ω^+ and Ω^- sides, respectively, $u_n = \mathbf{n} \cdot \nabla u = \frac{\partial u}{\partial n}$ is the normal derivative of solution $u(\mathbf{X})$, and $\mathbf{n}(\mathbf{X})$ is the unit normal direction at a point $\mathbf{X}$ on the interface pointing to Ω^+ side, see Fig. 1.1 for an illustration. We use $\mathbf{x}$ to represent a point in the domain while $\mathbf{X}$ a point on the interface Γ. Since a finite difference discretization will be used, we assume that $f(\mathbf{x}) \in C(\Omega^\pm)$, $\beta(\mathbf{x}) \in C^1(\Omega^\pm)$, excluding Γ; and $\Gamma \in C^2$, $w \in C^2(\Gamma)$, $v \in C^1(\Gamma)$. All the parameters and $\frac{\partial \beta}{\partial x}$ and $\frac{\partial \beta}{\partial y}$ are assumed to be bounded. For the regularity requirement of the problem, we also assume that $\beta(\mathbf{x}) \geq \beta_0 > 0$ and $f(\mathbf{x}) \in C^\nu(\Omega \backslash \Gamma)$, for a constant $\nu > 0$ so that $u(\mathbf{x}) \in C^{2+\nu}(\Omega^\pm)$, see [17, 2]. For the error analysis, piecewise higher regularity assumptions are needed for the solution.

2.1 The one-dimensional algorithm

A model interface problem in one dimension has the following form

$$
\begin{aligned}
&(\beta u_x)_x = f(x), \qquad x \in (a,\alpha) \cup (\alpha, b), \\
&u(a) = u_a, \quad u(b) = u_b, \qquad [u]_\alpha = w, \quad [\beta u_x]_\alpha = v,
\end{aligned}
\tag{2.3}
$$

where $a < \alpha < b$ is an interface (a point). We assume that conditions for $\beta(x)$, $f(x)$ described in the introduction section hold with $\Omega^- = (a, \alpha)$ and $\Omega^+ = (\alpha, b)$. We will drop the subscripts α in the jump expressions such as $[u]_\alpha$ and $[\beta u_x]_\alpha$ and simply use $[u]$ and $[\beta u_x]$.

Let $x_i = a + ih$ be a uniform mesh with $h = (b-a)/N$ and $i = 0, 1, \cdots, N$. We define $q = [u_x]_\alpha$ as the augmented variable. Assume that $x_j \le \alpha < x_{j+1}$. We call x_j and x_{j+1} as *irregular* grid points while others are called *regular* grid points. The finite difference scheme at a regular grid point x_i, $i \ne j$ and $i \ne j+1$ can be written as

$$
\frac{\beta_{i-1/2}\, U_{i-1} - 2\bar{\beta}_i\, U_i + \beta_{i+1/2}\, U_{i+1}}{\bar{\beta}_i h^2} = \frac{f(x_i)}{\bar{\beta}_i},
\tag{2.4}
$$

where

$$
\beta_{i-1/2} = \beta(x_i - h/2), \quad \beta_{i+1/2} = \beta(x_i + h/2), \quad \bar{\beta}_i = \frac{\beta_{i-1/2} + \beta_{i+1/2}}{2}.
\tag{2.5}
$$

At the irregular grid points x_j and x_{j+1}, we use the following equivalent differential equation

$$
u_{xx} + \frac{\beta_x u_x}{\beta} = \frac{f}{\beta}.
\tag{2.6}
$$

This is one of the key ideas of the new augmented approach. In this way, we can obtain the second jump conditions $[u_{xx}]$ in terms of lower order jump conditions and derivatives of the solution.

If we know the jump $[u_x] = q$ in addition to the original jump conditions $[u]$ and $[\beta u_x]$, then we know the following jump relations

$$
\begin{aligned}
&[u] = w, \qquad [u_x] = q, \\
&[u_{xx}] = \left[\frac{f}{\beta}\right] - \frac{\beta_x^+ u_x^+}{\beta^+} + \frac{\beta_x^- u_x^-}{\beta^-} = \left[\frac{f}{\beta}\right] - \left[\frac{\beta_x}{\beta}\right] u_x^- - \frac{\beta_x^+}{\beta^+} q.
\end{aligned}
\tag{2.7}
$$

If $\beta_x(x_j)/\beta(x_j) \ge 0$, then the finite difference discretization at the irregular grid point x_j

can be written as

$$\frac{U_{j-1}-2U_j+U_{j+1}}{h^2}+C^{FD}(U_{j-1},U_j,U_{j+1})+\frac{\beta_x(x_j)}{\beta(x_j)}\left(\frac{U_{j+1}-U_j}{h}+C\right)=\frac{f(x_j)}{\beta(x_j)}, \tag{2.8}$$

where C is a correction term, see [38]

$$C=-\frac{[u]}{h}-\frac{(x_{j+1}-\alpha)[u_x]}{h}=-\frac{w}{h}-\frac{(x_{j+1}-\alpha)\,q}{h}, \tag{2.9}$$

and $C^{FD}(U_{j-1},U_j,U_{j+1})$ is part of the finite difference equation,

$$\begin{aligned} C^{FD}(U_{j-1},U_j,U_{j+1}) &= -\frac{[u]}{h^2}-\frac{(x_{j+1}-\alpha)[u_x]}{h^2}-\frac{(x_{j+1}-\alpha)^2[u_{xx}]}{2h^2} \\ &= -\frac{w}{h^2}-\frac{(x_{j+1}-\alpha)\,q}{h^2}-\frac{(x_{j+1}-\alpha)^2[u_{xx}]}{2h^2}, \end{aligned} \tag{2.10}$$

in which $[u_{xx}]$ is discretized by, see (2.7),

$$\begin{aligned} [u_{xx}] &= \left[\frac{f}{\beta}\right]-\left[\frac{\beta_x}{\beta}\right]u_x^- -\frac{\beta_x^+}{\beta^+}q \\ &\approx \begin{cases} \left[\frac{f}{\beta}\right]-\frac{\beta_x^+}{\beta^+}q-\left[\frac{\beta_x}{\beta}\right]\frac{U_j-U_{j-1}}{h} & \text{if } \left[\frac{\beta_x}{\beta}\right]\le 0, \\ \left[\frac{f}{\beta}\right]-\frac{\beta_x^+}{\beta^+}q-\left[\frac{\beta_x}{\beta}\right]\left(\frac{U_{j+1}-U_j}{h}+C\right) & \text{otherwise.} \end{cases} \end{aligned}$$

The case when $\beta_x(x_j)/\beta(x_j)<0$ can be treated in a similar way. We omit the details here. We can derive a similar finite difference scheme at the irregular grid point x_{j+1}. The finite difference scheme has the following properties.

- It is consistent. The local truncation errors at regular grid points are $O(h^2)$, and $O(h)$ at irregular grids points x_j and x_{j+1}.

- The finite difference scheme can be written as

  $$A_h\mathbf{U}+BQ=\mathbf{F}_1 \tag{2.11}$$

  where the coefficient matrix A_h is irreducible, tri-diagonal, and diagonally dominant, $\mathbf{U}$ is the column vector formed by the finite difference solution, and B is a column vector with at most two nonzero entries at j-th and $(j+1)$-th locations, Q is the approximate value of $q=[u_x]$. Note that A_h is invertible and the two component of F_j and F_{j+1} have been modified.

2.1.1 Discretization of the flux jump condition

Next we discuss the interpolation scheme to approximate the interface condition $[\beta u_x] = v$. First we re-write the jump condition as follows

$$\begin{aligned} [\beta u_x] = \beta^+ u_x^+ - \beta^- u_x^- &= \beta^+ (u_x^- + q) - \beta^- u_x^- \\ \Longrightarrow \qquad \frac{\beta^+ - \beta^-}{\beta^+} u_x^- + q &= \frac{v}{\beta^+}. \end{aligned} \tag{2.12}$$

This can be discretized as

$$\frac{\beta^+ - \beta^-}{\beta^+} \Big(\gamma_1 U_{j-1} + \gamma_2 U_j + \gamma_3 U_{j+1} + C_3 \Big) + q = \frac{v}{\beta^+}, \tag{2.13}$$

where γ_1, γ_2, γ_3, and the correction term C_3 are determined again using the idea of the IIM so that the interpolation scheme is a second order approximation of (2.12), that is,

$$\frac{\beta^+ - \beta^-}{\beta^+} \Big(\gamma_1 u(x_{j-1}) + \gamma_2 u(x_j) + \gamma_3 u(x_{j+1}) + C_3 \Big) + [u_x] - \frac{v}{\beta^+} = O(h^2).$$

In the matrix-vector form, the above equation can be written as

$$S\mathbf{U} + GQ = F_2 \tag{2.14}$$

where S is a row vector whose sum is zero.

We define the residual of the flux jump condition given an approximation Q as

$$R(Q) = S\mathbf{U} + GQ - F_2, \tag{2.15}$$

which is the discrete form of $r(q) = [\beta u_x] - v$. If we put the two equations (2.11) and (2.14) together, we get

$$\begin{bmatrix} A_h & B \\ S & G \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ Q \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ F_2 \end{bmatrix}. \tag{2.16}$$

Eliminating $\mathbf{U}$ in equation (2.16) gives the Schur complement equation for Q

$$(G - SA_h^{-1}B)Q = F_2 - SA_h^{-1}\mathbf{F}_1. \tag{2.17}$$

The equation (2.17) can be solved if the Schur complement is nonsingular. Once Q is computed, one can substitute it in equation (2.11) to solve for $\mathbf{U}$. The cost of computation in this process is to solve linear systems with the form $A_h x = b$ three times, $A_h^{-1}\mathbf{F}_1$, $A_h^{-1}BQ$, and finally (2.11). Since matrix A_h is tridiagonal and row diagonally dominant, the Thomas algorithm is guaranteed to be stable and the solution can be obtained in $O(N)$ operations.

2.2 Convergence analysis of the 1D algorithm

In this section, we demonstrate second order convergence of the solution globally and its first order derivative at the interface, as well as the augmented variable of the proposed new method. The proof is simpler than that in the two dimensional case but serves the purpose of understanding the tools used in the proof.

We use the following notations. We denote the errors as $\mathbf{E}^u = \mathbf{U} - \mathbf{u}$ with $\mathbf{E}_i^u = U_i - u(x_i)$ for the solution and $E^q = Q - q$ for the augmented variable, respectively, where $u(x_i)$ is the true solution at x_i. We use C to represent a generic error constant. We start with the analysis by assuming that the coefficient $\beta(x)$ is a piecewise constant, the domain is $(0,\ 1)$, and a Dirichlet boundary condition at the two end points for simplicity.

Theorem 2.1. *Assume that $\beta(x)$ is a piecewise constant and $u(x)$ is in piecewise C^4 excluding the interface α. If Q is a second order accurate approximation to q, i.e. $|E^q| \le Ch^2$, then we also have $\|\mathbf{E}^u\|_\infty \le Ch^2$.*

Proof: Let $\mathbf{T}^u$ be the local truncation error of system (2.11), that is,

$$A_h \mathbf{u} + Bq = \mathbf{F}_1 + \mathbf{T}^u, \tag{2.18}$$

where $\mathbf{u}$ is the vector formed by the true solution at the grid points x_i, q is the jump of the derivative of the solution $[u_x]$ across the interface α. Subtracting equation (2.18) from (2.11) yields

$$A_h \mathbf{E}^u = \tilde{\mathbf{F}}^u, \tag{2.19}$$

where $\tilde{\mathbf{F}}^u = -\mathbf{T}^u - BE^q$. Notice that $|\mathbf{T}_i^u| \le Ch^2$ and $B_i = 0$ at regular grid points while $|\mathbf{T}_j^u| \le Ch$ and $|\mathbf{T}_{j+1}^u| \le Ch$, and $B_j \sim O(\frac{1}{h})$, $B_{j+1} \sim O(\frac{1}{h})$. Since $|E^q| \le Ch^2$, we have $\tilde{F}_i^u \approx O(h^2)$ at regular points while $\tilde{F}_j^u \sim O(h)$, $\tilde{F}_{j+1}^u \sim O(h)$. Also when β is piecewise constant, the matrix A_h can be simplified as

$$A_h = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ & & 1 & -2 \end{bmatrix}.$$

From [33], we have

$$(A_h)^{-1}_{ij} = hG(x_i;x_j) = \begin{cases} h(x_j-1)x_i, & i=1,2,...,j, \\ h(x_i-1)x_j, & i=j,j+1,...,N-1, \end{cases} \tag{2.20}$$

where

$$G(x;\bar{x}) = \begin{cases} (\bar{x}-1)x, & x \le \bar{x}, \\ (x-1)\bar{x}, & x \ge \bar{x}, \end{cases}$$

is the Green's function that is the solution of the following

$$\begin{aligned} &\Delta_x\, G(x;\bar{x}) = \delta(x-\bar{x}), \qquad 0<x<1, \quad 0<\bar{x}<1, \\ &G(0;\bar{x}) = 0, \qquad G(1;\bar{x}) = 0. \end{aligned}$$

The global error of u then can be represented as

$$E^u_i = h\sum_{j=1}^{N-1} \tilde{F}^u_j G(x_i;x_j). \tag{2.21}$$

Since $0 \le G(x_i;x_j) \le 1$, we have the inequality

$$\begin{aligned} |E^u_i| \le \left| h\sum_{j=1}^{N-1} \tilde{F}^u_j \right| &\le h\left(|\tilde{F}^u_j| + |\tilde{F}^u_{j+1}| + \sum_{k=1}^{j-1} |\tilde{F}_k| + \sum_{k=j+2}^{N-1} |\tilde{F}_k| \right) \\ &\sim h\left(O(h) + (N-2)O(h^2)\right) \sim O(h^2), \end{aligned}$$

since $N \sim 1/h$. This shows that $\|E^u\|_\infty \le Ch^2$, hence the proof is completed. ■

Next, we show that the Schur complement system is non-singular.

Theorem 2.2. *With the same assumptions as in Theorem 2.1 and $\beta^- \neq \beta^+$, then the coefficient matrix (a number for the 1D problem) of the Schur complement is non-singular.*

Proof: Note that from (2.11), that is, $A_h\mathbf{U}(Q) + BQ = \mathbf{F}_1$, we have $A_h^{-1}BQ = A_h^{-1}\mathbf{F}_1 - \mathbf{U}(Q)$ and the Schur complement can be re-written as

$$\begin{aligned} \left(G - SA_h^{-1}B\right)Q &= GQ - SA_h^{-1}BQ = GQ - SA_h^{-1}\mathbf{F}_1 + S\mathbf{U}(Q) \\ &= (S\mathbf{U}(Q) + GQ) - (SA_h^{-1}\mathbf{F}_1 + G\cdot 0) \\ &= (S\mathbf{U}(Q) + GQ) - (S\mathbf{U}(0) + G\cdot 0) \\ &= R(Q) - R(0). \end{aligned}$$

If $Q \neq 0$ and $\beta^- \neq \beta^+$, then $R(Q) \neq R(0)$. For the one-dimensional problem, we have $\left(G - SA_h^{-1}B\right) = \left(G - SA_h^{-1}B\right)1 = R(1) - R(0) \neq 0$[1]. ■

Now we are ready to show that the augmented variable Q is also second order accurate.

Theorem 2.3. *With the same assumptions as in Theorem 2.1 and $\beta^- \neq \beta^+$, then we have $|E^q| = |Q - q| \leq Ch^2$.*

Proof: Similarly to the definition of the local truncation error $\mathbf{T}^u$, we define the local truncation T^q of q as

$$S\mathbf{u} + Gq = F_2 + T^q, \tag{2.22}$$

where $\mathbf{u}$ and q are defined as before. From (2.16), we know that

$$\begin{bmatrix} A & B \\ S & G \end{bmatrix} \begin{bmatrix} \mathbf{E}^u \\ E^q \end{bmatrix} = \begin{bmatrix} -\mathbf{T}^u \\ -T^q \end{bmatrix}. \tag{2.23}$$

Eliminating $\mathbf{E}^u$, we get the Schur complement system for E^q

$$(G - SA^{-1}B)E^q = -T^q + SA^{-1}\mathbf{T}^u. \tag{2.24}$$

We already know that the $(G - SA^{-1}B)$ is non-singular and $\|T^q\|_\infty \leq Ch^2$. The key is to show that $\|SA^{-1}\mathbf{T}^u\|_\infty \leq Ch^2$.

Let $\mathbf{b} = A^{-1}\mathbf{T}^u$, from the definition of the Green function in (2.20), we can write

$$b_i = h \sum_{l=1}^{N-1} \mathbf{T}_j^u \, G(x_i; x_l). \tag{2.25}$$

At first glance, it seems that $E^q \sim O(h)$ since the interpolation operator $\|S\|_\infty \sim \ 1/h$. Nevertheless, the following analysis shows that the terms of $O(1/h)$ are cancelled out to $O(1)$ and thus $E^q \sim O(h^2)$ is true. Let $\Delta_i = b_i - b_{i-1}$, $i = 2, ..., N-1$. Then we have (note that both $SA^{-1}\mathbf{T}^u$ and $S\,\mathbf{b}$ are scalars for the 1D problem)

$$\begin{aligned} SA^{-1}\mathbf{T}^u &= S\,\mathbf{b} = S_{j-1}b_{j-1} + S_j b_j + S_{j+1}b_{j+1} \\ &= S_{j-1}(b_j - \Delta_j) + S_j b_j + S_{j+1}(b_j + \Delta_{j+1}) \\ &= (S_{j-1} + S_j + S_{j+1})b_j - S_{j-1}\Delta_j + S_{j+1}\Delta_{j+1} \\ &= -S_{j-1}\Delta_j + S_{j+1}\Delta_{j+1}. \end{aligned}$$

Notice that the term b_j is cancelled out. This is because the interpolation operator is for the

[1] Note that some of the proof is similar to the contents in Section 6.1.2 in [38].

first order derivative of $u(x)$, and the consistency condition requires that $S_{j-1} + S_j + S_{j+1} = 0$. Now what is left to prove is that $\Delta_j \sim \Delta_{j+1} \sim O(h^3)$, which leads to $E^q \sim h^2$ Since $S_{j-1} \sim S_{j+1} \sim O(1/h)$. The final step of the proof is explained below.

$$\begin{aligned}
|\Delta_i| = |b_i - b_{i-1}| &= h \sum_{l=1}^{N-1} |\mathbf{T}_l^u| \cdot |G(x_i; x_l) - G(x_{i-1}; x_l)| \\
&\le h^2 \sum_{l \neq j, j+1}^{N-1} |\mathbf{T}_l^u| + h^2 \left(|T_j^u + |T_{j+1}^u| \right) \qquad \text{from the continuity of } G(x_i, x_l), \\
&\approx O(h^3).
\end{aligned}$$

This completes the proof. ■

As a result of Theorem 2.1-Theorem 2.3, we conclude that the solution $\mathbf{U}$ is also second order approximation to $\mathbf{u}$, which is summarized in the following theorem.

Theorem 2.4. *With the same assumptions as in Theorem 2.1 and $\beta^- \neq \beta^+$, and β is piecewise constant, then $\|\mathbf{E}^u\|_\infty = \|\mathbf{U} - \mathbf{u}\|_\infty \le Ch^2$.*

Proof: Since the Schur complement matrix is a constant independent of h and $|T^q| \le Ch^2$, and we just proved $|SA^{-1}\mathbf{T}^u| \le Ch^2$, from (2.17), we have the conclusion. ■

Not only do we obtain a second order accurate solution and the augmented variable, but also second order accurate derivative u_x^- and u_x^+ if the derivative is computed using the scheme (2.12), that is,

$$u_x^- = \frac{\beta^+}{\beta^+ - \beta^-} \left(\frac{v}{\beta^+} - q \right), \tag{2.26}$$

assuming that $\beta^- \neq \beta^+$. Since the computed Q is second order accurate, we immediately have the following theorem.

Theorem 2.5. *Assume β is piecewise constants and U_x^- is computed using the above formula with q being replaced by Q, the computed augmented variable. Then $|U_x^- - u_x^-| \le Ch^2$, where $u_x^- = \lim_{x \to \alpha-} \frac{du}{dx}(x)$.*

2.2.1 An example of the 1D Stefan problem

Our numerical experiments in one-dimension have confirmed our theoretical analysis that both the solution $\mathbf{U} \approx u(x)$ and the augmented variable $Q \approx [u_x]$ are second order accurate in the L^∞ norm. We show an example of a 1D Stefan problem, see for example, [8, 16], in which the

Table 2.1: A grid refinement analysis for the Stefan problem at the final time $t = 3$. The computed solution, the first order derivative $u_x^-(\alpha(t))$, and the free boundary $\alpha(t)$ all have average second order convergence.

N	$\|u - U\|_\infty$	r	$\|[u_x] - Q\|$	r	$\|\alpha - \mathcal{A}\|$	r
16	2.3160×10^{-2}		3.9232×10^{-4}		4.0734×10^{-2}	
32	6.4260×10^{-3}	1.8496	3.0046×10^{-4}	0.3849	1.1473×10^{-2}	1.8280
64	1.9403×10^{-3}	1.7276	1.2357×10^{-5}	4.6038	3.5210×10^{-3}	1.7042
128	4.8957×10^{-4}	1.9867	1.8056×10^{-7}	6.0967	8.8986×10^{-4}	1.9843
256	1.0044×10^{-4}	2.2852	7.3479×10^{-8}	1.2971	1.7986×10^{-4}	2.3067

free boundary $\alpha(t)$ is moving. The governing equations are

$$\tfrac{\partial u}{\partial t} = \tfrac{\partial^2 u}{\partial x^2}, \quad \text{for} \quad 0 < x < \alpha(t), \quad t > 0,$$

$$u(x,t) = 0, \quad \text{for} \quad x \geq \alpha(t), \quad t > 0,$$

where $\alpha(t)$ is subject to the Stefan condition

$$\frac{d\alpha}{dt}(t) = -\frac{\partial u}{\partial x}\left(\alpha(t), t\right), \qquad \alpha(0) = 0.$$

The boundary and initial conditions are

$$\frac{\partial u}{\partial x}(0,t) = f(t), \qquad u(\alpha(t),t) = 0, \qquad u(x,0) = 0.$$

The model is from [54]. We can find an analytic solution listed below,

$$u(x,t) = 1 - \frac{\operatorname{erf}\left(x/(2\sqrt{t})\right)}{\operatorname{erf}(\omega)}, \qquad \alpha(t) = 2\omega\sqrt{t},$$

where erf is the error function and ω is the solution of the transcendental equation $\sqrt{\pi}\,\omega \cdot \operatorname{erf}(\omega)e^{\omega^2} = 1$. The function $f(t, \alpha(t))$ is determined from the analytic solution.

We use a second order time splitting technique to solve the problem, that is, we solve the differential equation with $\alpha(t)$ fixed, then update the new location of $\alpha(t)$. The augmented equation now is the boundary condition at $\alpha(t)$. In Table 2.1, we show a grid refinement analysis of the errors in the solution at all grid points, and the free boundary $\alpha(t)$, and the first order derivative $u_x^-(\alpha(t))$, at the final time $t = 3$. We use lower case for the analytic solution, and upper case for the computed solutions. We observe that all of them have average second order convergence.

2.3 The algorithm for two dimensional problems

In this section, we present the algorithm for two dimensional problems. The key is the modification of the finite difference scheme at irregular grid points. We first discuss the interface relations using an equivalent representation of the interface problem.

2.3.1 The jump relations in the local coordinates

As explained in the introduction section, we re-write the elliptic interface problem near the interface as

$$\Delta u + \frac{\beta_x}{\beta} u_x + \frac{\beta_y}{\beta} u_y = \frac{f}{\beta}, \tag{2.27}$$

$$[u]_\Gamma = w, \qquad [u_n]_\Gamma = q, \tag{2.28}$$

where $q(\mathbf{X})$ is the augmented variable only defined along the interface Γ which should be chosen such that the flux jump condition

$$[\beta u_n]_\Gamma (\mathbf{X}) = v(\mathbf{X})$$

is satisfied. In this way, the Laplacian term Δu has been separated from $\beta(x, y)$ which makes the discretization easier with our proposed augmented method. This is one of the key ideas of the new method.

From the description of Section 3.1 in [35, 38], we restate some theoretical results on the reformulated elliptic interface problem (2.27)-(2.28). Assume that the interface in parametric form is

$$\Gamma = \Big\{ \ (X(s), Y(s)) \,, \quad X(s) \in C^2, \ Y(s) \in C^2 \Big\}, \tag{2.29}$$

where s is a parameter, for example, the arc-length. At a point of the interface (X, Y), the local coordinate system in the normal and tangential directions is defined as (see Figure 2.1 for an illustration),

$$\begin{cases} \xi = (x - X)\cos\theta + (y - Y)\sin\theta \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta, \end{cases} \tag{2.30}$$

where θ is the angle between the x-axis and the normal direction, pointing to the Ω^+ subdomain. Under such new coordinates system, the interface can be parameterized as

$$\xi = \chi(\eta) \qquad \text{with} \qquad \chi(0) = 0, \quad \chi'(0) = 0. \tag{2.31}$$

The curvature of the interface at (X, Y) is $\chi''(0)$.

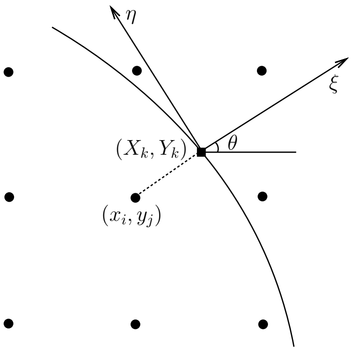

Figure 2.1: A diagram of an irregular grid point (x_i, y_j), its orthogonal projections on the interface (X_k, Y_k), and the local coordinates at (X_k, Y_k) in the normal and tangential directions.

If we know the jump in the solution $[u] = w$ and the normal derivative $[u_n] = q$ (not the original flux jump condition $[\beta u_n] = v$), then we can have the following jump relations at a point (X, Y) on the interface which is necessary to derive the accurate finite difference method.

Theorem 2.6. *For the elliptic interface problem (2.27)-(2.28), given $[u] = w$ and $[u_n] = q$, then at the interface, the following jump relations hold*

$$\begin{aligned}
&[u] = w, \qquad [u_\eta] = w', \qquad [u_\xi] = q, \\
&[u_{\eta\eta}] = -q\chi'' + w'', \qquad [u_{\xi\eta}] = w'\chi'' + q', \\
&[u_{\xi\xi}] = q\left(\chi'' - \frac{\beta_\xi^+}{\beta^+}\right) - w'' - \left[\frac{\beta_\xi}{\beta}\right]u_\xi^- - \left[\frac{\beta_\eta}{\beta}\right]u_\eta^- - \frac{\beta_\eta^+}{\beta^+}w' + \left[\frac{f}{\beta}\right]
\end{aligned} \tag{2.32}$$

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 $(\xi, \eta) = (0, 0)$.

Here we skip the derivation which is similar to those derived in equation (3.5) in Section 3.1 in [38] assuming that $[u] = w$ and $[\beta u_n] = v$ are given. Note also that we can express the jump conditions in terms of u^+, u_η^+, and u_ξ^+.

Once we have the jump relations in the local coordinates, we can get back the jump relations in the x- and y- directions according to (9.47) in [38]

$$\begin{aligned}
&[u_x] = [u_\xi]\cos\theta - [u_\eta]\sin\theta, \qquad [u_y] = [u_\xi]\sin\theta + [u_\eta]\cos\theta, \\
&[u_{xx}] = [u_{\xi\xi}]\cos^2\theta - 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\sin^2\theta, \\
&[u_{yy}] = [u_{\xi\xi}]\sin^2\theta + 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\cos^2\theta.
\end{aligned} \tag{2.33}$$

2.3.2 The finite difference scheme for the 2D problem

For simplification of discussion, we use a uniform a mesh

$$x_i = a + ih, \quad i = 0, 1, \cdots M; \qquad y_j = c + jh, \quad j = 0, 1, \cdots, N, \tag{2.34}$$

assuming $\Omega = (a, b) \times (c, d)$. The interface Γ is represented by the zero level set of a Lipschitz continuous function $\varphi(x, y)$, that is

$$\Gamma = \Big\{ \; (x, y), \quad \varphi(x, y) = 0, \quad (x, y) \in \Omega \Big\} . \tag{2.35}$$

In the neighborhood of the interface, we assume that $\varphi(x, y) \in C^2$. In implementation, the level set function is defined at the grid points as $\{\varphi_{ij}\}$ corresponding to $\varphi(x_i, y_j)$. At a grid point (x_i, y_j), we define

$$\varphi_{ij}^{max} \;=\; \max\left\{\varphi_{i-1,j}, \varphi_{ij}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}, \tag{2.36}$$

$$\varphi_{ij}^{min} \;=\; \min\left\{\varphi_{i-1,j}, \varphi_{ij}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}. \tag{2.37}$$

A grid point (x_i, y_j) is called *regular* if $\varphi_{ij}^{max}\varphi_{ij}^{min} > 0$, otherwise it is called *irregular*.

The set of orthogonal projections (X_k, Y_k), $k = 1, 2, \cdots, N_b$ of all irregular grid points on the interface from a particular side, say Ω^+ side, forms a discretization of the interface Γ. We refer the reader to Section 1.6.4 in [38] on how to find approximate orthogonal projections. Then the discrete augmented variable Q_k of the continuous one $q(s)$ is defined at those orthogonal projections. Given a discrete quantity along the interface, we can interpolate the quantity at the discrete points to get its value, the tangential derivative anywhere along the interface. For example, assume that (x_i, y_j) is an irregular grid point, and the interface cuts the grid line at (x_{ij}^*, y_j) corresponding to the orthogonal projection $\mathbf{X}_k = (X_k, Y_k)$. We need to get the values of Q and its tangential derivative at (x_{ij}^*, y_j). We first reconstruct the interface in the local coordinates as $\xi \approx C\eta^2 + D\eta^3$ with error $O(\eta^4)$. We refer the readers to Section 11.1.5 in [38] about how to find C and D. We then approximate $Q(\eta)$ as $Q(\eta) = Q_k + \omega_1\eta + \omega_2\eta^2$ locally with error $O(\eta^3)$ and $Q'(\eta) = \omega_1 + 2\omega_2\eta$ with error $O(\eta^2)$. The coefficient ω_1 and ω_2 are determined from Q values at the two closest orthogonal projections.

The finite difference scheme at a regular grid point

At a regular grid point (x_i, y_j), the finite difference scheme is the classic conservative one with the scaling

$$\frac{\beta_{i-1/2,j}U_{i-1,j} + \beta_{i+1/2,j}U_{i+1,j} + \beta_{i,j-1/2}U_{i,j-1} + \beta_{i,j+1/2}U_{i,j+1} - \bar{\beta}_{ij}U_{i,j}}{h^2\bar{\beta}_{ij}} = \frac{f_{ij}}{\bar{\beta}_{ij}} \tag{2.38}$$

where $f_{ij} = f(x_i, y_j)$, $\beta_{i-1/2,j} = \beta(x_i - h/2, y_j)$ and so on, and

$$\bar{\beta}_{ij} = \beta_{i-1/2,j} + \beta_{i+1/2,j} + \beta_{i,j-1/2} + \beta_{i,j+1/2}. \tag{2.39}$$

The finite difference scheme at an irregular grid point

We assume that we know the jump conditions $[u] = w$ and $[u_n] = q$, not the original flux jump condition $[\beta u_n] = v$. This makes it easier to derive an accurate and stable finite difference scheme. At an irregular grid point, we discretize the re-written equation (2.27) using a dimension by dimension approach, and an upwinding discretization for the first order derivative terms.

Let (x_i, y_j) be an irregular grid point. If the interface does not cut through the interval (x_{i-1}, x_{i+1}) along the line $y = y_j$, that is, (x_i, y_j) is regular in the x-direction, then the finite difference approximation for $(\beta u_x)_x$ before the scaling is

$$(\beta u_x)_x \approx \frac{\beta_{i-1/2,j}U_{i-1,j} + \beta_{i+1/2,j}U_{i+1,j} - (\beta_{i-1/2,j} + \beta_{i+1/2,j})U_{i,j}}{h^2}. \tag{2.40}$$

The final finite difference equation will be scaled in a similar way as those at regular grid points.

Now assume the interface cuts the grid line $y(x) = y_j$ in the interval (x_{i-1}, x_{i+1}), say at (x_{ij}^*, y_j), with $x_{ij}^* = x_i + \alpha_{ij}^x h$, $0 \le |\alpha_{ij}^x| < 1$. Without loss of generality, we assume that $(x_i, y_j) \in \Omega^-$. We discretize the reformulated equation (2.27), that is,

$$u_{xx}^- + u_{yy}^- + \frac{\beta_x^- u_x^-}{\beta^-} + \frac{\beta_y^- u_y^-}{\beta^-} = \frac{f^-}{\beta^-}, \tag{2.41}$$

where f^-, β^-, $\cdots$, are the limiting values at (x_{ij}^*, y_j) from Ω^- side. We use an upwinding scheme for the first order term $\beta_x^- u_x^- / \beta^-$, that is,

$$\frac{\beta_x^- u_x^-}{\beta^-} \approx \begin{cases} \dfrac{\beta_x^-}{\beta^-}\left(\dfrac{U_{i,j} - U_{i-1,j}}{h} + \dfrac{\tilde{C}_{ij}^x}{h}\right) & \text{if } \dfrac{\beta_x^-}{\beta^-} \le 0, \\ \dfrac{\beta_x^-}{\beta^-}\left(\dfrac{U_{i+1,j} - U_{i,j}}{h} + \dfrac{\bar{C}_{ij}^x}{h}\right) & \text{otherwise,} \end{cases} \tag{2.42}$$

where for example, $\tilde{C}_{ij}^x = 0$ if $(x_{i-1}, y_j) \in \Omega^-$. Otherwise we have

$$\tilde{C}_{ij}^x = -\Big([u] + [u_x]\left(1 - |\alpha_{ij}^x|\right) h \Big), \quad \text{where} \quad \alpha_{ij}^x = \frac{x_{ij}^* - x_i}{h}, \tag{2.43}$$

see Lemma 10.6 in [38] for the formulas of the correction, where the jumps again are defined at (x_{ij}^*, y_j). Similarly, for the second order term u_{xx}, the finite difference approximation for u_{xx}

can be written as

$$u_{xx}^{-}(x_i, y_j) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j} - C_{ij}^{x}}{h^2}, \tag{2.44}$$

where the correction term C_{ij}^{x} is

$$C_{ij}^{x} = [u] + [u_x]\left(1 - |\alpha_{ij}^{x}|\right) h + [u_{xx}] \frac{(1 - |\alpha_{ij}^{x}|)^2 h^2}{2}. \tag{2.45}$$

Approximating $[u_{xx}]$ and $[u_{yy}]$

Given $[u] = w$ and $[u_n] = q$ along the interface, from (2.33) and (2.32) we have the following

$$\begin{aligned}
[u_x] &= \cos\theta\,[u_\xi] - \sin\theta\,[u_\eta] = q\cos\theta - w'\sin\theta, \\
[u_y] &= \sin\theta\,[u_\xi] + \cos\theta\,[u_\eta] = q\sin\theta + w'\cos\theta, \\
[u_{xx}] &= \cos^2\theta\,[u_{\xi\xi}] - 2\sin\theta\cos\theta\,[u_{\xi\eta}] + \sin^2\theta\,[u_{\eta\eta}] \\
&= -2\sin\theta\cos\theta\,(w'\chi'' + q') + \sin^2\theta\,(-q\chi'' + w'') \\
&\qquad + \cos^2\theta \left\{ q\left(\chi'' - \frac{\beta_\xi^+}{\beta^+}\right) - w'' - \frac{\beta_\eta^+}{\beta^+} w' + \left[\frac{f}{\beta}\right] \right. \\
&\qquad \left. - \left[\frac{\beta_\xi}{\beta}\right]\left(\cos\theta\, u_x^- + \sin\theta\, u_y^-\right) - \left[\frac{\beta_\eta}{\beta}\right]\left(-\sin\theta\, u_x^- + \cos\theta\, u_y^-\right) \right\},
\end{aligned}$$

$$\begin{aligned}
[u_{yy}] &= \sin^2\theta\,[u_{\xi\xi}] + 2\sin\theta\cos\theta\,[u_{\xi\eta}] + \cos^2\theta\,[u_{\eta\eta}] \\
&= 2\sin\theta\cos\theta\left(w'\chi'' + q'\right) + \cos^2\theta\left(-q\chi'' + w''\right) \\
&\qquad + \sin^2\theta \left\{ q\left(\chi'' - \frac{\beta_\xi^+}{\beta^+}\right) - w'' - \frac{\beta_\eta^+}{\beta^+} w' + \left[\frac{f}{\beta}\right] \right. \\
&\qquad \left. - \left[\frac{\beta_\xi}{\beta}\right]\left(\cos\theta\, u_x^- + \sin\theta\, u_y^-\right) - \left[\frac{\beta_\eta}{\beta}\right]\left(-\sin\theta u_x^- + \cos\theta\, u_y^-\right) \right\},
\end{aligned}$$

where w' and q' are the first order, and w'' is the second order, derivatives along the interface, respectively. In the derivation above, we have used the following formulas

$$u_\xi = u_x\cos\theta + u_y\sin\theta, \qquad u_y = -u_x\sin\theta + u_y\cos\theta. \tag{2.46}$$

Most of terms in $[u_{xx}]$ and $[u_{yy}]$ are computable except terms of u^-, u_x^- and u_y^-. Note that these functions are defined on the interface. Using Taylor expansion, we have $u^-(X,Y) = u^-(x_i, y_j) + O(h)$, $u_x^-(X,Y) = u_x^-(x_i, y_j) + O(h)$ and $u_y^-(X,Y) = u_y^-(x_i, y_j) + O(h)$. We simply replace u^- with U_{ij} and treat u_x^- and u_y^- using the upwinding scheme to increase the diagonal dominance of the resulting linear system of finite difference equations. For example, for the terms containing u_x^- in $[u_{xx}]$ we use

$$\left(\left[\frac{\beta_\eta}{\beta}\right]\sin\theta-\left[\frac{\beta_\xi}{\beta}\right]\cos\theta\right)\cos^2\theta\, u_x^- = \begin{cases} A_{tmp}\left(\dfrac{U_{i+1,j}-U_{i,j}}{h}+\dfrac{\tilde{C}_{ij}^x}{h}\right) & \text{if } A_{tmp}\ge 0,\\ A_{tmp}\left(\dfrac{U_{i,j}-U_{i-1,j}}{h}+\dfrac{\bar{C}_{ij}^x}{h}\right) & \text{otherwise,}\end{cases}$$

$$\text{where}\quad A_{tmp}=\left(\left[\frac{\beta_\eta}{\beta}\right]\sin\theta-\left[\frac{\beta_\xi}{\beta}\right]\cos\theta\right)\cos^2\theta,$$

and once again, for example, $\tilde{C}_{ij}^x=0$ if $(x_{i-1},y_j)\in\Omega^-$. Otherwise

$$\tilde{C}_{ij}^x = -\Big([u]+[u_x]\left(1-|\alpha_{ij}^x|\right)h\Big). \tag{2.47}$$

The linear system of finite difference equations can be written as

$$A_h\mathbf{U}+B\mathbf{Q}=\mathbf{F}_1, \tag{2.48}$$

where $\mathbf{U}$ is the vector formed by the finite difference approximation $\{U_{ij}\}$ to the solution $\{u(x_i,y_j)\}$, $\mathbf{Q}$ is the vector formed by the discrete augmented variable $\{Q_k\}$ to the augmented variable $\{\,[\,\frac{\partial u}{\partial n}(X_k,Y_k)\,]\,\}$, F_1 is the modified right hand side, B is a sparse matrix corresponding to the correction terms for the $[u_n]$ term.

Remark 2.1.

- *The finite difference stencil is still the standard five-point centered stencil. This is different from the maximum principal preserving scheme [45] in which the finite difference stencil is a nine-point one.*
- *A_h is diagonally dominant and invertible. No optimization is needed compared to [45] because we assume that $[u_n]$ is given instead of $[\beta u_n]$, which makes it easier to discretize the interface problem. The trade-off is that we also need to solve for the augmented variable.*

2.3.3 Discretizing the flux jump condition

At every approximate orthogonal projection of all irregular grid points on the interface, we use the same least squares interpolation described in Section 4 in [44] to interpolate the flux jump condition $[\beta u_n]=v$.

At one orthogonal projection $\mathbf{X}_k=(X_k,Y_k)$ corresponding to an irregular grid point (x_i,y_j), the second order accurate least squares interpolation scheme approximating $[\beta u_n]=v$ can be written as

$$\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \gamma_{ij}U_{ij} + L_k\left(\beta(\mathbf{x}), \mathbf{W}, \mathbf{Q}, \mathbf{V}\right) = 0 \tag{2.49}$$

where δ_h is a parameter of $2h \sim 3h$, L_k stands for a linear relation of its augmenters, the discrete form of $w(\mathbf{X})$, $q(\mathbf{X})$, and $v(\mathbf{X})$. The consistency condition requires that

$$\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \gamma_{ij} = 0. \tag{2.50}$$

Note that the interpolation coefficients should depend on the index k as well, we omit it for simplicity of notations.

In matrix vector form, the interpolation at all projections of irregular grid points from one particular side can be written as

$$S\mathbf{U} + G\mathbf{Q} = \mathbf{F}_2, \tag{2.51}$$

for some sparse matrices S and G. If we put (2.48) and (2.51) together we get

$$\begin{bmatrix} A_h & B \\ S & G \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{2.52}$$

Eliminating U in equation (2.52) gives the Schur complement equation for $\mathbf{Q}$

$$(G - SA_h^{-1}B)\mathbf{Q} = \mathbf{F}_2 - SA_h^{-1}\mathbf{F}_1, \quad \text{or} \quad A_h^{schur}\mathbf{Q} = \mathbf{F}^{schur}. \tag{2.53}$$

We use the GMRES iterative method to solve the Schur complement system and do not explicitly form the matrices A_h, B, S, G, and A_h^{schur}. The matrix and vector multiplication $A_h^{schur}\mathbf{Q}$ needed for the GMRES iteration involves two consecutive steps: the first step is to solve the interface problem (2.48) given $\mathbf{Q}$; the second step is to find the residual of the flux jump condition, that is, $R(\mathbf{Q}) = [\beta(\mathbf{X})\mathbf{U}_n(\mathbf{Q})] - \mathbf{V}$. We refer the reader to Section 5.1 in [44] for the details.

2.3.4 Computing the gradient on the interface

At one orthogonal projection $\mathbf{X}_k$ corresponding to an irregular grid point (x_i, y_j), we use a similar (two-sided SVD) interpolation to approximate the normal derivative at $\mathbf{X}_k$ from Ω^-

side

$$u_n^-(\mathbf{X}_k) = \sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \widetilde{\gamma_{ij}}\, U_{ij} + \widetilde{L_k}\ (\beta(\mathbf{x}), \mathbf{W}, \mathbf{Q}, \mathbf{V}) \tag{2.54}$$

to get one of $u_n^-(\mathbf{X}_k)$ or $u_n^+(\mathbf{X}_k)$, then use $q(\mathbf{X}_k)$ to get the other, say $u_n^+(\mathbf{X}_k) = q(\mathbf{X}_k)+u_n^-(\mathbf{X}_k)$. The linear system of equations has the same coefficient matrix as that in (2.49) for γ_{ij}'s, so there is almost no additional cost. The term $\widetilde{L_k}\ (\beta(\mathbf{x}), \mathbf{W}, \mathbf{Q}, \mathbf{V})$ is again a correction term due to the jumps in the involved quantities.

If needed, at a grid point, the partial derivatives u_x and u_y can be calculated using the standard 3-point central finite difference formula with (at an irregular grid point) or without (at a regular grid point) a correction term. Beadle and Layton [2] have shown that the computed derivatives using IIM are second order accurate in the L^∞ norm at all grid points.

2.3.5 A new preconditioning strategy

The number of GMRES iterations grows linearly with the mesh size N if there is no preconditioning technique employed. The preconditioning technique proposed in [44] works well for a piecewise constant coefficient but not for a variable coefficient. The idea of the new preconditioning technique is more like a diagonal preconditioning technique for the Schur complement. At an orthogonal projection $\mathbf{X}_k = (X_k, Y_k)$ where the augmented variable is defined, we use the re-scaled residual of the flux jump condition

$$R^{rescaled}(\mathbf{X}_k) = \frac{\beta^+(\mathbf{X}_k)U_n^+(\mathbf{X}_k) - \beta^-(\mathbf{X}_k)U_n^-(\mathbf{X}_k) - v(\mathbf{X}_k)}{\bar{\beta}(\mathbf{X}_k)}, \tag{2.55}$$

where $\bar{\beta}(\mathbf{X}_k) = (\beta^-(\mathbf{X}_k) + \beta^+(\mathbf{X}_k))/2$, to discretize the flux jump condition.

2.4 Convergence proof for the 2D problems

In this section, we provide a convergence proof for two dimensional problems. For simplicity, we assume that a Dirichlet boundary condition is prescribed along the boundary $\partial\Omega$. We use $\mathbf{U}$ and $\mathbf{u}$ to represent the vectors of approximate and exact solution at grid points; $\mathbf{T}^u$ and $\mathbf{E}^u = \mathbf{U}-\mathbf{u}$ are the vectors of the local truncation errors and the global error. We have $\mathbf{E}^u|_{\partial\Omega_h} = \mathbf{0}$ for the values at grid points on the boundary. Similarly, we define $\mathbf{T}^q$ and $\mathbf{E}^q = \mathbf{Q} - \mathbf{q}$ as the vectors of the local truncation error and the global error for the augmented variable. According to the definition, we have

$$A_h\mathbf{U} + B\mathbf{Q} = \mathbf{F}_1, \qquad A_h\mathbf{E}^u + B\mathbf{E}^q = \mathbf{T}^u, \tag{2.56}$$

$$S\mathbf{U} + G\mathbf{Q} = \mathbf{F}_2, \qquad S\mathbf{E}^u + G\mathbf{E}^q = \mathbf{T}^q, \tag{2.57}$$

where the local truncation error vector $\mathbf{T}^u$ is defined as $\mathbf{T}^u = \mathbf{F}_1 - A_h\mathbf{u} - B\mathbf{q}$ and so on.

Lemma 2.1. *Assume that $u(\mathbf{x})$ is in piecewise $C^4(\Omega \setminus \Gamma)$ excluding the interface Γ. If the augmented variable is a second order approximation to $[\frac{\partial u}{\partial n}(\mathbf{X})]$, that is, $\|\mathbf{E}^q\|_\infty \le Ch^2$, then the computed solution of the finite difference equations (2.48) is also second order accurate, that is, $\|\mathbf{E}\|_\infty \le Ch^2$.*

Proof: From the construction of the numerical scheme we know that a component of $B\mathbf{E}^q$ is zero at a regular grid point $\mathbf{x}_{ij}$ and bounded by Ch at an irregular grid point $\mathbf{x}_{ij}$ since $\|\mathbf{E}^q\|_\infty \le Ch^2$ as one of the conditions in the theorem. Note that A_h is an M-matrix and $A_h\mathbf{E}^u = -B\mathbf{E}^q + \mathbf{T}^u$ that is bounded by Ch^2 at regular grid points and Ch at irregular grid points. From Theorem 3.3 in [38] or the convergence analysis of IIM in [2], we conclude that the global error is bounded by Ch^2. Also from [58, 59], the partial derivatives using the IIM is also second order accurate. ■

The next step is to show that the computed augmented variable is also second order accurate by a factor of $\log h$. In this case, we first assume that the coefficient is piecewise constant so that we can apply some theoretical results from [37].

Theorem 2.7. *Assume that $u(\mathbf{x})$ is in piecewise $C^4(\Omega\setminus\Gamma)$ excluding the interface Γ, and the coefficient $\beta(\mathbf{x})$ is a piecewise constant β^- and β^+, then computed augmented variable is second order accurate by a fact of $|\log h|$, that is $\|\mathbf{E}^q\|_\infty \le Ch^2|\log h|$.*

Proof: From (2.56)-(2.57), we have

$$\left(G - SA_h^{-1}B\right)\mathbf{E}^q = -\mathbf{T}^q + SA_h^{-1}\mathbf{T}^u. \tag{2.58}$$

Note that solvability of the above linear system has been shown in [44]. We first prove that the right hand side above is bounded by Ch^2. Since the interpolation for the flux jump condition is a second order one, we have $\|\mathbf{T}^q\|_\infty \le Ch^2$. For the second term, from the interpolation scheme in (2.49), we consider one component and carry out the derivation

$$\begin{aligned}\left(SA_h^{-1}\mathbf{T}^u\right)_k &= \sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \gamma_{ij}\left(A_h^{-1}\mathbf{T}^u\right)_{ij} \\ &= \sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \gamma_{i,j}\sum_{l,\,r} G^h(\mathbf{x}_{lr},\mathbf{x}_{ij})\,h^2\,\tau_{lr} \\ &= \sum_{l,\,r} h^2\,\tau_{lr}\left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h}\gamma_{i,j}G^h(\mathbf{x}_{lr},\mathbf{x}_{ij})\right),\end{aligned} \tag{2.59}$$

where $G^h(\mathbf{x}_{lr},\mathbf{x}_{ij})$ is the discrete Green function defined as

$$\mathbf{G}^h\left(\mathbf{x}_{ij},\mathbf{x}_{lm}\right)=\left(A_h^{-1}\mathbf{e}_{lm}\frac{1}{h^2}\right)_{ij}, \qquad \mathbf{G}^h\left(\partial\Omega_h,\mathbf{x}_{lm}\right)=0, \tag{2.60}$$

where $\mathbf{e}_{lm}$ is the unit grid function whose values are zero at all grid points except at $\mathbf{x}_{lm} = (x_l, y_m)$ where its component is $e_{lm} = 1$, see for example [19]. Note that in the neighborhood of $|\mathbf{x}_{ij} - \mathbf{X}_k| \le \delta_h$, the points involved in the interpolation is close to $\mathbf{X}_k$, we can continue to derive

$$\begin{aligned}\left(SA_h^{-1}\mathbf{T}^u\right)_k &= \sum_{l,\,r} h^2\,\tau_{lr}\left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h}\gamma_{i,j}\left(G_I^h(\mathbf{x}_{lr},\mathbf{X}_k)+h\frac{G^h(\mathbf{x}_{lr},\mathbf{x}_{ij})-G_I^h(\mathbf{x}_{lr},\mathbf{X}_k)}{h}\right)\right) \\ &= \sum_{l,\,r} h^2\,\tau_{lr}\left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h}\gamma_{i,j}G_I^h(\mathbf{x}_{lr},\mathbf{X}_k)\right) \\ &\qquad + \sum_{l,\,r} h^3\,\tau_{lr}\left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h}\gamma_{i,j}\left(\frac{\partial G_I^h(\mathbf{x}_{lr},\mathbf{X}_k)}{\partial\mathbf{x}}\right)+O(h)\right) \\ &= \sum_{l,\,r} h^3\,\tau_{lr}\left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h}\gamma_{i,j}\left(\frac{\partial G_I^h(\mathbf{x}_{lr},\mathbf{X}_k)}{\partial\mathbf{x}}\right)+O(h)\right).\end{aligned}$$

The first term in the top line above is zero due to the consistency of the interpolation scheme for the flux jump condition. We have $|\tau_{lr}| \le Ch^2$ and at regular grid points, and $|\tau_{lr}| \le Ch$ at irregular grid points, from the estimate of $\frac{\partial G_I^h}{\partial \mathbf{x}}$ (3.16) in [37] we further derive

$$
\begin{aligned}
\left|\left(SA_h^{-1}\mathbf{T}^u\right)_k\right| &\le \sum_{l,\,r} h^3\,|\tau_{lr}| \left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} |\gamma_{ij}| \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) + O(h)\right) \\
&\le \sum_{l,\,r,\Omega_h^{reg}} h^3\,|\tau_{lr}| \left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} |\gamma_{ij}| \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) + O(h)\right) \\
&\qquad + \sum_{l,\,r,\Omega_h^{irr}} h^3\,|\tau_{lr}| \left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} |\gamma_{ij}| \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) + O(h)\right) \\
&\le \sum_{l,\,r,\Omega_h^{reg}} h^4 \left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) + O(h)\right) \\
&\qquad + \sum_{l,\,r,\Omega_h^{irr}} h^3 \left(\sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) + O(h)\right) \\
&\le h^2 \sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \left(\sum_{l,\,r,\Omega_h^{reg}} \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) h^2 + O(h^3)\right) \\
&\qquad + h^2 \sum_{|\mathbf{x}_{ij}-\mathbf{X}_k|\le\delta_h} \left(\sum_{l,\,r,\Omega_h^{irr}} \left(\frac{C}{(\|\mathbf{x}_{lr}-\mathbf{X}_k\|_2+h)}\right) h + O(h^3)\right) \\
&\le Ch^2|\log h| + Ch^2,
\end{aligned}
$$

where Ω_h^{irr} and Ω_h^{reg} are the sets of all irregular and regular grid points, respectively. In the derivation above we have used the facts that $|\gamma_{ij}| \sim 1/h$, $|\tau_{lr}| \le Ch^2$ at regular grid points and $|\tau_{lr}| \le Ch$ at irregular grid points, respectively. We have also used the estimate of the Riemann sum for the double integral $\iint 1/(x^2+y^2+h)dxdy \le C|\log h|$. Note also that the total number of regular grid points is $O(1/h^2)$ while the total number of irregular grid points is $O(1/h)$. It has been shown that Schur complement matrix A_h^{schur} is non-singular, thus we have $\|A_h^{schur}\mathbf{E}^q\|_\infty \le Ch^2|\log h|$.

We have shown that the right hand side for the error of the augmented variable has the order of $h^2\log h$. From Section 6.1.2 in [38], we know that the left hand side of (2.58) is

$$
A_h^{schur}\mathbf{E}^q = \left[\beta\frac{\partial\tilde{\mathbf{U}}}{\partial n}\left(\mathbf{E}^q\right)\right] - \left[\beta\frac{\partial\tilde{\mathbf{U}}}{\partial n}\left(\mathbf{0}\right)\right], \tag{2.61}
$$

where $\tilde{\mathbf{U}}(\mathbf{E}^q)$ can be regarded as the solution of the numerical method applied to the following problem

$$\nabla \cdot (\beta \nabla \tilde{u}) = \mathbf{T}_I^u(\mathbf{x}); \qquad \tilde{u}|_{\partial\Omega} = 0, \tag{2.62}$$

$$[\tilde{u}]_\Gamma = 0, \qquad \left[\beta \frac{\partial \tilde{u}}{\partial n}\right]_\Gamma = \mathbf{T}_I^q(\mathbf{X}), \tag{2.63}$$

where $\mathbf{T}_I^u(\mathbf{x}) \in C$ is an interpolation function of $\mathbf{T}^u$ on the entire domain while $\mathbf{T}_I^q(\mathbf{X}) \in C$ is an interpolation function of $\mathbf{T}^q$ along the interface. From the maximum principle, we know that $|\tilde{u}| \leq Ch^2$ and $|\frac{\partial \tilde{u}^\pm}{\partial n}| \leq Ch^2$. Therefore the second term in (2.61) is bounded by Ch^2. Thus we have

$$\begin{aligned} A_h^{schur}\mathbf{E}^q &= \left[\beta \frac{\partial \tilde{\mathbf{U}}}{\partial n}(\mathbf{E}^q)\right] = \beta^+ \frac{\partial \tilde{\mathbf{U}}}{\partial n}^+(\mathbf{E}^q) - \beta^- \frac{\partial \tilde{\mathbf{U}}}{\partial n}^-(\mathbf{E}^q) + O(h^2) \\ &= \beta^+\mathbf{E}^q - [\beta]\frac{\partial \tilde{\mathbf{U}}}{\partial n}^-(\mathbf{E}^q) + O(h^2), \end{aligned}$$

since β is a piecewise constant that has been divided by, from [2], we know that the solution and the derivative are both second order accurate when the IIM is applied, which implies that $\|\frac{\partial \tilde{\mathbf{U}}}{\partial n}^-(\mathbf{E}^q)\|_\infty \leq Ch^2$. We have already proved that $\|A_h^{schur}\mathbf{E}^q\|_\infty \leq Ch^2$, this leads to $\|\mathbf{E}^q\|_\infty \leq Ch^2$. ■

Remark 2.2. *In the preconditioning strategy, we can write, for example, equation (6.24) in [38],*

$$\frac{\partial \tilde{\mathbf{U}}}{\partial n}^-(\mathbf{E}^q) = \gamma \left[\beta \frac{\partial \tilde{\mathbf{U}}}{\partial n}(\mathbf{E}^q)\right] + F_\Gamma + O(h^2), \tag{2.64}$$

where γ is a constant, and F_Γ is a vector, then we have

$$\begin{aligned} A_h^{schur}\mathbf{Q} &= \left[\beta \frac{\partial \mathbf{U}}{\partial n}(\mathbf{Q})\right] - \left[\beta \frac{\partial \mathbf{U}}{\partial n}(\mathbf{0})\right] = \beta^+ \frac{\partial \mathbf{U}}{\partial n}^+(\mathbf{Q}) - \beta^- \frac{\partial \mathbf{U}}{\partial n}^-(\mathbf{Q}) - \left[\beta \frac{\partial \mathbf{U}}{\partial n}(\mathbf{0})\right] \\ &= \left(\beta^+ - \beta^-\gamma\right)\mathbf{Q} - \beta^- F_\Gamma - \left[\beta \frac{\partial \mathbf{U}}{\partial n}(\mathbf{0})\right] + O(h^2), \end{aligned}$$

which means that the Schur complement matrix is nearly a diagonal. This may explain why the number of the GMRES iterations is independent of the mesh size and the jump in β. For variable coefficient $\beta(\mathbf{x})$, with the new preconditioning strategy, we would have

$$A_h^{schur}\mathbf{Q} \;=\; = D(\bar{\beta}(\mathbf{x}))\mathbf{Q} + \tilde{\mathbf{F}}_\Gamma + O(h^2),$$

where $D(\bar{\beta}(\mathbf{x}))$ is a diagonal matrix whose entries are $(\beta_k^+ - \beta_k^-)/\bar{\beta}_k$, $\bar{\beta}_k = (\beta_k^+ + \beta_k^-)/2$.

Remark 2.3. *While the proof above is for piecewise constant coefficients, the conclusion is also true, or at least asymptotically in terms of h, for variable coefficient $\beta(\mathbf{x}) \geq \beta_0 > 0$ assuming that $\beta(\mathbf{x}) \in C^{\infty}(\Omega^{\pm})$ since those terms involved are lower order terms of h. This is because the coefficient matrix $A_h(\beta) = A_h(I + B_h)$ and $\|B_h\| \to 0$ as $h \to 0$, where A_h is the discrete Laplacian. This is another advantage of using the reformulated PDE.*

2.5 Numerical examples

We present a variety of numerical experiments to show the performance of the new augmented method for accurate solutions and its first order gradient at the interface. All the experiments are computed with double precision and are performed on a desktop computer with Pentium(R) Dual-Core CPU, 2.59 GHz, 4GB memory. We also list the CPU time (s) in tables. We present errors in L^{∞} norms and estimate the convergence order using

$$r = \frac{1}{\log 2} \log \frac{\|\mathbf{E}_{2h}\|_{\infty}}{\|\mathbf{E}_h\|_{\infty}}. \tag{2.65}$$

The tolerance of the GMRES iteration is set to be 10^{-6} and the initial value is set to be $\mathbf{0}$ in all computations. In all tables listed in this section, we use "Iter" to represent the number of GMRES iterations, "N_b" the number of control points, "N" the number of the grid lines in each direction of the rectangular domain and "CPU(s)" the run time in seconds.

Example 2.1.

$$u(\mathbf{x}) = \begin{cases} \sin(x+y) & in\ \Omega^-, \\ \log(x^2+y^2) & in\ \Omega^+, \end{cases} \qquad \beta(\mathbf{x}) = \begin{cases} \sin(x+y)+2 & in\ \Omega^-, \\ \cos(x+y)+2 & in\ \Omega^+, \end{cases} \tag{2.66}$$

where the interface is the zero level set of $\varphi(x,y) = \sqrt{x^2+y^2} - 0.5$, and $\Omega = [-1,1] \times [-1,1]$. The source term $f(\mathbf{x})$, and the interface jump conditions: $[u]$ and $[\beta u_n]$ are derived from the exact solution.

This is an example with a genuine piecewise smooth non-linear solution, and a variable coefficient with a variable discontinuity along the interface. We present a grid refinement analysis in Table 2.2. The second column is the maximum error of the solution while the third column is the approximate convergence order. The fourth and sixth column are the errors of the normal derivatives at the interface from Ω^- and Ω^+ sides, respectively. The fifth and seventh columns are the approximate convergence order of the computed normal derivative. The eighth column is the number of is the number of GMRES iterations, and the last but one is orthogonal projections of irregular grid points from Ω^+ side. The last column is the total CPU time in seconds. We

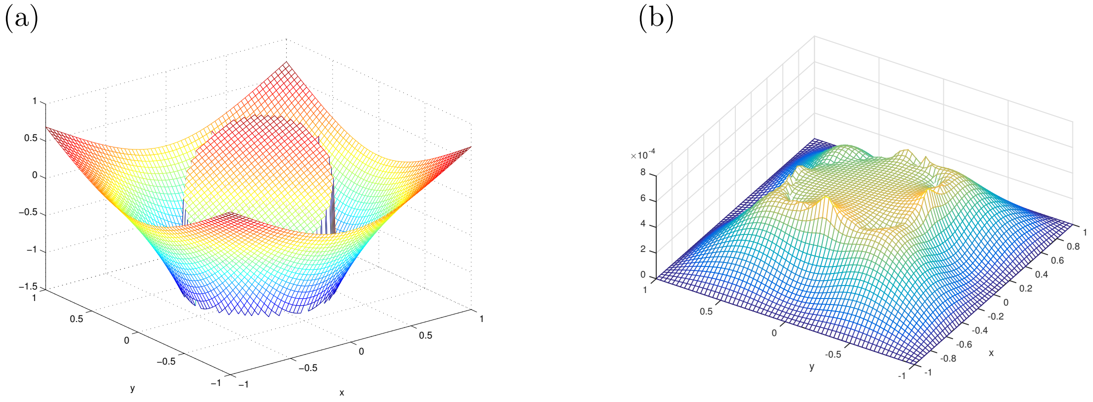

Figure 2.2: (a): The solution plot of Example 2.1. (b): The error plot of the computed solution. The error seems to be piecewise smooth as well which is important for accurate gradient computation.

can observe from Table 2.2 that the new augmented IIM is second order accurate both in the solution globally and the gradient at the interface from each side. The total CPU time also shows that the method is very fast with the optimal computational complexity $O(N^2 \log(N^2))$. We also observe that the number of GMRES iteration is a constant independent of the mesh size.

Table 2.2: A grid refinement analysis for Example 2.1 with a modest variable jump in the coefficient.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	3.1245×10^{-3}		1.5905×10^{-2}		2.1712×10^{-2}		4	48	0.01
64	7.0327×10^{-4}	2.15	4.4752×10^{-3}	1.82	5.5256×10^{-3}	1.97	4	92	0.03
128	1.1565×10^{-4}	2.60	1.1182×10^{-3}	2.00	1.3993×10^{-3}	1.98	4	184	0.11
256	2.7720×10^{-5}	2.06	2.9096×10^{-4}	1.94	3.7998×10^{-4}	1.88	4	364	0.46
512	6.2087×10^{-6}	2.15	7.3489×10^{-5}	1.98	9.8004×10^{-5}	1.95	4	728	2.45

Now we use the same exact solution and interface but with a large jump in the coefficient along the interface

$$\beta(\mathbf{x}) = \begin{cases} 10e^{10x} & \text{in } \Omega^-, \\ \sin(x+y)+2 & \text{in } \Omega^+. \end{cases} \tag{2.67}$$

The jump ratio varies from 1 : 10 to 1.45 : 1482 along the interface, a quite dramatic change. The results are shown in Table 2.3. We observe that the errors are larger than that in Table 2.2. This is due to the variations of the coefficient. Since the re-scaled PDE has the form $\Delta u +$

$\frac{1}{\beta}\nabla\beta(x)\cdot\nabla u+\cdots$, we would expect the error term contains $\frac{\beta_x}{\beta}u_x$ and $\frac{\beta_y}{\beta}u_y$ that are $O(1)$ for Table 2.2 and $O(10^2)$ for Table 2.3 due to the term $10e^{10x}$. This explains well in the difference in the errors. Nevertheless, all the nice features are the same as the previous example.

Table 2.3: A grid refinement analysis for Example 2.1 with a large variation in the jump ratio of the coefficient.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	6.0115×10^{-1}		3.9014		1.7570		11	48	0.02
64	1.4706×10^{-1}	2.03	8.6974×10^{-1}	2.16	4.2479×10^{-1}	2.04	10	92	0.07
128	4.3411×10^{-2}	1.76	2.5265×10^{-1}	1.78	1.2537×10^{-1}	1.76	10	184	0.27
256	1.1266×10^{-2}	1.94	6.5293×10^{-2}	1.95	3.2493×10^{-2}	1.94	10	364	1.24
512	2.9178×10^{-3}	1.94	1.6916×10^{-2}	1.94	8.4205×10^{-3}	1.94	10	728	7.85

An example with more general jump conditions

There are some applications in which we may have more general jump conditions. Here we consider an example with a more general jump condition, $c(\mathbf{X})u_n^+ - d(\mathbf{X})u_n^- = v(\mathbf{X})$ with $c(\mathbf{X}) = x^2+1$, $d(\mathbf{X}) = y^2+1$. Our method still can work with the modified augmented equation (2.51) (now it is $c(\mathbf{X})u_n^+ - d(\mathbf{X})u_n^- = v(\mathbf{X})$) and different preconditioning techniques. The convergency analysis may not apply directly anymore. In Table 2.4, we list the grid refinement analysis which has the same predicted convergence and efficiency.

Table 2.4: A grid refinement analysis with a different jump condition $c(\mathbf{X})u_n^+ - d(\mathbf{X})u_n^- = v(\mathbf{X})$.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	1.0220×10^{-2}		2.4784×10^{-2}		2.5529×10^{-2}		4	48	0.047
64	2.9257×10^{-3}	1.80	6.5909×10^{-3}	1.91	6.2884×10^{-3}	2.02	4	92	0.172
128	7.9724×10^{-4}	1.87	1.7941×10^{-3}	1.87	1.5052×10^{-3}	2.06	4	184	0.578
256	2.0781×10^{-4}	1.93	4.6908×10^{-4}	1.93	3.7353×10^{-4}	2.01	4	364	2.531
512	5.3324×10^{-5}	1.96	1.2033×10^{-4}	1.96	9.5584×10^{-5}	1.96	4	728	10.141
1024	1.3253×10^{-5}	2.00	3.7729×10^{-5}	1.67	3.0206×10^{-5}	1.66	4	1452	44.281

Example 2.2. *A general interface example with piecewise constant coefficient.*

The example is from [44]. The interface is the zero level set function

$$\varphi(\mathbf{x}) = r - (0.5 + 0.2\sin(5\theta)), \tag{2.68}$$

and the true solution is

$$u(\mathbf{x}) = \begin{cases} \dfrac{r^2}{\beta^-} & \mathbf{x} \in \Omega^-, \\ \dfrac{r^4 + C_0 \log(2r)}{\beta^+} & \mathbf{x} \in \Omega^+. \end{cases} \tag{2.69}$$

The interface is complicated and has relatively large curvature at some places, see Figure 1.1. We repeat this example with the new preconditioning technique with $\beta^+ = 1000$ and $\beta^- = 1$ on the domain $\Omega = [-1,1] \times [-1,1]$. The results are shown in Table 2.5 that are almost the same as the original fast IIM in [44]. Once again we observe that both the solution and the gradient are second order accurate and the number of GMRES iterations is independent of the mesh size. For this example, the interface has large curvature at some places. We need a reasonable fine mesh to resolve the interface.

Table 2.5: A grid refinement analysis for Example 2.2 with a piecewise constant coefficient $\beta^+ = 1000$ and $\beta^- = 1$ and a complicated interface.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	1.2880×10^{-1}		3.0969		3.0971×10^{-3}		14	78	0.04
64	1.9431×10^{-1}	-0.59	4.0397	-0.38	4.0396×10^{-3}	-0.38	14	154	0.13
128	1.8734×10^{-2}	3.37	4.9723×10^{-1}	3.02	4.9765×10^{-4}	3.02	11	308	0.28
256	2.3009×10^{-3}	3.02	1.2096×10^{-1}	2.03	1.2352×10^{-4}	2.01	10	612	0.99
512	3.6351×10^{-4}	2.66	2.2790×10^{-2}	2.40	2.4786×10^{-5}	2.31	9	1226	21.50

2.5.1 An example for more general self-adjoint elliptic interface problems

With some modifications, the method developed in this paper has been generalized to more general interface problems

$$\nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) - \sigma(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}). \tag{2.70}$$

The regularity requirement for the existence of the solution includes additional conditions $\sigma(\mathbf{x}) \in C(\Omega^{\pm})$ and $\sigma(\mathbf{x}) \geq 0$. While we still get second order accuracy both in the solution and the gradient, the coefficient matrix from the modified algorithm may not be an M-matrix anymore. Nevertheless, those affected entries are of $O(1)$ compared $O(1/h^2)$ when $\sigma(\mathbf{x}) = 0$, that is, $A_h^{\sigma\neq 0} = A_h^{\sigma=0}(I+B_h)$ with $\|B_h\| \leq Ch^2$. Thus we have asymptotic convergence as those presented in the paper as $h \to 0$.

Example 2.3. *A general example with $\sigma(\mathbf{x}) \neq 0$. We present a more general example with a non-zero $\sigma(\mathbf{x})$ term with different interfaces, an ellipse and a five-star. The true solution and coefficient are*

$$u(\mathbf{x}) = \begin{cases} -x^3 + 2y^3 & \text{in } \Omega^-, \\ \sin(x+y) & \text{in } \Omega^+, \end{cases} \qquad \beta(\mathbf{x}) = \begin{cases} 1 + e^{x+2y} & \text{in } \Omega^-, \\ \sin(2x-y)+3 & \text{in } \Omega^+, \end{cases} \tag{2.71}$$

$$\sigma(\mathbf{x}) = \begin{cases} \cos(xy) + 2 & \text{in } \Omega^-, \\ x^2 + y^2 + 1 & \text{in } \Omega^+, \end{cases}, \tag{2.72}$$

where again $\Omega = [-1,1] \times [-1,1]$. This is a very general example for a self-adjoint elliptic interface problem with non-linear solution. We test our method for two different interfaces.

In Table 2.6, we show a grid refinement analysis for an elliptic interface $\varphi(x,y) = (x/0.6)^2 + (y/0.4)^2 - 1$. We observe once again second order convergence for the global solution and the gradient at the interface.

Table 2.6: A grid refinement analysis for Example 2.3 for a general elliptic interface problems with the interface $(x/0.6)^2 + (y/0.4)^2 = 1$.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	8.4180×10^{-3}		9.0987×10^{-2}		8.0826×10^{-2}		4	48	0.10
64	9.0036×10^{-4}	3.22	2.3473×10^{-2}	1.95	1.9611×10^{-2}	2.04	4	96	0.20
128	1.5842×10^{-4}	2.50	4.1771×10^{-3}	2.49	3.6921×10^{-3}	2.40	4	188	1.60
256	3.7209×10^{-5}	2.09	1.0639×10^{-3}	1.97	9.4238×10^{-4}	1.97	4	372	3.02
512	9.3380×10^{-6}	1.99	3.2952×10^{-4}	1.69	2.4187×10^{-4}	1.96	4	740	15.02

In Table 2.7, we show a grid refinement analysis for a skinny ellipse $\varphi(x,y) = x^2+(y/0.25)^2-1$ in the domain $[-1.5,\ 1.5] \times [-1.5,\ 1.5]$. Once we have the mesh fine enough to resolve the interface (here $N \geq 64$), we observe once again second order convergence for the global solution and the gradient at the interface although the largest error often appears near the tips of the longer axis of the ellipse.

If we increase the aspect ratio of the ellipse further, we can approximate the situations in which the domain has cracks, see Figure 2.4 in which we tried to find the electric potential in a domain containing an approximated crack $\varphi(x,y) = (x/0.5)^2 + (y/0.0625)^2 - 1$ within the

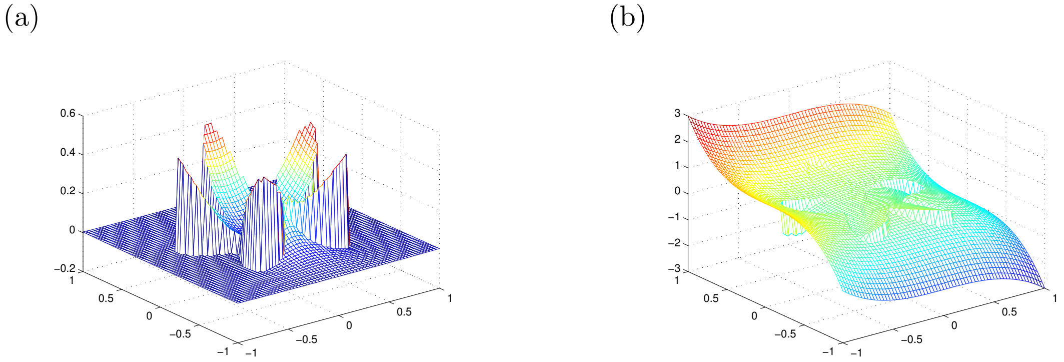

Figure 2.3: (a): The computed solution plot of Example 2.2. (b): The solution plot of Example 2.3.

Table 2.7: A grid refinement analysis for Example 2.3 for a general elliptic interface problems with the interface $x^2 + (y/0.25)^2 = 1$.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	8.2459×10^{-2}		1.3352		1.2700		5	44	0.078
64	4.7769×10^{-2}	0.78	8.9448×10^{-1}	0.57	6.7590×10^{-1}	0.90	5	88	0.156
128	6.5830×10^{-3}	2.85	1.6557×10^{-1}	2.43	1.5748×10^{-1}	2.10	5	176	1.250
256	9.2772×10^{-4}	2.82	7.1232×10^{-2}	1.21	4.7413×10^{-2}	1.73	5	352	2.250
512	1.7125×10^{-4}	2.43	1.4364×10^{-2}	2.31	1.1780×10^{-2}	2.00	5	704	19.703
1024	4.5859×10^{-5}	1.90	3.3351×10^{-3}	2.10	2.7234×10^{-3}	2.11	5	1408	77.812

domain $[-1,\ 1] \times [-1,\ 1]$. In this case, we have the PDE $\nabla \cdot (\beta \nabla u) = 0$, $[u]_\Gamma = 0$ and $[\beta u_n] = 0$, where β is the conductivity. The potential is given at the boundary with high potential on the right. Figure 2.4 (a) is the case with the ratio $\beta^+ : \beta^- = 1 : 1000$, while Figure 2.4 (b) the ratio is $\beta^+ : \beta^- = 1000 : 1$. Note that, we have tested the code against the analytic solution (2.71) for which we get the same convergence order. More sophisticated techniques and analysis can be found in [15, 67, 55, 52, 66].

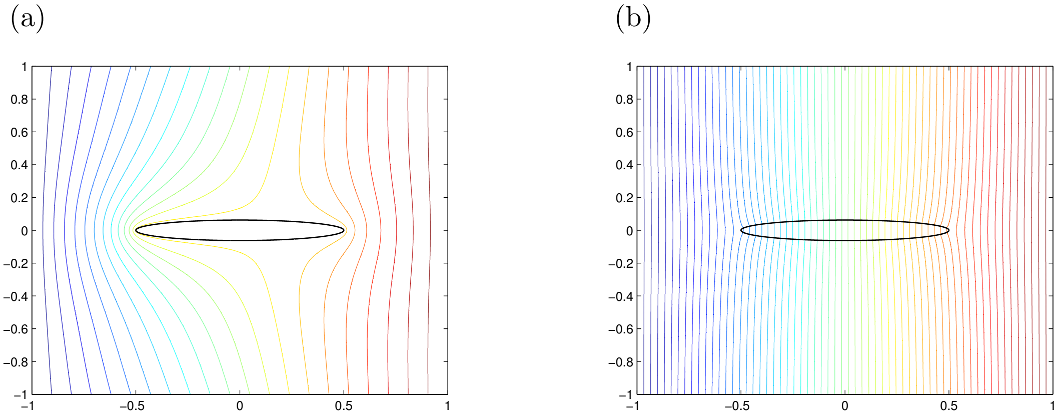

Figure 2.4: Electric potential in a domain containing a thin elliptic object. (a) The conductivity of the object is large (1 : 1000); (b) The conductivity of the object is small (1000 : 1).

In Table 2.8, we show a grid refinement analysis for the five-star interface $\varphi(x, y) = r - (0.5 + 0.2 \sin 5\theta)$ in polar coordinates (r, θ), $0 \le \theta < 2\pi$. While we still observe average second order convergence for the global solution and the gradient at the interface, the errors are fluctuated more even though the average convergence rate is the same, compared with the elliptic interface. We do observe that again, for complicated interfaces, we need to resolve the interface to obtain an accurate solution and its gradient.

Table 2.8: A grid refinement analysis for Example 2.3 for a general elliptic interface problems with a five-star interface, see Figure 1.1.

N	$E(u)$	r	$E(u_n^-)$	r	$E(u_n^+)$	r	Iter	N_b	CPU(s)
32	9.7746×10^{-1}		2.8683×10^{1}		4.7767		11	78	0.03
64	6.5486×10^{-2}	3.89	1.9417	3.88	3.2336×10^{-1}	3.88	10	154	0.10
128	1.5688×10^{-2}	2.06	6.4504×10^{-1}	1.58	1.0623×10^{-1}	1.60	9	308	0.29
256	1.8890×10^{-3}	3.05	1.6617×10^{-1}	1.95	3.6157×10^{-2}	1.55	9	612	1.07
512	3.8770×10^{-4}	2.28	2.9833×10^{-2}	2.47	6.7852×10^{-3}	2.41	9	1226	19.22

Chapter 3

A Direct Method for Accurate Solution and Gradient Computations for Elliptic Interface Problems

In this chapter, we propose a direct IIM for the elliptic interface problem

$$\nabla \cdot (\beta(\mathbf{x}) \nabla u(\mathbf{x})) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \setminus \Gamma, \quad \Omega = \Omega^+ \cup \Omega^-, \tag{3.1}$$

$$[u](\mathbf{X}) = w(\mathbf{X}), \qquad [\beta u_n](\mathbf{X}) = v(\mathbf{X}), \qquad \mathbf{X} \in \Gamma, \tag{3.2}$$

in one and two space dimensions. where for example, $[u] = [u]_\Gamma(\mathbf{X}) = u^+(\mathbf{X}) - u^-(\mathbf{X})$ is the difference of the limiting values of $u(\mathbf{X})$ from Ω^+ and Ω^- sides, respectively, $u_n = \mathbf{n} \cdot \nabla u = \frac{\partial u}{\partial n}$ is the normal derivative of solution $u(\mathbf{X})$, and $\mathbf{n}(\mathbf{X})$ is the unit normal direction at a point $\mathbf{X}$ on the interface pointing to Ω^+ side, see Fig. 1.1 for an illustration. We use $\mathbf{x}$ to represent a point in the domain while $\mathbf{X}$ a point on the interface Γ. Since a finite difference discretization will be used, we assume that $f(\mathbf{x}) \in C(\Omega^\pm)$, $\beta(\mathbf{x}) \in C^1(\Omega^\pm)$, excluding Γ; and $\Gamma \in C^2$, $w \in C^2(\Gamma)$, $v \in C^1(\Gamma)$. All the parameters and $\frac{\partial \beta}{\partial x}$ and $\frac{\partial \beta}{\partial y}$ are assume to be bounded. For the regularity requirement of the problem, we also assumed that $\beta(\mathbf{x}) \geq \beta_0 > 0$ and $f(\mathbf{x}) \in C^\nu(\Omega \backslash \Gamma)$, for a constant $\nu > 0$ so that $u(\mathbf{x}) \in C^{2+\nu}(\Omega^\pm)$, see [17, 2]. For the error analysis, piecewise higher regularity assumptions are needed for the solution.

3.1 The one-dimensional algorithm

The model problem in one dimension has the following form

$$(\beta u_x)_x - \sigma(x)u = f(x), \quad x \in (a,\alpha) \cup (\alpha,b), \tag{3.3}$$

$$u(a) = u_a, \quad u(b) = u_b, \quad [u]_a = w, \quad [\beta u_x]_\alpha = v, \tag{3.4}$$

where $a < \alpha < b$ is an interface (a point). We assume that the conditions for $\beta(x), \sigma(x)$, and $f(x)$ described in the introduction section hold in $\Omega^- = (a,\alpha)$ and $\Omega^+ = (\alpha,b)$.

Let $x_i = a + ih$ be a uniform mesh with $h = (b-a)/N$ and $i = 0,1,\cdots,N$. Assume that $x_j \le \alpha < x_{j+1}$. We call x_j and x_{j+1} *irregular* grid points while others are called *regular* grid points. The finite difference scheme at a regular grid point $x_i, i \neq j$ and $i \neq j+1$ can be written as

$$\frac{\beta_{i-1/2}U_{i-1} - 2\bar{\beta}_i U_i + \beta_{i+1/2}U_{i+1}}{\bar{\beta}_i h^2} - \frac{\sigma_i}{\bar{\beta}_i}U_i = \frac{f(x_i)}{\bar{\beta}_i}, \tag{3.5}$$

where

$$\beta_{i-1/2} = \beta(x_i - h/2), \quad \beta_{i+1/2} = \beta(x_i + h/2), \quad \bar{\beta}_i = (\beta_{i-1/2} + \beta_{i+1/2})/2. \tag{3.6}$$

At the irregular grid points x_j and x_{j+1}, we use the following nonconservative equivalent differential equation

$$u_{xx} + \frac{\beta_x}{\beta}u_x - \frac{\sigma u}{\beta} = \frac{f}{\beta}. \tag{3.7}$$

If $\beta_x(x_j)/\beta(x_j) \ge 0$, then the finite difference discretization at the irregular grid point x_j can be written as

$$\frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} + C_1 + \frac{\beta_x(x_j)}{\beta(x_j)}\Big(\frac{U_{j+1} - U_j}{h} + C_2\Big) - \frac{\sigma(x_j)}{\beta(x_j)}U_j = \frac{f(x_j)}{\beta(x_j)}, \tag{3.8}$$

where C_2 is the correction term for the upwind scheme

$$C_2 = -\frac{[u]}{h} - \frac{x_{j+1} - \alpha}{h}[u_x], \tag{3.9}$$

and C_1 is the correction term for the central difference scheme

$$C_1 = -\frac{[u]}{h^2} - \frac{x_{j+1} - \alpha}{h^2}[u_x] - \frac{(x_{j+1} - \alpha)^2}{2h^2}[u_{xx}]. \tag{3.10}$$

The case when $\beta_x(x_j)/\beta(x_j) < 0$ can be treated in a similar way. We omit the details here. We can derive a similar finite difference scheme at the irregular grid point x_{j+1}.

Remark 3.1. *The choices of the upwind scheme for u_x are based on the sign of its coefficient β_x/β. Our intention is to make the coefficient matrix A diagonally dominant when the correction* *terms C_1, C_2 are zero. For problems with nonzero correction terms, it is still possible to use optimization techniques to manipulate the formulation of C_1, C_2, such that this matrix structure is preserved. This approach is applied to design a maximum principle preserving scheme in the 2D algorithm and will be explained later.*

Remark 3.2. *At regular grid points, the local truncation error is $O(h^2)$. While at irregular gird points, it is easy to show that the local truncation error is $O(h)$ by applying the Taylor expansion. Thus we only need to find a second order approximation of $[u_x]$ and a first order approximation of $[u_{xx}]$ to keep the local truncation error at irregular grid points to be $O(h)$. The finite difference schemes we derived are consistent.*

3.1.1 Interpolation schemes for $[u_x]$ and $[u_{xx}]$

In this section, we discuss how to derive interpolation schemes for $[u_x]$ and $[u_{xx}]$. First from the jump conditions (3.4), we can get the following jump relations

$$u^+ = u^- + w, \quad u_x^+ = \frac{\beta^-}{\beta^+} u_x^- + \frac{v}{\beta^+}. \tag{3.11}$$

Note that the differential equation (3.7) also provides additional two relations, which are

$$u_{xx}^+ = \frac{f^+}{\beta^+} + \frac{\sigma^+ u^+}{\beta^+} - \frac{\beta_x^+ u_x^+}{\beta^+}, \tag{3.12}$$

$$u_{xx}^- = \frac{f^-}{\beta^-} + \frac{\sigma^- u^-}{\beta^-} - \frac{\beta_x^- u_x^-}{\beta^-}. \tag{3.13}$$

Let x_k be an irregular point, where $k = j,\ j+1$. The Taylor expansion of $u(x_k)$ about the interface α is

$$u(x_k) = u^\pm + (x_k - \alpha) u_x^\pm + \frac{1}{2}(x_k - \alpha)^2 u_{xx}^\pm + O(h^3), \tag{3.14}$$

where the "+" or "−" sign is chosen depending on whether x_k lies on the "+" or "−" side of α. Using jump relations (3.11)-(3.13) to rewrite the limiting values u^+, u_x^+, u_{xx}^+, and u_{xx}^- shown in (3.14) in terms of u^-, and u_x^-, we obtain

$$u(x_j) = \xi_1 u^- + \xi_2 u_x^- + \xi_3 + O(h^3), \tag{3.15}$$

$$u(x_{j+1}) = \eta_1 u^- + \eta_2 u_x^- + \eta_3 + O(h^3), \tag{3.16}$$

where

$$\begin{aligned}\xi_1 &= 1 + \frac{\sigma^-}{2\beta^-}(x_j - \alpha)^2,\\ \xi_2 &= x_j - \alpha - \frac{\beta_x^-}{2\beta^-}(x_j - \alpha)^2,\\ \xi_3 &= \frac{f^-}{2\beta^-}(x_j - \alpha)^2,\end{aligned} \tag{3.17}$$

and

$$\begin{aligned}\eta_1 &= 1 + \frac{\sigma^+}{2\beta^+}(x_{j+1} - \alpha)^2,\\ \eta_2 &= \frac{\beta^-}{\beta^+}(x_{j+1} - \alpha) - \frac{\beta^-\beta_x^+}{2{\beta^+}^2}(x_{j+1} - \alpha)^2,\\ \eta_3 &= w + \frac{v(x_{j+1} - \alpha)}{\beta^+} + \frac{\beta^+(f^+ + \sigma^+ w) - \beta_x^+ v}{2{\beta^+}^2}(x_{j+1} - \alpha)^2.\end{aligned} \tag{3.18}$$

Solving the two equations above for u^- and u_x^- yields the following interpolation schemes

$$u^- = \delta_1 u_j + \delta_2 u_{j+1} + \delta_3 + O(h^3), \tag{3.19}$$

$$u_x^- = \gamma_1 u_j + \gamma_2 u_{j+1} + \gamma_3 + O(h^2), \tag{3.20}$$

where

$$\begin{aligned}\delta_1 &= \frac{\eta_2}{\xi_1\eta_2 - \eta_1\xi_2}, \quad \delta_2 = \frac{-\xi_2}{\xi_1\eta_2 - \eta_1\xi_2}, \quad \delta_3 = -\delta_1\xi_3 - \delta_2\eta_3,\\ \gamma_1 &= \frac{-\eta_1}{\xi_1\eta_2 - \eta_1\xi_2}, \quad \gamma_2 = \frac{\xi_1}{\xi_1\eta_2 - \eta_1\xi_2}, \quad \gamma_3 = -\gamma_1\xi_3 - \gamma_2\eta_3.\end{aligned} \tag{3.21}$$

Note that if we assume $\beta_x^- \geq 0$ and $\beta_x^+ \leq 0$ which includes the case where β is piecewise constant, according to (3.17)-(3.18) we have

$$\xi_1 > 1, \quad \xi_2 < 0, \quad \eta_1 > 1, \quad \eta_2 > 0, \tag{3.22}$$

which guarantees the denominator $\xi_1\eta_2 - \eta_1\xi_2$ in (3.21) is positive and bounded away from zero, i.e.

$$\begin{aligned}\xi_1\eta_2 - \eta_1\xi_2 &> \frac{\beta^-}{\beta^+}(x_{j+1} - \alpha) + (\alpha - x_j)\\ &\geq \min\left\{\frac{\beta^-}{\beta^+}, 1\right\} h.\end{aligned} \tag{3.23}$$

The inequality (3.22) is still satisfied when $\beta_x^- < 0$ or $\beta_x^+ > 0$ so long as the mesh step size h is sufficiently small. A sufficient but not necessary condition on h for a nonzero denominator can be easily derived as

$$|h\beta_x^{\pm}| < 2\beta^{\pm}. \tag{3.24}$$

Using jump relations (3.11)-(3.13), one can then get

$$u^+ = \delta_1 u_j + \delta_2 u_{j+1} + \delta_3 + w + O(h^3), \tag{3.25}$$

$$u_x^+ = \rho\gamma_1 u_j + \rho\gamma_2 u_{j+1} + \rho\gamma_3 + \frac{v}{\beta^+} + O(h^2), \tag{3.26}$$

$$u_{xx}^- = \frac{f^-}{\beta^-} + \frac{\sigma^-}{\beta^-}(\delta_1 u_j + \delta_2 u_{j+1} + \delta_3) - \frac{\beta_x^-}{\beta^-}(\gamma_1 u_j + \gamma_2 u_{j+1} + \gamma_3) + O(h), \tag{3.27}$$

$$u_{xx}^+ = \frac{f^+}{\beta^+} + \frac{\sigma^+}{\beta^+}(\delta_1 u_j + \delta_2 u_{j+1} + \delta_3 + w) - \frac{\beta_x^+}{\beta^+}(\rho\gamma_1 u_j + \rho\gamma_2 u_{j+1} + \rho\gamma_3 + \frac{v}{\beta^+}) + O(h), \tag{3.28}$$

where $\rho = \beta^-/\beta^+$ is the jump ratio. Hence, the schemes for derivative jumps can be set as $[u_x] = u_x^+ - u_x^-$ and $[u_{xx}] = u_{xx}^+ - u_{xx}^-$. Notice that the scheme for $[u_x]$ is second order accurate while that for $[u_{xx}]$ is only first order.

Remark 3.3. *The interpolation schemes we derived above only involve two grid points, which will play an important role in the stability proof. Also in the case where $\sigma = 0$, we only need the scheme of u_x^- to get approximations for $[u_x]$ and $[u_{xx}]$. In this situation, we have $\eta_1 = \xi_1$ and hence $\gamma_1 + \gamma_2 = 0$. We say the interpolation scheme is consistent if the sum of the coefficients equals zero.*

Substituting the interpolation schemes we derived for $[u_x]$ and $[u_{xx}]$ back into the finite difference scheme (3.8) yields

$$A_h \mathbf{U} = F. \tag{3.29}$$

The coefficient matrix A_h is tri-diagonal since the interpolation schemes only involve two irregular grid points. Hence, the system can be solved in $O(N)$ operations.

3.2 Convergence analysis of the 1D algorithm

In this section, we show second order convergence of the finite difference solution and its first order derivative at the interface in the case where β is a piecewise constant and $\sigma = 0$. Without loss of generality, we assume the domain is $(0, 1)$ and a homogeneous Dirichlet boundary condition at the two ends for simplicity.

Since β is piecewise constant, we can incorporate it into the source term $f(x)$ on the right-hand side of the equation. We begin by studying the Green's function solution to the BVP

$$u''(x) = f(x), \quad x \in (0, \alpha) \cup (\alpha, 1), \tag{3.30}$$

with homogeneous boundary conditions and jump conditions

$$u(0) = 0, \quad u(1) = 0, \quad [u]_\alpha = 0, \quad [\beta u_x]_\alpha = 0. \tag{3.31}$$

For any fixed point $\bar{x} \in [0, 1]$ and $\bar{x} \neq \alpha$, the Green's function $G(x; \bar{x})$ is the function of x that solves the particular BVP of the above form with $f(x) = \delta(x - \bar{x})$, where $\delta(x)$ is the Dirac delta function.

Let us assume $\bar{x} < \alpha$. A little algebra shows that the piecewise linear function $G(x; \bar{x})$ has the form

$$G(x; \bar{x}) = \begin{cases} k_1 x & \text{for } 0 \leq x \leq \bar{x}, \\ k_2 x - \bar{x} & \text{for } \bar{x} \leq x \leq \alpha, \\ k_3 x - k_3 & \text{for } \alpha \leq x \leq 1, \end{cases} \tag{3.32}$$

where k_1, k_2, k_3 subject to the following constrains.

$$\begin{cases} k_2 - k_1 = 1 & \text{from} \quad u''(x) = \delta(x - \bar{x}), \\ (\alpha - 1)k_3 = \alpha k_2 - \bar{x} & \text{from} \quad [u] = 0 \\ \beta^+ k_3 = \beta^- k_2 & \text{from} \quad [\beta u_x] = 0. \end{cases} \tag{3.33}$$

Solving the above system of equations yields

$$\begin{aligned} k_1 &= \frac{\beta^+(\bar{x} - \alpha) - \beta^-(1 - \alpha)}{\omega}, \\ k_2 &= \frac{\beta^+ \bar{x}}{\omega}, \\ k_3 &= \frac{\beta^- \bar{x}}{\omega}, \end{aligned} \tag{3.34}$$

where $\omega = \beta^+\alpha + \beta^-(1 - \alpha)$. The case where $\bar{x} > \alpha$ can be analyzed in the same way and the proof is skipped.

Theorem 3.1. *Let B be a matrix of the size of A_h from* (3.29)*. Let B_{ij} denote the entry at ith row and jth column and let B_j denote the jth column of B. Define*

$$B_{ij} = h\, G(x_i; x_j). \tag{3.35}$$

Then, B is the inverse of A_h, i.e. $B = A_h^{-1}$.

Proof. We only need to show $A_hB_j = e_j$, where e_j is the jth column of the identity matrix. By the way we construct the Green's function, this is true if the following conditions are satisfied.

1. The interpolation scheme for the correction term C_1 from (3.10) only involves the solution at two irregular grid points and brings no error for piecewise linear solution.

2. The finite difference schemes (3.5) and (3.8) have zero truncation error for piecewise linear solution if the first condition holds.

3. The interpolation scheme for correction term C_1 from (3.10) does not involves any constant terms. i.e. the scheme has the form $C_1 = \zeta_1 u_j + \zeta_2 u_{j+1}$ for some constants ζ_1, ζ_2.

Note that A_hU+C is the discrete form of the operation $\frac{d^2u}{dx^2}$ satisfying the Dirichlet homogeneous boundary conditions and jump conditions, where C is a vector of constants contributed by jump relations. For the problem we considered, the terms $[u]$ and $[u_{xx}]$ are all zero and hence the formula for the correction term C_1 from (3.10) becomes

$$C_1 = -\frac{x_{j+1} - \alpha}{h^2}[u_x]. \tag{3.36}$$

For piecewise linear solution, the remainder term $O(h^3)$ in the Taylor expansion (3.14) disappears. Since the scheme we derived for $[u_x]$ is second order accurate and only involves two irregular grid points, the scheme for the correction term C_1 brings no error and the first condition holds. (If the interpolation scheme involves solution at three grid points, say u_{j-1}, u_j, and u_{j+1}, and $\bar{x} = x_j$, then the error term will not disappear. This is the reason why we derive a two-point interpolation scheme.)

The second condition will be automatically satisfied if the first condition holds. This is because the central difference scheme is second order accurate and hence brings in no error for a piecewise linear solution. Hence, the second condition ensures that $A_hB_j + C = e_j$.

To prove the third condition, we only need to show the scheme (3.20) for u_x^- involves no constant term. i.e. $\gamma_3 = 0$. Since the terms w, v, $[u_{xx}]$, f^+, f^- are all zero, the terms ξ_3, and η_3 in (3.21) are zero. Therefore, $\gamma_3 = 0$ and the third condition holds. Because of the homogeneous boundary conditions, we know $A_hB_j = e_j$. This completes the proof. □

Theorem 3.2. *For the case where β is piecewise constant and $\sigma = 0$, the finite difference scheme that we proposed is convergent and has second order accuracy.*

Proof. Denote $\mathbf{E}^u = \mathbf{U} - \hat{\mathbf{U}}$, where $\hat{\mathbf{U}}$ is the vector of true values of u evaluated at grid points. Let $\mathbf{T}^u$ be the local truncation error of the system (3.29), that is,

$$A_h\hat{\mathbf{U}} = F + \mathbf{T}^u. \tag{3.37}$$

Subtracting equation (3.37) from (3.29) yields

$$A_h \mathbf{E}^u = \mathbf{T}^u. \tag{3.38}$$

Notice that the magnitude of $\mathbf{T}^u$ is $O(h^2)$ at regular grid points and is $O(h)$ at irregular ones. By using the Green's function, the global error of u can be represented as

$$\mathbf{E}_i^u = h \sum_{j=1}^{N-1} \mathbf{T}_j^u G(x_i; x_j). \tag{3.39}$$

Since $|G(x_i; x_j)|$ is uniformly bounded by a constant, let us say C, we have the inequality

$$\begin{aligned} |\mathbf{E}_i^u| &< C \left| h \sum_{j=1}^{N-1} \mathbf{T}_j^u \right| \\ &\leq Ch \Bigg(|\mathbf{T}_j^u| + |\mathbf{T}_{j+1}^u| + \sum_{k=1}^{j-1} |\mathbf{T}_k^u| + \sum_{k=j+2}^{N-1} |\mathbf{T}_k^u| \Bigg) \\ &\approx Ch \Big(2O(h) + (N-2)O(h^2) \Big) \\ &\approx O(h^2). \end{aligned}$$

This shows that $\|\mathbf{E}^u\|_\infty \leq Ch^2$, hence completing the proof. □

Theorem 3.3. *The scheme* (3.20) *generates a second order approximation for the first derivative of the solution from each side of the interface.*

Proof. Let U_x^- be the numerical solution of u_x^-. Denote $E^q = U_x^- - u_x^-$ and T^q be the residual of the equation (3.20), that is,

$$u_x^- = \gamma_1 u_j + \gamma_2 u_{j+1} + \gamma_3 + T^q, \tag{3.40}$$

where in this case

$$\gamma_1 = \frac{-1}{(\alpha - x_j) + \rho(x_{j+1} - \alpha)}, \tag{3.41}$$

$$\gamma_2 = \frac{1}{(\alpha - x_j) + \rho(x_{j+1} - \alpha)}. \tag{3.42}$$

Subtracting equation (3.40) from (3.20) yields

$$E^q = \gamma_1 \mathbf{E}_j^u + \gamma_2 \mathbf{E}_{j+1}^u + T^q. \tag{3.43}$$

Notice that T^q is $O(h^2)$, and $\mathbf{E}_j^u, \mathbf{E}_{j+1}^u$ are $O(h^2)$. From (3.41)-(3.42), we know γ_1, γ_2 are $O(1/h)$ and $\gamma_1 + \gamma_2 = 0$. Hence,

$$\begin{aligned}
|E^q| &= |\gamma_1 \mathbf{E}_j^u + \gamma_2 \mathbf{E}_{j+1}^u + T^q| \\
&= |\gamma_2 (\mathbf{E}_{j+1}^u - \mathbf{E}_j^u) + T^q| \\
&\le |\gamma_2| \cdot |\mathbf{E}_{j+1}^u - \mathbf{E}_j^u| + |T^q| \\
&\approx O(h^2),
\end{aligned}$$

if $|\mathbf{E}_{j+1}^u - \mathbf{E}_j^u| \approx O(h^3)$. Notice that $|G(x_{j+1}; x_k) - G(x_j; x_k)| \le Ch$ for some constant C since k_1, k_2, k_2 from (3.34) are bounded for all different x_k. Hence, we have

$$\begin{aligned}
|\mathbf{E}_{j+1}^u - \mathbf{E}_j^u| &= h \sum_{k=1}^{N-1} |\mathbf{T}_k^u| \cdot |G(x_{j+1}; x_k) - G(x_j; x_k)| \\
&\le Ch^2 \sum_{k=1}^{N-1} |\mathbf{T}_k^u| \\
&\approx O(h^3).
\end{aligned}$$

Therefore, we have $|E^q| \approx O(h^2)$. This completes the proof. □

3.3 The algorithm for two dimensional problems

In this section, we present the algorithm for two dimensional problems. Assuming the space domain is $\Omega = (a, b) \times (c, d)$, for simplicity of discussion, we use a uniform mesh

$$x_i = a + ih, \quad i = 0, 1, \cdots M; \quad y_j = c + jh, \quad j = 0, 1, \cdots, N, \tag{3.44}$$

where h is the space step size. The interface Γ is represented by the zero level set of a Lipschitz continuous function $\phi(x, y)$, that is

$$\Gamma = \big\{(x, y), \quad \phi(x, y) = 0, \quad (x, y) \in \Omega\big\}. \tag{3.45}$$

The level set function $\phi(x, y)$ satisfies

$$\begin{aligned}
\phi(x, y) < 0 \quad &\text{for } (x, y) \in \Omega^-, &(3.46)\\
\phi(x, y) = 0 \quad &\text{for } (x, y) \in \Gamma, &(3.47)\\
\phi(x, y) > 0 \quad &\text{for } (x, y) \in \Omega^+. &(3.48)
\end{aligned}$$

In the neighborhood of the interface, we assume that $\phi(x, y) \in C^2$. In implementation, the level set function is defined at the grid points as $\{\phi_{ij}\}$. At a grid point (x_i, y_j), we define

$$\phi_{ij}^{max} = \max \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}, \tag{3.49}$$

$$\phi_{ij}^{min} = \min \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}. \tag{3.50}$$

A grid point (x_i, y_j) is called *regular* if $\phi_{ij}^{max}\phi_{ij}^{min} > 0$, otherwise it is called irregular. The intersections on the interface with the grid lines are called control points, see Figure 3.1 for an illustration.

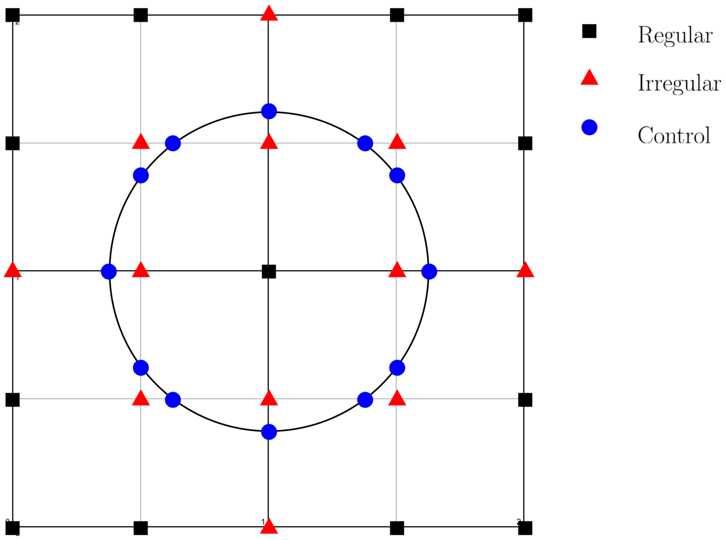

Figure 3.1: A diagram of regular, irregular, and control points in a square domain with uniform mesh and circle interface.

The interface information, such as the normal unit vector and signed curvature, etc., can be easily computed from

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{3.51}$$

$$\kappa = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}. \tag{3.52}$$

We next discuss how to derive the finite difference scheme.

3.3.1 The finite difference scheme in 2D

At a regular grid point (x_i, y_j), the finite difference scheme is the classic conservative one with the scaling by discretizing equation (1.1)

$$\frac{\beta_{i-\frac{1}{2},j}U_{i-1,j} + \beta_{i+\frac{1}{2},j}U_{i+1,j} + \beta_{i,j-\frac{1}{2}}U_{i,j-1} + \beta_{i,j+\frac{1}{2}}U_{i,j+1} - 4\bar{\beta}_{i,j}U_{i,j}}{\bar{\beta}_{i,j}h^2} - \frac{\sigma_{i,j}}{\bar{\beta}_{i,j}}U_{i,j} = \frac{f_{i,j}}{\bar{\beta}_{i,j}}, \quad (3.53)$$

where $f_{i,j} = f(x_i, y_j)$, $\beta_{i-\frac{1}{2},j} = \beta(x_i - \frac{h}{2}, y_j)$ and so on, and

$$\bar{\beta}_{i,j} = (\beta_{i-\frac{1}{2},j} + \beta_{i+\frac{1}{2},j} + \beta_{i,j-\frac{1}{2}} + \beta_{i,j+\frac{1}{2}})/4. \quad (3.54)$$

To derive a scheme at irregular grid point, we reformulate equation (1.1) as

$$u_{xx} + u_{yy} + \frac{\beta_x}{\beta}u_x + \frac{\beta_y}{\beta}u_y - \frac{\sigma}{\beta}u = \frac{f}{\beta}. \quad (3.55)$$

We will discretize equation (3.55) dimension by dimension. The central difference scheme is applied for the second order derivative terms while an upwind scheme is used for the first order derivative terms.

Consider the discretization in x-direction. Let (x_i, y_j) be an irregular grid point. The finite difference approximations for u_{xx} and $\frac{\beta_x}{\beta}u_x$ have the following forms respectively.

$$u_{xx} \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j} + \hat{C}^x_{i,j}}{h^2}, \quad (3.56)$$

$$\frac{\beta_x}{\beta}u_x \approx \begin{cases} \dfrac{\beta_x}{\beta}\dfrac{U_{i,j} - U_{i-1,j} + \tilde{C}^x_{i,j}}{h} & \text{if } \dfrac{\beta_x}{\beta} \le 0, \\ \dfrac{\beta_x}{\beta}\dfrac{U_{i+1,j} - U_{i,j} + \tilde{C}^x_{i,j}}{h} & \text{otherwise .} \end{cases} \quad (3.57)$$

Notice that the way we use upwind scheme depends on the sign of β_x/β. As explained in Remark 3.1, our intention is to make the coefficient matrix an M-matrix when all the correction terms are zero. This makes it possible to design a maximum principle preserving scheme by manipulating the formulation of the correction terms. Now let's first suppose the interface does not cut through the interval (x_{i-1}, x_{i+1}) along the line $y = y_j$, that is, (x_i, y_j) is regular in the x-direction, then the correction terms are zeros, i.e. $\hat{C}^x_{i,j} = \tilde{C}^x_{i,j} = 0$.

Next we assume the interface cuts the grid line $y(x) = y_j$ in the interval (x_i, x_{i+1}), say at (x_i^*, y_j). Without loss of generality, we assume that $(x_i, y_j) \in \Omega^-$. By using Taylor expansion, one can derive the formula for the correction terms.

$$\hat{C}^x_{i,j} = -[u] - (x_{i+1} - x_i^*)[u_x] - \frac{(x_{i+1} - x_i^*)^2}{2}[u_{xx}], \tag{3.58}$$

$$\tilde{C}^x_{i,j} = \begin{cases} 0 & \text{if } \dfrac{\beta_x}{\beta} \leq 0, \\ -[u] - (x_{i+1} - x_i^*)[u_x] & \text{otherwise.} \end{cases} \tag{3.59}$$

In this particular case, the corrected values $U_{i+1,j} + \hat{C}^x_{i,j}$ and $U_{i+1,j} + \tilde{C}^x_{i,j}$ (when $\beta_x > 0$) can be interpreted as the smooth extensions of the solution at (x_{i+1}, y_j) from Ω^- side. Combing these two correction terms, the discretization in x-direction can be rewritten as

$$u_{xx} + \frac{\beta_x}{\beta} u_x \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} + \frac{\beta_x}{\beta}\frac{U_{i+1,j} - U_{i,j}}{h} + C^x_{i,j}, \tag{3.60}$$

where

$$C^x_{i,j} = \frac{\hat{C}^x_{i,j}}{h^2} + \frac{\beta_x}{\beta}\frac{\tilde{C}^x_{i,j}}{h}. \tag{3.61}$$

In a similar way, we define $C^y_{i,j}$ as the correction term in the y-direction.

Remark 3.4. *When we use a level set representation of the interface, it can only cut a grid line once between two grid points. For example, in the above description, (x_i^*, y_j) should be the only intersection point in the interval (x_i, x_{i+1}). But it is possible for the same irregular grid point (x_i, y_j) to have another intersection point in the left interval (x_{i-1}, x_i). In this case, the formulas* (3.58)-(3.59) *are still valid but only consider the corrections contributed by $U_{i+1,j}$. One can simply modify them by adding the corrections contributed by $U_{i-1,j}$. E.g. using the same assumption that $(x_i, y_j) \in \Omega^-$, the formula* (3.58) *will become*

$$\begin{aligned} \hat{C}^x_{i,j} = &- [u]_R - (x_{i+1} - x_i^*)[u_x]_R - \frac{(x_{i+1} - x_i^*)^2}{2}[u_{xx}]_R \\ &- [u]_L - (x_{i-1} - x_i^*)[u_x]_L - \frac{(x_{i-1} - x_i^*)^2}{2}[u_{xx}]_L, \end{aligned} \tag{3.62}$$

where $[\cdot]_R$ and $[\cdot]_L$ represent the jump values at the right and left intersection point respectively. The other correction terms can be derived in a similar way.

The formula for the correction terms in other situations can be derived in a similar way. Note that it is enough to have the local truncation errors of the finite difference scheme at irregular grid points to be first order without affecting global second order accuracy. Hence, we only need to have first order approximations of the correction terms $C^x_{i,j}, C^y_{i,j}$. Before deriving the interpolation schemes for correction terms, we need to study the jump relations.

3.3.2 Jump relations and the coordinate transformation

To be self-contained, the following contents are presented and cited from [35, 38]. Assume that the interface in parametric form is

$$\Gamma = \{(X(s), Y(s)), \quad X(s) \in C^2, Y(s) \in C^2\}, \tag{3.63}$$

where s is a parameter, for example, the arc-length. At a point of the interface (X, Y), the local coordinate system in the normal and tangential directions is defined as

$$\begin{cases} \xi = (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta, \end{cases} \tag{3.64}$$

where θ is the angle between the x-axis and the normal direction, pointing to the Ω^+ subdomain, see Figure 5.3 for an illustration. In the new coordinate system, the interface can be parameterized by

$$\xi = \chi(\eta) \quad \text{with} \quad \chi(0) = 0, \quad \chi'(0) = 0. \tag{3.65}$$

The signed curvature of the interface at (X, Y) is $\chi''(0)$, the formula of which is provided by (5.20).

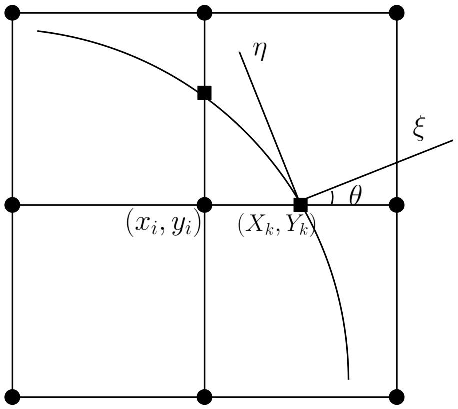

Figure 3.2: A diagram of an irregular grid point (x_i, y_i), its control point (X_k, Y_k) on the x-axis, and the local coordinates at (X_k, Y_k) in the normal and tangential directions, where θ is the angle between the x-axis and the normal direction.

If we know the jump conditions $[u] = w$ and $[\beta u_n] = v$, then we can have the following jump relations at a point (X, Y) on the interface which are necessary to derive an accurate finite difference method.

Theorem 3.4. *For elliptic interface problem* (1.1)-(1.2)*, given the curvature of the interface, κ, the two jump conditions $[u] = w$, $[\beta u_n] = v$ lead to the following jump relations in the local coordinate.*

$$
\begin{aligned}
[u] &= w, && (3.66)\\
[u_\eta] &= w', && (3.67)\\
[\beta u_\xi] &= v, && (3.68)\\
[u_{\eta\eta}] &= D_1 = -\kappa[u_\xi] + w'', && (3.69)\\
[\beta u_{\eta\xi}] &= D_2 = \kappa[\beta u_\eta] - [\beta_\eta u_\xi] + v',. && (3.70)
\end{aligned}
$$

The elliptic equation (1.1) *itself gives additional two jump relations in Cartesian coordinates.*

$$
\begin{aligned}
u^+_{xx} + u^+_{yy} &= D_3 = -\frac{\beta^+_x}{\beta^+}u^+_x - \frac{\beta^+_y}{\beta^+}u^+_y + \frac{\sigma^+}{\beta^+}u^+ + \frac{f^+}{\beta^+}, && (3.71)\\
u^-_{xx} + u^-_{yy} &= D_4 = -\frac{\beta^-_x}{\beta^-}u^-_x - \frac{\beta^-_y}{\beta^-}u^-_y + \frac{\sigma^-}{\beta^-}u^- + \frac{f^-}{\beta^-}. && (3.72)
\end{aligned}
$$

Here we skip the derivation which is similar to those derived in equation (3.5) in Section 3.1 in [38]. Through the coordinate transformation, one can get seven jump relations in Cartesian coordinates and express the limiting values in terms of $u^-, u^-_x, u^-_y, u^-_{xx}, u^-_{xy}$. Hence, we have the following,

$$
\begin{aligned}
u^+ &= u^- + w && (3.73)\\
u^+_x &= (s^2 + \rho c^2)u^-_x + (\rho - 1)scu^-_y - sw' + \frac{c}{\beta^+}v, && (3.74)\\
u^+_y &= sc(\rho - 1)u^-_x + (\rho s^2 + c^2)u^-_y + cw' + \frac{s}{\beta^+}v, && (3.75)\\
u^+_{xx} &= [s^4 + (4\rho - 2)s^2c^2 + c^4]u^-_{xx} + 2(\rho - 1)(s^3c - sc^3)u^-_{xy} \\
&\quad + (s^2 - c^2)D_1 - (2sc/\beta^+)D_2 + c^2D_3 - [(2\rho - 1)s^2c^2 + c^4]D_4, && (3.76)\\
u^+_{xy} &= 2(\rho - 1)(s^3c - sc^3)u^-_{xx} + [\rho s^4 - (2\rho - 4)s^2c^2 + \rho c^4]u^-_{xy} \\
&\quad - 2scD_1 + [(c^2 - s^2)/\beta^+]D_2 + scD_3 - [\rho s^3c - (\rho - 2)sc^3]D_4, && (3.77)\\
u^+_{yy} &= -[s^4 + (4\rho - 2)s^2c^2 + c^4]u^-_{xx} - 2(\rho - 1)(s^3c - sc^3)u^-_{xy} \\
&\quad - (s^2 - c^2)D_1 + (2sc/\beta^+)D_2 + s^2D_3 + [(2\rho - 1)s^2c^2 + c^4]D_4, && (3.78)\\
u^-_{yy} &= -u^-_{xx} + D_4. && (3.79)
\end{aligned}
$$

where $\rho = \beta^-/\beta^+$ is the jump ratio, $\beta^\pm$ are the limiting values from each side of the interface, and s, c stand for $\sin\theta$ and $\cos\theta$ respectively. Notice that D_1, D_2, D_3, and D_4 can all be expressed in terms of u^-, u_x^-, u_y^- by using formula (3.73)-(3.75).

3.3.3 The approximation of correction terms

In this section, we describe how to interpolate the correction terms at the irregular grid points. Here we derive the scheme for $C_{i,j}^x$ used in (3.60) as an example. We will use the jump relations (3.73)-(3.79) to rewrite the correction term $C_{i,j}^x$ and the Taylor expansions of the solution at grid points in terms of the limiting values $u^-, u_x^-, u_y^-, u_{xx}^-, u_{xy}^-$. Using the method of undetermined coefficients, a nine-point interpolation scheme can be derived.

Let (x_i, y_j) be an irregular point. Without loss of generality, we assume the interface cuts through the interval (x_{i-1}, x_{i+1}) along the line $y = y_j$ and intersects at the point (x_i^*, y_j). We will use the compact nine points (x_{i+i_k}, y_{j+j_k}) to derive the scheme, where i_k, j_k take values in the set $\{0, \pm 1\}$. The Taylor expansion of $u(x_{i+i_k}, y_{j+j_k})$ about (x_i^*, y_j) is

$$\begin{aligned} u(x_{i+i_k}, y_{j+j_k}) = u^\pm &+ (x_{i+i_k} - x_i^*)u_x^\pm + (y_{j+j_k} - y_j)u_y^\pm + \frac{1}{2}(x_{i+i_k} - x_i^*)^2 u_{xx}^\pm \\ &+ \frac{1}{2}(y_{j+j_k} - y_j)^2 u_{yy}^\pm + (x_{i+i_k} - x_i^*)(y_{j+j_k} - y_j)u_{xy}^\pm + O(h^3), \end{aligned} \tag{3.80}$$

where the "+" or "−" sign is chosen depending on whether (x_{i+i_k}, y_{j+j_k}) lies on the "+" or "−" side of Γ. Note that the right-hand side of the above equation can be expressed in terms of $u^-, u_x^-, u_y^-, u_{xx}^-, u_{xy}^-$ by using the jump relations (3.73)-(3.79). Hence, we get the scheme

$$U(x_{i+i_k}, y_{j+j_k}) = c_{i_k,j_k}^1 U^- + c_{i_k,j_k}^2 U_x^- + c_{i_k,j_k}^3 U_y^- + c_{i_k,j_k}^4 U_{xx}^- + c_{i_k,j_k}^5 U_{xy}^- + c_{i_k,j_k}^6, \tag{3.81}$$

for $i_k, j_k = 0, \pm 1$. Similarly, the equation (3.61) can be reformulated as

$$C_{i,j}^x = a_1 U^- + a_2 U_x^- + a_3 U_y^- + a_4 U_{xx}^- + a_5 U_{xy}^- + a_6, \tag{3.82}$$

where the coefficients a_1-a_6 are some constants determined by the model problem.

Assume the interpolation scheme for $C_{i,j}^x$ has the form

$$C_{i,j}^x = \sum_{i_k,j_k} \gamma_{i_k,j_k} U_{i+i_k,j+j_k} + \gamma_c, \tag{3.83}$$

where i_k, j_k take values of $0, \pm 1$. The coefficients $\{\gamma' s\}$ are determined by solving the linear system

$$\begin{array}{cc} & \begin{array}{ccccc} u_{i-1,j-1} & u_{i,j-1} & \cdots & u_{i,j+1} & u_{i+1,j+1} \end{array} \\ \begin{array}{c} u^- \\ u^-_x \\ u^-_y \\ u^-_{xx} \\ u^-_{xy} \end{array} & \begin{bmatrix} c^1_{-1,-1} & c^1_{0,-1} & \cdots & c^1_{0,1} & c^1_{1,1} \\ c^2_{-1,-1} & c^2_{0,-1} & \cdots & c^2_{0,1} & c^2_{1,1} \\ c^3_{-1,-1} & c^3_{0,-1} & \cdots & c^3_{0,1} & c^3_{1,1} \\ c^4_{-1,-1} & c^4_{0,-1} & \cdots & c^4_{0,1} & c^4_{1,1} \\ c^5_{-1,-1} & c^5_{0,-1} & \cdots & c^5_{0,1} & c^5_{1,1} \end{bmatrix} \end{array} \begin{bmatrix} \gamma_{-1,-1} \\ \gamma_{0,-1} \\ \cdots \\ \gamma_{0,1} \\ \gamma_{1,1} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \tag{3.84}$$

and setting

$$\gamma_c = a_6 - \sum_{i_k,j_k} c^6_{i_k,j_k} \cdot \gamma_k \tag{3.85}$$

Notice that the linear system (5.76) contains 5 equations with 9 unknown variables and hence is underdetermined. One can then use SVD to get the minimum norm solution. In this case, the scheme (3.83) derived for $C^x_{i,j}$ is first order accurate. At this point, the finite difference scheme used to discretize the space at irregular grid point is fully determined. In practice, the correction terms are part of coding of the numerical algorithm. The correction terms are only needed at irregular grid points near the interface. Thus the additional cost in determining the correction terms is $O(N)$ instead of that for the linear solver of the finite difference equations which is at least $O(N^2)$.

Denote $\mathbf{U}$ as the numerical solution of (1.1) at grid points from our approach. One can then write down the discretization of equation (1.1) in matrix-vector form as

$$A_h\mathbf{U} = F, \tag{3.86}$$

where A_h is the discrete elliptic operator, and vector F is assembled from source term f, boundary conditions, and correction terms. Notice that the coefficient matrix becomes the discrete Laplacian when β is constant and $\sigma = 0$. In this case, the above system of equations (3.86) can be easily solved by a fast Poisson solver and the solution is second order accurate. For problems with piecewise constant β, the second order convergence of the solution can be numerically proved by enforcing discrete maximum principle and is presented in the next section 3.3.4. For more general problems, the second order convergence for both solution and gradient will be demonstrated by numerical examples in Section 3.4.

The solution will then be used to recover the gradient at the interface. For a control point on the interface, let us assume (x_i, y_j) is its closest irregular grid point. The gradient in the normal direction at this control point can be computed by using the formula

$$\begin{aligned} U_n^- &= \cos\theta U_x^- + \sin\theta U_y^-, && (3.87)\\ U_n^+ &= \rho\cos\theta U_x^- + \rho\sin\theta U_y^- + v/\beta^+, && (3.88)\end{aligned}$$

where the first equation (3.87) comes from the coordinate transformation defined in (3.64) while the second one (3.88) is simply the result of expanding the flux jump condition in (1.2). In a similar way, one can derive the formula for the tangential derivatives

$$\begin{aligned} U_\tau^- &= -\sin\theta U_x^- + \cos\theta U_y^-, && (3.89)\\ U_\tau^+ &= -\sin\theta U_x^- + \cos\theta U_y^- + w', && (3.90)\end{aligned}$$

where the second equation (3.90) comes from jump relation (3.67). Assume the interpolation schemes for the gradients have the same forms as (3.83). Then take U_n^- and U_n^+ as examples. They can be determined from the linear system (5.76) by replacing the right-hand side vector with $[0, \cos\theta, \sin\theta, 0, 0, 0]^T$ and $[0, \rho\cos\theta, \rho\sin\theta, 0, 0, v/\beta^+]^T$ respectively. The interpolation schemes for computing the gradients in other directions can be derived in a similar way.

3.3.4 Discrete maximum principle and error analysis

In this section, we propose an alternative discrete maximum principle preserving scheme to solve the model problem. The difference from the original version is the way we solve the linear system (5.76). In terms of performance, this alternative approach is no better than the original one but has extra computational cost for solving the optimization problems. The only purpose of introducing this method is for the error analysis. In general, if efficiency and simplicity are preferred, the original algorithm will be the choice. Otherwise, if the theoretical convergence proof is important, this method may be considered. The following are the details.

In this alternative method, we impose the sign restrictions on the coefficients $\{\gamma' s\}$ from the system (5.76)

$$\begin{aligned} &\gamma_{i_k,j_k} \geq 0 \quad \text{if } (i_k, j_k) \neq (0,0),\\ &\gamma_{i_k,j_k} < 0 \quad \text{if } (i_k, j_k) = (0,0).\end{aligned} \tag{3.91}$$

The coefficients are determined by the quadratic programming problem

$$\min_{\gamma}\{\frac{1}{2}\gamma^T \mathbf{I}\gamma\}, \quad \text{s.t.} \begin{cases} \mathbf{C}\gamma = \mathbf{a} \\ \gamma_{i_k,j_k} \geq 0 \quad \text{if } (i_k, j_k) \neq (0,0), \\ \gamma_{i_k,j_k} < 0 \quad \text{if } (i_k, j_k) = (0,0), \end{cases} \tag{3.92}$$

where $\mathbf{I}$ is identity matrix, $\mathbf{C}\gamma = \mathbf{a}$ is the linear system (5.76). We conjecture that the solution for this optimization problem always exists when the mesh is sufficiently fine and verify it by numerical tests.

Conjecture 3.1. *If the mesh size h is small enough and $\beta^+ \neq \beta^-$, the optimization problem (3.92) defined with five equalities and the sign constraints has a solution. The solution of the coefficients $\{\gamma_{i_k,j_k}\}$ also satisfies*

$$|\gamma_{i_k,j_k}| \leq \frac{C_u}{h^2} \max\left(\rho, \frac{1}{\rho}\right), \tag{3.93}$$

$$R = |R_1| + |R_2| \geq \frac{C_d|\rho - 1|}{h^2} > 0, \tag{3.94}$$

where constant C_u and C_d are constants, and

$$\begin{aligned} R_1 &= \gamma_{-1,1} + \gamma_{0,1} + \gamma_{1,1} - \gamma_{-1,-1} - \gamma_{0,-1} - \gamma_{1,-1}, \\ R_2 &= \gamma_{1,-1} + \gamma_{1,0} + \gamma_{1,1} - \gamma_{-1,-1} - \gamma_{-1,0} - \gamma_{-1,1}. \end{aligned} \tag{3.95}$$

The values R_1, R_2 defined above are the coefficients of the first derivative terms in the Taylor expansion of the form $\sum_{i_k,j_k} \gamma_{i_k,j_k} \Phi(x_{i+i_k}, y_{j+j_k})$ about the center point (x_i, y_j), and $\Phi(x, y) \in C^2(\Omega)$ is a test function used to prove the stability in Theorem 3.5. Notice that when the space step h is sufficiently small, the terms containing σ, κ, β_x, and β_y in (5.76) are higher order terms and can be neglected. Hence, in the numerical verification, we only need to consider the case where β is piecewise constant, $\sigma = 0$ and the curvature $\kappa = 0$ (the interface is a straight line), see Figure 3.3. Now Let $x^* = x_i + \alpha h$. The optimization problem only depends on three variables α, θ, and ρ. We choose different values of these parameters to verify the conjecture.

$$\begin{aligned} &\alpha_i = i\Delta\alpha, \quad \Delta\alpha = \frac{1}{M}, \quad i = 0, 1, \cdots, M, \\ &\theta_j = j\Delta\theta, \quad \Delta\theta = \frac{\pi}{2N}, \quad j = -N, -N+1, \cdots, N, \\ &\rho_k = 10^{k\Delta\rho}, \quad \Delta\rho = \frac{H}{L}, \quad k = -L, -L+1, \cdots, L. \end{aligned} \tag{3.96}$$

We choose $M = 100$, $N = 50$, $H = 5$, and $L = 50$. The jump ratio ranges from 10^{-5} to 10^5, which should cover most applications. The numerical tests show that the solution to the optimization problem always exists and the two conditions (3.93)-(3.94) hold. The results are presented in Figure 3.4.

Remark 3.5. *In the case where the mesh is coarse, the optimization solver may fail to give a solution or provide a wrong solution. We can either add a few more grid points that are close to the interface or loosen the equality restrictions by ignoring the last two equations in system*

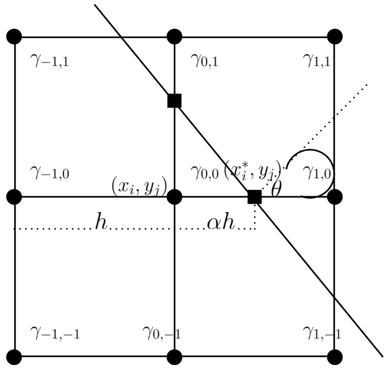

Figure 3.3: A diagram of the grid points involved in the optimization problem.

(5.76). *The breakdown happens only at a few grid points when the jump ratio is large or the grid is rather coarse. Loosening the equality restrictions at a few points does not affect global second-order accuracy due to the nature of the ellipticity. The details can be found in section 3.5 from [38].*

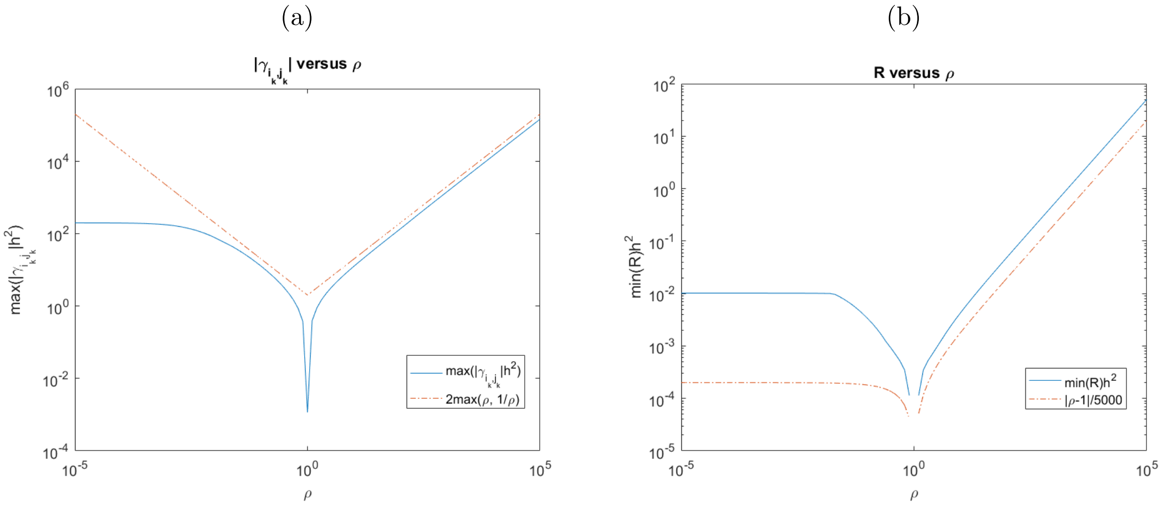

Figure 3.4: (a) The largest magnitude of the coefficients γ_{i_k,j_k} versus jump ratio ρ. (b) The size of variable R versus jump ratio ρ.

Now in our algorithm, the following conditions hold.

1. The discrete elliptic operator A_h has the form

$$(A_h\mathbf{U})_{i,j} = \sum_{i_k,j_k} c_{i_k,j_k} U_{i+i_k,j+j_k},$$

   where the indices i_k, j_k take values of $0, \pm 1$ and

$$c_{i_k,j_k} \geq 0 \quad \text{if } (i_k, j_k) \neq (0,0), \quad c_{i_k,j_k} < 0 \quad \text{if } (i_k, j_k) = (0,0).$$

2. The discrete elliptic operator A_h satisfies a discrete maximum principle, i.e.

$$|c_{0,0}| \geq \sum_{(i_k,j_k)\neq(0,0)} c_{i_k,j_k}.$$

3. The discrete set of interior points J_Ω is connected.

4. At least one of the equations from system (3.86) involves a boundary value given by Dirichlet boundary condition.

The following lemma, which is similar to Theorem 6.2 of Morton and Mayer [56], is used to prove second order convergence of the solution.

Lemma 3.1. *Assume that the following conditions hold.*

1. *J_Ω can be partitioned into two disjoint regions*

$$J_\Omega = J_1 \cup J_2, \quad J_1 \cap J_2 \neq \emptyset. \tag{3.97}$$

2. *There exists a nonnegative mesh function Φ define on J_Ω satisfying*

$$(A_h\Phi)_{i,j} \geq C_1 > 0 \quad \forall (x_i, y_j) \in J_1, \quad (A_h\Phi)_{i,j} \geq C_2 > 0 \quad \forall (x_i, y_j) \in J_2. \tag{3.98}$$

3. *The truncation error at a grid point (x_i, y_j) satisfies*

$$|T_{i,j}| \leq T_1 \quad \forall (x_i, y_j) \in J_1, \quad |T_{i,j}| \leq T_2 \quad \forall (x_i, y_j) \in J_2. \tag{3.99}$$

Then the error is bounded by

$$|e_{i,j}| \leq \left[\max_{A \in J_{\partial\Omega}} \Phi_A\right] \max\left\{\frac{T_1}{C_1}, \frac{T_2}{C_2}\right\}. \tag{3.100}$$

In the following theorem, we will use Lemma 3.1 to prove the stability of the 2D algorithm using an optimization solver for the case where β is piecewise constant and $\sigma = 0$. Let J_1 and J_2 be the sets of regular and irregular grid points respectively. In our algorithm, the truncation error T_1 at regular grid point is $O(h^2)$ while that error T_2 at irregular one is $O(h)$. The key idea is to find two lower bounds $C_1 \sim O(1)$ and $C_2 \sim O(1/h)$ such that T_1/C_1 and T_2/C_2 are $O(h^2)$.

Theorem 3.5. *For elliptic interface problem* (1.1)-(1.2), *assume the following.*

1. *The coefficient β is piecewise constant, $\beta^+ \neq \beta^-$, and $\sigma = 0$.*
2. *The quadratic programming problem has a solution $\{\gamma_{i_k,j_k}\}$ at every irregular grid point.*
3. *There exist constants C_3 and C_4 such that*

$$|\gamma_{i_k,j_k}| \leq \frac{C_3}{h^2} \max\left(\rho, \frac{1}{\rho}\right), \quad R \geq \frac{C_4|\rho - 1|}{h^2} > 0, \tag{3.101}$$

where R is defined in (3.95).

Then the numerical method has second order convergence rate. i.e.

$$\|u(x_i, y_j) - \mathbf{U}_{i,j}\|_\infty \leq C_5 h^2. \tag{3.102}$$

Proof. Consider the nonnegative (not strictly) convex functions

$$\Phi^1(x, y) = x^2, \quad \Phi^2(x, y) = y^2, \tag{3.103}$$

and the nonnegative strictly convex function

$$\Phi(x, y) = \Phi^1(x, y) + \Phi^2(x, y) = x^2 + y^2. \tag{3.104}$$

Let J_1 be the set of regular grid points and J_2 be that of irregular grid points. We have

$$(A_h\Phi)_{i,j} \geq 4\beta^\pm > 0 \quad \forall (x_i, y_j) \in J_1, \tag{3.105}$$

where the "+" or "−" sign is chosen depending on whether (x_i, y_j) lines on the "+" or "−" side of Γ. If $(x_i, y_j) \in J_2$, then we have

$$(A_h\Phi)_{i,j} \geq 4\beta^\pm + \sum_{i_k,j_k} \gamma_{i_k,j_k} \Phi^1_{i+i_k,j+j_k} + \sum_{i_k,j_k} \gamma_{i_k,j_k} \Phi^2_{i+i_k,j+j_k}, \tag{3.106}$$

where $4\beta^\pm$ is the result from the standard central difference stencil while the rest parts are the correction terms. Applying the Taylor expansion on $\{\Phi^1_{i+i_k,j+j_k}\}$ about the center point (x_i, y_j)

yields

$$\begin{aligned}\sum_{i_k,j_k}\gamma_{i_k,j_k}\Phi^1_{i+i_k,j+j_k} &= \sum_{i_k,j_k}\gamma_{i_k,j_k}\Phi^1_{i,j} + hR_1(\Phi^1_x)_{i,j} + O(1)\\ &= hR_1(\Phi_x)_{i,j} + O(1)\\ &\geq 0,\end{aligned} \tag{3.107}$$

where the second equality comes from the fact $\sum_{i_k,j_k}\gamma_{i_k,j_k} = 0$ and the last inequality is true because Φ^1 is convex and coefficients $\{\gamma_{i_k,j_k}\}$ also satisfy the discrete maximum principle, see (3.92). In a similar way, we have

$$\begin{aligned}\sum_{i_k,j_k}\gamma_{i_k,j_k}\Phi^2_{i+i_k,j+j_k} &= hR_2(\Phi_y)_{i,j} + O(1)\\ &\geq 0.\end{aligned} \tag{3.108}$$

Notice that $\gamma_{i_k,j_k} \sim O(h^2)$ and hence $hR_1(\Phi^1_x)_{i,j}, hR_2(\Phi_y)_{i,j} \sim O(1/h)$, the size of which dominates in the correction terms. Combining (3.107)-(3.108) yields

$$\begin{aligned}(A_h\Phi)_{i,j} &\geq 4\beta^{\pm} + hR1(\Phi_x)_{i,j} + hR_2(\phi_y)_{i,j} + O(1)\\ &\geq h\cdot R\cdot|\min(\Phi_x,\Phi_y)| + O(1)\\ &\geq \frac{C_4|\rho-1|}{h}|\min(\Phi_x,\Phi_y)| + O(1)\\ &> \frac{C_6}{h} + O(1) > 0.\end{aligned} \tag{3.109}$$

Now by Lemma 3.1 we have

$$\frac{|T_{i,j}|}{(A_h\Phi)_{i,j}} \leq \frac{C_1h^2}{4\beta^{\pm}} \quad \forall(x_i,y_j)\in J_1, \tag{3.110}$$

$$\frac{|T_{i,j}|}{(A_h\Phi)_{i,j}} \leq \frac{C_2}{C_6}h^2 \quad \forall(x_i,y_j)\in J_2, \tag{3.111}$$

and hence

$$\|u(x_i,y_j) - \mathbf{U}_{i,j}\|_\infty \leq C_5h^2, \tag{3.112}$$

□

Remark 3.6. *By Conjecture 3.1, the second and third conditions in Theorem 3.5 are satisfied if the first condition hold. Hence, the second order convergence of the solution is proved for the case where β is piecewise constant and $\sigma = 0$. For more general problems, σ, β_x, and β_y are higher order terms in* (5.76)*. So, the solution will be asymptotically second order convergent.*

The second order convergence proof for the gradient is still an open question for the direct IIM, and we only demonstrate it in the numerical tests.

3.3.5 An outline of the algorithm

In this section, we give an outline of our algorithm below.

Step 1: Generate a uniform mesh on the domain $\Omega = (a, b) \times (c, d)$ and use level set function to represent the interface.

Step 2: Determine the irregular grid points using equation (5.17)-(5.18) and calculate the coordinates of the control points which are the intersection points of interface and grid lines.

Step 3: Use the standard central difference scheme at the regular grid points.

Step 4: Solve the linear system (5.76) to determine the interpolation schemes for the correction terms $C^x_{i,j}$ and $C^y_{i,j}$ in (3.61) and hence the finite difference stencil in equation (3.55)-(3.59) at the irregular grid points. (The optimization can also be applied by solving the quadratic programming problem (3.92) to get a discrete maximum principle preserving scheme.)

Step 5: Solve the system of linear equations (3.86) to get second order numerical solution

Step 6: Replace $C^x_{i,j}$ in (3.82)-(3.83) with $U^{\pm}_n$ defined in (3.87)-(3.88) and the right-hand side vector in (5.76) with $[0, \cos\theta, \sin\theta, 0, 0, 0]^T$ and $[0, \rho\cos\theta, \rho\sin\theta, 0, 0, v/\beta^+]^T$ for U^-_n and U^+_n respectively. Then the solution $\{\gamma_{i_k,j_k}\}$ solved from linear system (5.76) is the set of coefficients used to interpolate the normal derivatives at the interface. The interpolation schemes for the gradients in other directions can be derived in a similar way.

3.4 Numerical examples

We present a variety of numerical experiments to show the performance of the new direct IIM approaches. All examples are computed with the double precision and are performed on a laptop with Intel(R) Core(TM) i7-4650U CPU, 1.70GHz, 8.00 GB memory. We also list CPU time(s) in the tables. We present errors in L^∞ norm and estimate the convergence order r using (2.65). We use Algebraic Multigrid (AMG) preconditioned GMRES iterative method to solve the discrete linear system. The tolerance is set to be 10^{-13} and the initial values is $\mathbf{0}$ in all computations. In all tables listed in this section, we use "Iter" to represent the number of GMRES iterations, "N_b" the number of control points, "N" the number of mesh intervals in each direction of the rectangular domain, and "CPU(s)" the run time in seconds. In the following examples, we will test each of the examples by using the original algorithm without the optimization. The grid refinement analysis is performed to demonstrate the second order convergence for both solution and gradient at the interface. We also conduct the eigenvalue analysis to show that the inverse of the coefficient matrix A_h is norm-bounded in L^2 norm and hence the method is stable. For the modified algorithm using the optimization, we test it on Example 3.3 and 3.4 to compare with the original algorithm. Unless otherwise stated, the following numerical results are from the original algorithm without the optimization.

Example 3.1.

$$u(\boldsymbol{x}) = \begin{cases} \sin(x+y) & \textit{in } \Omega^-, \\ \log(x^2+y^2+1) & \textit{in } \Omega^+, \end{cases} \tag{3.113}$$

with

$$\beta(\boldsymbol{x}) = \begin{cases} \sin(x+y)+2 & \textit{in } \Omega^-, \\ \cos(x+y)+2 & \textit{in } \Omega^+, \end{cases} \qquad \sigma(\boldsymbol{x}) = 0, \tag{3.114}$$

where the interface is the zero level set of $\phi(x,y) = \sqrt{x^2+y^2} - 0.5$*, and* $\Omega = (-1,1) \times (-1,1)$*. The source term* $f(\mathbf{x})$*, and the interface jump conditions:* $[u]$ *and* $[\beta u_n]$ *are derived from the exact solutions.*

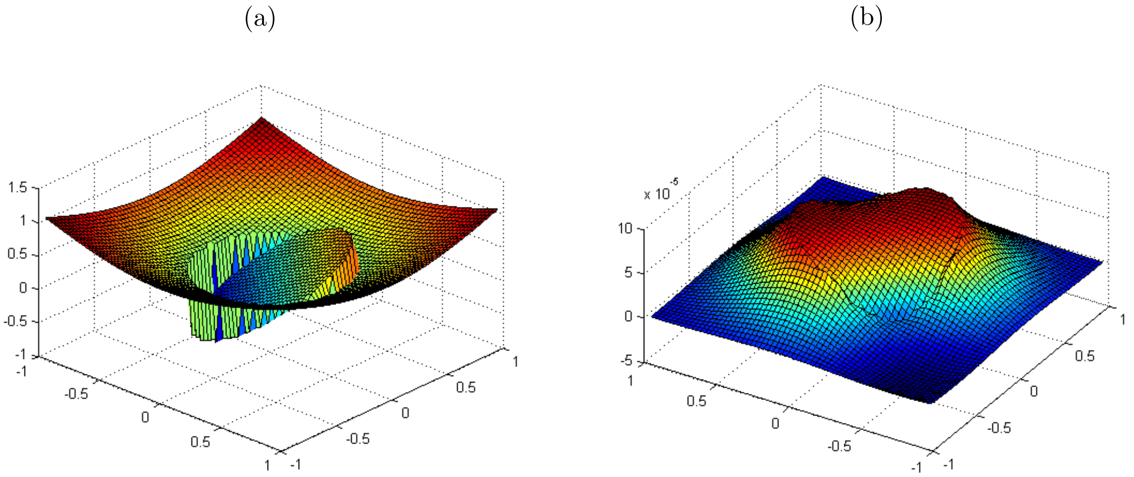

Figure 3.5: (a): The solution plot of Example 3.1 with modest jump. (b): The error plot of the computed solution. The error seems to be piecewise smooth as well, which is important for accurate gradient computation.

Table 3.1: A grid refinement analysis for Example 3.1 with a modest variable jump in the coefficient, where $E(u)$ is the error of the solution and $E(u_n)$, $E(u_\tau)$ are the errors of normal and tangential derivatives at the interface respectively. All the errors are presented in L_∞ norm.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	1.04×10^{-3}		9.39×10^{-3}		5.59×10^{-3}		1	28	0.06
32	3.78×10^{-4}	1.46	2.41×10^{-3}	1.96	1.72×10^{-3}	1.70	1	60	0.10
64	8.00×10^{-5}	2.24	6.36×10^{-4}	1.92	7.98×10^{-4}	1.11	1	124	0.45
128	2.31×10^{-5}	1.79	1.89×10^{-4}	1.75	2.13×10^{-4}	1.91	17	252	0.34
256	5.65×10^{-6}	2.03	4.75×10^{-5}	1.99	4.94×10^{-5}	2.11	20	508	0.77
512	1.52×10^{-6}	1.89	1.35×10^{-5}	1.82	1.33×10^{-5}	1.89	19	1020	2.09

Table 3.2: A grid refinement analysis for Example 3.1 with a large variation in the jump ratio of the coefficient.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	1.28×10^{-2}		5.31×10^{-2}		1.54×10^{-2}		1	28	0.05
32	3.37×10^{-3}	1.93	1.50×10^{-2}	1.82	4.39×10^{-3}	1.81	1	60	0.12
64	8.90×10^{-4}	1.92	4.07×10^{-3}	1.88	1.28×10^{-3}	1.78	1	124	0.47
128	2.49×10^{-4}	1.84	1.18×10^{-3}	1.79	3.58×10^{-4}	1.84	19	252	0.39
256	6.28×10^{-5}	1.99	3.03×10^{-4}	1.96	9.67×10^{-5}	1.89	22	508	0.86
512	1.54×10^{-5}	2.03	7.72×10^{-5}	1.97	2.85×10^{-5}	1.76	22	1020	2.30

This is an almost arbitrary example with a piecewise smooth non-linear solution, a variable coefficient with discontinuity along the interface. We present a grid refinement analysis in Table 3.1. The second column is the maximum error of the solution while the third column is the approximate convergence order. The fourth and sixth column are the errors of the normal and tangential derivatives at the interface respectively and are defined as the averages of the L_∞ errors from each side of the interface Ω^+ and Ω^-, i.e.

$$E(u_n) = \frac{1}{2}(\|\mathbf{E}(u_n^+)\|_\infty + \|\mathbf{E}(u_n^-)\|_\infty), \quad E(u_\tau) = \frac{1}{2}(\|\mathbf{E}(u_\tau^+)\|_\infty + \|\mathbf{E}(u_\tau^-)\|_\infty).$$

The fifth and seventh columns are the approximate convergence order of the computed normal and tangential derivatives. The last three columns are number of GMRES iterations, number of control points, and the total CPU time in seconds, respectively. We can observe from Table 3.1 that the new direct IIM has average second order accuracy in both the solution globally and the gradient at the interface from each side of the interface. We also see that it only takes one iteration to solve the linear system when the mesh is coarse and the number of GMRES iterations is independent of the mesh size when N is large enough. From Figure 3.6, one can see that the largest eigenvalue of the coefficient matrix is negative and bounded away from zero. Therefore, the inverse of the coefficient matrix A_h is bounded in L^2 norm by a constant and the method is stable. The total CPU time also shows the method is very fast.

Now we use the same exact solution and interface and $\sigma = 0$ but with a large jump in the coefficient along the interface

$$\beta(\mathbf{x}) = \begin{cases} e^{5x} & \text{in } \Omega^-, \\ \sin(x+y)+2 & \text{in } \Omega^+. \end{cases} \tag{3.115}$$

Notice that β^- ranges from 6.7379×10^{-3} to 1.4841×10^2 along the interface, and hence the jump ratio changes dramatically. Nevertheless, all the nice features are the same as the previous example as shown in Table 3.2 and Figure 3.6.

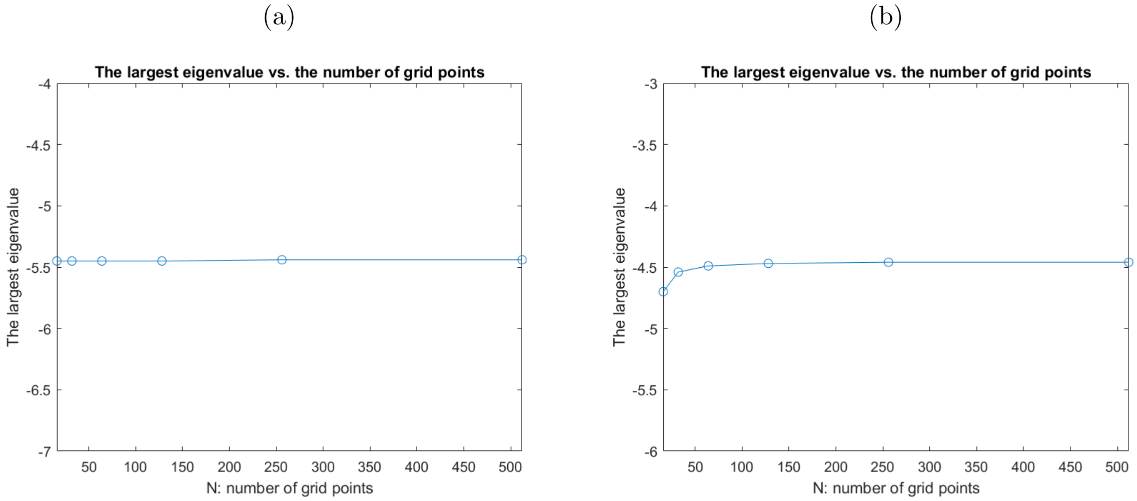

Figure 3.6: (a): Plot of the largest eigenvalue versus the number of grid points for Example 3.1 with modest jump using the algorithm without the optimization. (b): Plot of the largest eigenvalue versus the number of grid points for Example 3.1 with large jump using the algorithm without the optimization.

Example 3.2.

$$u(\boldsymbol{x}) = \begin{cases} x^2 - y^2 & \textit{in } \Omega^-, \\ \sin(x)\cos(y) & \textit{in } \Omega^+, \end{cases} \tag{3.116}$$

Table 3.3: A grid refinement analysis for Example 3.2 with $\beta^- = 1$ and $\beta^+ = 1000$.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	3.21×10^{-4}		1.65×10^{-3}		7.60×10^{-3}		1	20	0.03
32	1.37×10^{-4}	1.23	7.45×10^{-4}	1.15	2.28×10^{-3}	1.74	1	44	0.13
64	3.97×10^{-5}	1.79	1.93×10^{-4}	1.95	6.86×10^{-4}	1.73	1	92	0.22
128	1.01×10^{-5}	1.98	6.01×10^{-5}	1.69	1.80×10^{-4}	1.93	20	188	0.29
256	2.63×10^{-6}	1.94	1.66×10^{-5}	1.86	5.22×10^{-5}	1.79	24	380	0.75
512	6.85×10^{-7}	1.94	4.00×10^{-6}	2.05	1.32×10^{-5}	1.98	25	764	2.23

Table 3.4: A grid refinement analysis for Example 3.2 with $\beta^- = 1000$ and $\beta^+ = 1$.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	8.42×10^{-4}		4.26×10^{-3}		9.77×10^{-4}		1	20	0.03
32	1.66×10^{-4}	2.35	1.37×10^{-3}	1.64	8.89×10^{-4}	0.14	1	44	0.11
64	4.66×10^{-5}	1.83	2.89×10^{-4}	2.24	3.15×10^{-4}	1.50	1	92	0.17
128	8.77×10^{-6}	2.41	7.15×10^{-5}	2.01	8.17×10^{-5}	1.95	25	188	0.32
256	2.21×10^{-6}	1.99	1.92×10^{-5}	1.90	1.85×10^{-5}	2.14	31	380	0.84
512	5.40×10^{-7}	2.04	5.11×10^{-6}	1.91	5.31×10^{-6}	1.80	33	764	2.81

$$\beta(\boldsymbol{x}) = \begin{cases} 1 & in\ \Omega^-, \\ 1000 & in\ \Omega^+, \end{cases} \quad or \quad \begin{cases} 1000 & in\ \Omega^-, \\ 1 & in\ \Omega^+, \end{cases} \qquad \sigma(\boldsymbol{x}) = 0, \tag{3.117}$$

where the interface is an ellipse represented by zero level set as

$$\phi(x, y) = \frac{x^2}{0.5^2} + \frac{y^2}{0.25^2} - 1, \tag{3.118}$$

and the domain is $\Omega = [-1, 1] \times [-1, 1]$. *The source term* $f(\mathbf{x})$, *and the interface jump conditions:* $[u]$ *and* $[\beta u_n]$ *are derived from the exact solutions.*

These are two extreme cases where the piecewise function $\beta(\mathbf{x})$ has a huge jump at the interface. The jump ratio $\rho = \beta^-/\beta^+$ is 0.001 or 1000 in this example. The grid refinement for these two cases are shown in Table 3.3 and 3.4. From the tables, we observe the average second order convergence rate for both the solution globally and gradient at the interface. Compared with Example 3.1, we find that it requires finer grid to resolve the problem for extremely small jump ratio ρ.

Example 3.3. *A general interface example with* $\sigma = 0$ *and a complicated interface.*

Table 3.5: A grid refinement analysis for Example 3.3 with a large variation in the jump ratio and a complicated interface using the algorithm without the optimization.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	1.50×10^{-2}		5.21×10^{-2}		3.25×10^{-2}		1	48	0.09
32	4.09×10^{-3}	1.87	1.59×10^{-2}	1.71	1.18×10^{-2}	1.46	1	108	0.21
64	1.18×10^{-3}	1.79	5.14×10^{-3}	1.63	3.82×10^{-3}	1.63	1	212	0.74
128	2.47×10^{-4}	2.26	1.24×10^{-3}	2.05	1.03×10^{-3}	1.89	21	428	0.81
256	6.40×10^{-5}	1.95	3.09×10^{-4}	2.00	3.02×10^{-4}	1.77	27	860	1.33
512	1.66×10^{-5}	1.95	8.51×10^{-5}	1.86	7.21×10^{-5}	2.07	25	1728	3.42

Table 3.6: A grid refinement analysis for Example 3.3 with a large variation in the jump ratio and a complicated interface using the algorithm with the optimization.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	7.60×10^{-3}		3.72×10^{-2}		3.19×10^{-2}		1	48	0.09
32	3.76×10^{-3}	1.02	2.25×10^{-2}	0.73	1.12×10^{-2}	1.51	1	108	0.25
64	8.95×10^{-4}	2.07	6.44×10^{-3}	1.80	2.71×10^{-3}	2.05	1	212	0.79
128	2.13×10^{-4}	2.07	1.71×10^{-3}	1.91	7.78×10^{-4}	1.80	20	428	0.86
256	5.74×10^{-5}	1.89	4.42×10^{-4}	1.95	2.15×10^{-4}	1.86	30	860	1.46
512	1.44×10^{-5}	2.00	1.18×10^{-4}	1.91	5.40×10^{-5}	1.99	41	1728	4.56

In this example, we use the same exact solution in Example 3.1, but the interface is a 5-leaf rose shape, which is represented by zero level set function

$$\phi(\mathbf{x}) = (0.5 + 0.2\sin(5\theta)) - r, \tag{3.119}$$

The interface is both convex and concave and has relatively large curvature at some places. We choose the coefficient $\beta(\mathbf{x})$ as the one used in Example 3.1, which is expressed in equation (4.9). The domain is $\Omega = (-1, 1) \times (-1, 1)$. Notice that, in this example, we get a huge change in the jump ratio and the interface has large curvature at some places. We test this example using both the algorithms with and without the optimization. The grid refinement for these two cases are shown in Table 3.5 and 3.6. The results from both algorithms look similar and once again we observe that when the mesh is sufficiently fine, both the solution and the gradient are second order accurate. For the algorithm with the optimization, it takes more GMRES iterations to solve the system when the mesh is sufficiently fine. Although the coefficient matrix produced by this algorithm is M-matrix,

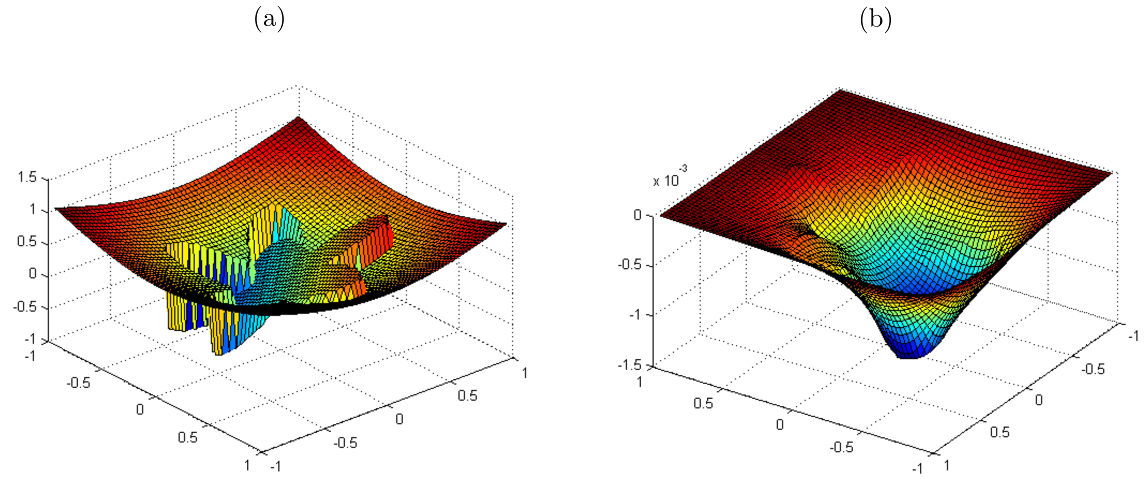

Figure 3.7: (a): The computed solution plot of Example 3.3. (b): The error plot of Example 3.3. The error seems to be piecewise smooth.

it does not necessarily imply better computational efficiency. For the original algorithm without the optimization, it runs faster in general. Also the largest eigenvalues of the coefficient matrix produced by this algorithm is negative and bounded away from zero, which is shown in Figure 3.9. This demonstrates the stability of the original algorithm in L^2 norm.

Table 3.7: A grid refinement analysis for Example 3.4 for a general elliptic interface problem with a complicated interface using the algorithm without the optimization.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	5.03×10^{-3}		2.11×10^{-2}		2.24×10^{-2}		1	48	0.09
32	1.71×10^{-3}	1.56	6.30×10^{-3}	1.74	5.65×10^{-3}	1.99	1	108	0.22
64	4.41×10^{-4}	1.95	1.54×10^{-3}	2.03	1.40×10^{-3}	2.01	1	212	0.76
128	1.10×10^{-4}	2.01	4.32×10^{-4}	1.83	3.66×10^{-4}	1.94	16	428	0.67
256	2.78×10^{-5}	1.98	1.12×10^{-4}	1.95	8.53×10^{-5}	2.10	20	860	1.19
512	6.91×10^{-6}	2.01	2.87×10^{-5}	1.96	2.30×10^{-5}	1.89	19	1728	3.02

Example 3.4. *A general example with $\sigma \neq 0$.*

We present a more general example with a non-zero $\sigma(\mathbf{x})$ term with the same general

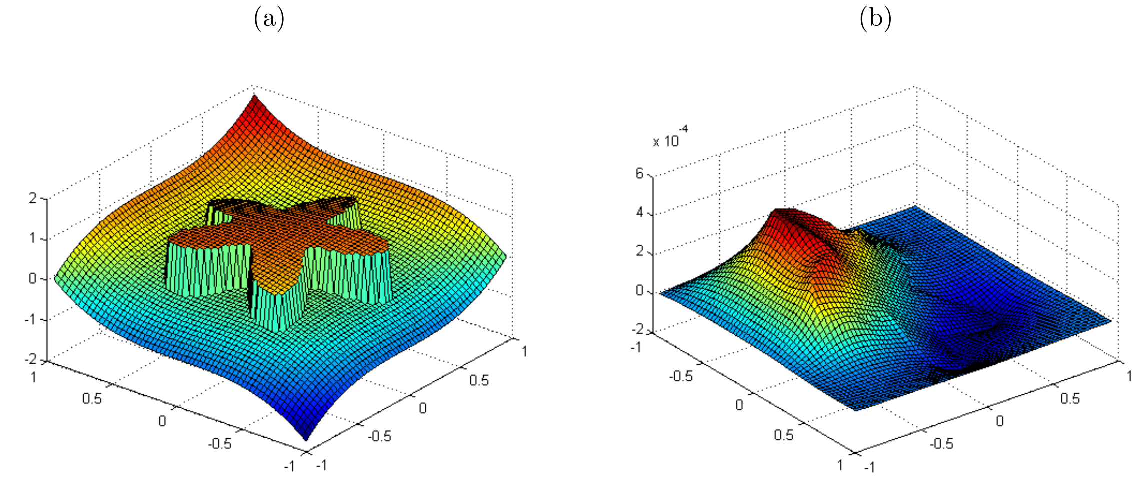

Figure 3.8: (a): The computed solution plot of Example 3.4. (b): The error plot of Example 3.4. The error seems to be piecewise smooth.

Table 3.8: A grid refinement analysis for Example 3.4 for a general elliptic interface problem with a complicated interface using the algorithm with the optimization.

N	$E(u)$	r	$E(u_n)$	r	$E(u_\tau)$	r	Iter	N_b	CPU(s)
16	6.75×10^{-3}		2.22×10^{-2}		2.35×10^{-2}		10	48	0.08
32	1.82×10^{-3}	1.89	6.36×10^{-3}	1.80	6.26×10^{-3}	1.91	11	108	0.23
64	4.79×10^{-4}	1.93	1.57×10^{-3}	2.02	1.56×10^{-3}	2.00	11	212	0.77
128	1.19×10^{-4}	2.01	4.58×10^{-4}	1.78	4.12×10^{-4}	1.92	11	428	0.92
256	3.03×10^{-5}	1.97	1.18×10^{-4}	1.96	9.83×10^{-5}	2.08	12	860	1.29
512	7.54×10^{-6}	2.01	3.01×10^{-5}	1.97	2.66×10^{-5}	1.89	12	1728	3.19

interface (a five-leaf rose) as in Example 3.3. The true solution is

$$u = \begin{cases} \cos(x+y) & \text{if} \quad (x,y) \in \Omega^-, \\ x^3 + y^3 & \text{if} \quad (x,y) \in \Omega^+, \end{cases} \tag{3.120}$$

with the coefficients

$$\beta = \begin{cases} x+y+3 & \text{if} \quad (x,y) \in \Omega^-, \\ \sin(x+y)+2 & \text{if} \quad (x,y) \in \Omega^+, \end{cases} \qquad \sigma = \begin{cases} x^2+y^2+1 & \text{if} \quad (x,y) \in \Omega^-, \\ e^{x+y} & \text{if} \quad (x,y) \in \Omega^+, \end{cases} \tag{3.121}$$

where again $\Omega = (-1,1) \times (-1,1)$.

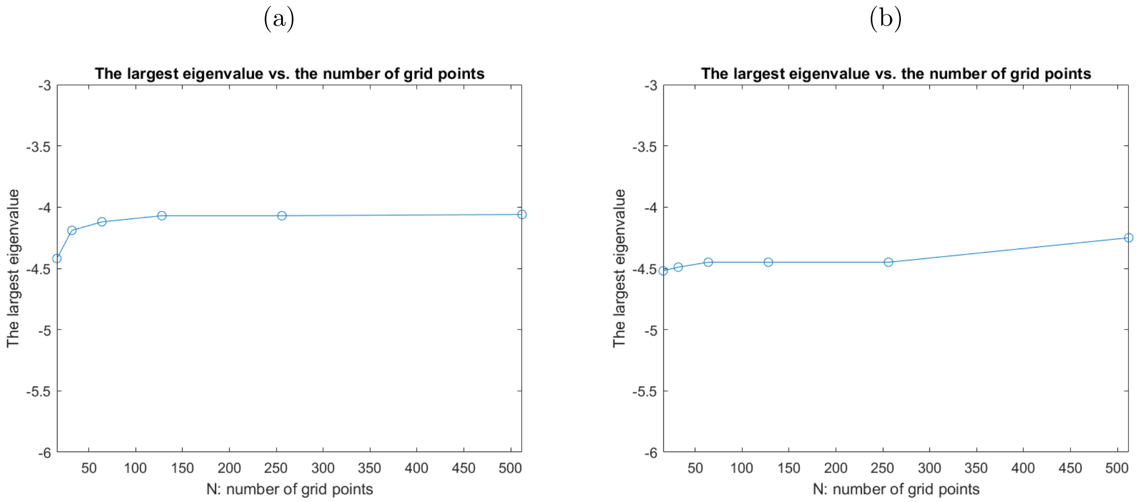

Figure 3.9: (a): Plot of the largest eigenvalue versus the number of grid points for Example 3.3 using the algorithm without the optimization. (b): Plot of the largest eigenvalue versus the number of grid points for Example 3.4 using the algorithm without the optimization.

This is a very general example for a self-adjoint elliptic interface problem with a nonlinear solution. The shape of the interface is complicated and σ has a finite discontinuity at the interface. We test this example using both the algorithms with and without the optimization. The results from the grid refinement analysis are listed in Table 3.7 and Table 3.8. One can see that the performance of both algorithms is similar. Again, we observe average second order convergence for the global solution and the gradient from each side of the interface. We do see that the mesh needs to be fine enough to resolve the solution and its gradient because of the complicated interface. Also, the number of GMRES iterations is independent of mesh size when N is large. For the original algorithm without the optimization, we plot the distribution of the largest eigenvalue of the coefficient matrix in Figure 3.9. From that, one can see that all the eigenvalues locate in the left part of the complex plane and the largest one is bounded away from zero. Numerically, we showed the stability of our approach without using the optimization. For the algorithm using the optimization, one can observe that its CPU time is slightly higher. Usually, this additional computational cost by using the optimization is neglectable compared with the cost from the linear solver.

Chapter 4

On Multi-scale ADI Methods for Parabolic PDEs with a Discontinuous Coefficient

In this chapter, two ADI methods based on augmented approaches are proposed to solve two-dimensional heat equation

$$u_t = (\beta u_x)_x + (\beta u_y)_y - f, \qquad (x,y) \in \Omega^+ \cup \Omega^- \setminus \Gamma, \tag{4.1}$$

$$u(x,y,0) = u_0(x,y), \qquad \text{Given BC on } \partial\Omega^+, \tag{4.2}$$

with a fixed interface $\Gamma \in C^2$, where $u_t = \frac{\partial u}{\partial t}$, $u_x = \frac{\partial u}{\partial x}$ and so on, BC stands for a boundary condition along $\partial\Omega^+$ which can be Dirichlet, Neumann, or mixed along different parts of the boundary $\partial\Omega^+$ (see Fig. 1.2 for an illustration). Two more conditions are needed at the interface Γ to close the system, typically the natural jump conditions which state that both the solution and the flux are continuous across the interface Γ. The coefficient is discontinuous and the domain has many inclusions with different material parameters.

The key to an ADI method is to discretize the PDE dimension by dimension. For a heat equation with a constant coefficient, say $\beta = 1$, this can be and has been done easily since we know the jump relations $[u], [u_x], [u_{xx}]$, and $[u], [u_y], [u_{yy}]$ from the problem. That is, w, v, and Γ directly, for example, we can write

$$\frac{u(x_{i-1},y_j) - 2u(x_i,y_j) + u(x_{i+1},y_j)}{h^2} \approx u_{xx}(x_i,y_j) - (C_x)_{ij}, \tag{4.3}$$

where $h^2(C_x)_{ij} = \pm\big([u] + [u_x](x_{i+1} - x^*) + (x_{i+1} - x^*)^2[u_{xx}]/2\big)$ assuming that x_{i-1} and x_i are from the same side of the interface different from x_{i+1}, x^* is the intersection of the interface Γ

with the grid line $y = y_j$. The correction term depends on known quantities w, v, and Γ linearly and is similar to the Taylor expansion except with the jump conditions. Thus the ADI method from time t^k to t^{k+1} can be written in the matrix-vector form as

$$A\,\mathbf{U}^{k+1} = \mathbf{F}_1^k - B_1\,[\mathbf{U}]^{k^*} - B_2\,[\mathbf{U}_\mathbf{n}]^{k^*} \tag{4.4}$$

where A is the classical ADI operator, $\mathbf{U}^{k+1}$ is the finite difference solution at t^{k+1}, typically $k^* = k + 1/2$ is the time level between k and $k+1$, and $B_1\,[\mathbf{U}]^{k^*}$ and $B_2\,[\mathbf{U}_\mathbf{n}]^{k^*}$ correspond to the correction terms resulted from $[u]$ and $[u_n]$, respectively. Note that $\mathbf{U}^k$ term has been absorbed into $\mathbf{F}_1^k$, and the dimension of $[\mathbf{U}]^{k^*}$ and $[\mathbf{U}_\mathbf{n}]^{k^*}$ are $O(N)$ compared with that of U which is $O(N^2)$.

However, if β has a jump across the interface Γ, then we do not have the dimensional jump relations anymore. the augmented strategy is to introduce one or several intermediate variables so that we can still use the ADI method. For example, we can set $\mathbf{Q}^{k^*} = [\mathbf{U}_\mathbf{n}]^{k^*}$ as unknown to have

$$A\,\mathbf{U}^{k+1} + B_2\mathbf{Q}^{k^*} = \mathbf{F}_1^k - B_1[\mathbf{U}]^{k^*} \tag{4.5}$$

where $\mathbf{Q}^{k^*}$ is an approximation to $[\mathbf{U}_\mathbf{n}]^{k^*}$ which should be chosen to satisfy the interface condition $[\beta u_n] = v$. In discretization, often a local interpolation is used to approximate the $[\beta u_n] = v$ which can be written as another matrix-vector form

$$C\,\mathbf{U}^{k+1} + D\mathbf{Q}^{k^*} = \mathbf{F}_2^k \tag{4.6}$$

for some sparse matrices C and D. The solution $(\mathbf{U}^{k+1}, \mathbf{Q}^{k^*})$ satisfies the following system of equations,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{U}^{k+1} \\ \mathbf{Q}^{k^*} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1^k \\ \mathbf{F}_2^k \end{bmatrix}. \tag{4.7}$$

The above system of equations provides a discretization process using the ADI method.

Another key of the augmented method is to solve the above system of equations efficiently. Note that the Schur complement for $\mathbf{Q}^{k^*}$ is

$$(D - CA^{-1}B)\mathbf{Q}^{k^*} = \mathbf{F}_2^k - CA^{-1}\mathbf{F}_1^k = \bar{\mathbf{F}}^k, \qquad \text{or} \qquad S\mathbf{Q}^{k^*} = \bar{\mathbf{F}}^k, \tag{4.8}$$

where $A^{-1}\mathbf{F}^k$ is the result of one step of the ADI for a regular problem. Often we can solve the Schur complement system efficiently since the dimension of the unknown is $O(N)$. For fixed interface and time step size, we just need to form S and perform the LU decomposition once. For time dependent problems and adaptive time step size, the GMRES method is recommended.

4.1 Two augmented formulations and jump relations

For simplicity of the discussion, we assume that the coefficient is a piecewise constant

$$\beta(\mathbf{x}) = \begin{cases} \beta^+ & \text{if } \mathbf{x} \in \Omega^+, \\ \beta^- & \text{if } \mathbf{x} \in \Omega^-, \end{cases} \tag{4.9}$$

and $\beta(\mathbf{x}) \geq \beta_0 > 0$. We also omit the time dependence at this moment but focus on the space discretization. In order to use an ADI method efficiently with the Immersed Interface Method (IIM) [35, 38], we need to get the jump relations in the coordinate directions such as $[\beta u]$, $[\beta u_x]$, $[\beta u_{xx}]$, $[\beta u_y]$, $[\beta u_{yy}]$. These jump relations are obtained using the original interface conditions (1.5), the partial differential equation (1.3), and a carefully selected augmented variable.

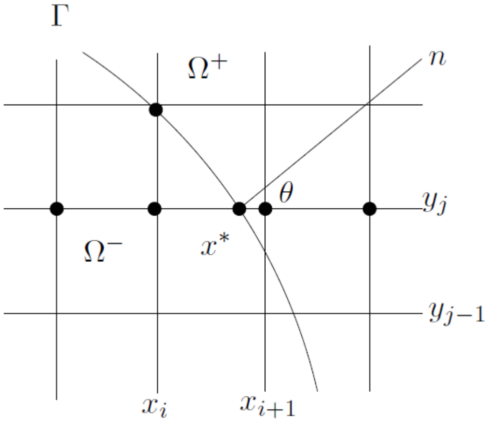

Figure 4.1: A diagram of the local coordinates system (ξ, η) at a point (X^*, Y^*) on the interface Γ and the geometry in the neighborhood.

It is easier to derive jump relations using the local coordinates in the normal and tangential directions. At a point $\mathbf{X}^* = (X^*, Y^*)$ on the interface Γ, we define the local coordinates as

$$\begin{aligned} \xi &= (x - X^*)\cos\theta + (y - Y^*)\sin\theta, \\ \eta &= -(x - X^*)\sin\theta + (y - Y^*)\cos\theta, \end{aligned} \tag{4.10}$$

where θ is the angle between the x-axis and the normal direction at (X^*, Y^*). In a neighborhood of this point, the interface lies roughly in the η-direction, so we can parameterize Γ locally by

$$\xi = \chi(\eta), \quad \eta = \eta, \qquad \text{with } \chi(0) = 0, \;\; \chi'(0) = 0, \;\; \text{and} \;\; \chi''(0) = \kappa, \tag{4.11}$$

where κ is the curvature at $\mathbf{X}^*$, see Figure 4.1 for an illustration. For convenience we use the same notation for the solution $u(x,y,t)$ and other quantities involved both in the original coordinates and the new local coordinates.

4.1.1 Setting $[\beta u]$ as an augmented variable

Our first ADI method is to set $[\beta u]$ as an unknown coupled with the original problem

$$\begin{aligned} &u_t = (\beta u_x)_x + (\beta u_y)_y - f, \qquad (x,y) \in \Omega^+ \cup \Omega^- \setminus \Gamma, \\ &[\beta u]_\Gamma = q, \qquad [\beta u_n]_\Gamma = v, \qquad (X,Y) \in \Gamma. \end{aligned} \tag{4.12}$$

The solution $u(\mathbf{x},t)$ above is a functional of $q(s)$ and can be denoted as $u_q(\mathbf{x},t)$. The solution of the original problem corresponds to a particularly solution $(q^*,\ u_{q^*}(\mathbf{x},t))$ such that

$$[u_{q^*}]_\Gamma = w. \tag{4.13}$$

Now assume that we know the jump $[\beta u] = q$ and $[\beta u_n] = v$, then we have almost immediately

$$\begin{aligned} [\beta u_x] &= [\beta u_\xi] \cos\theta - [\beta u]_\eta \sin\theta = v\,\cos\theta - \frac{dq}{d\eta}\sin\theta, \\ [\beta u_y] &= [\beta u_\xi] \sin\theta + [\beta u]_\eta \cos\theta = v\,\sin\theta + \frac{dq}{d\eta}\cos\theta. \end{aligned} \tag{4.14}$$

To get second order jump relations in the original coordinates, we have the following theorem.

Theorem 4.1. *Let $u(x,y) \in C^2(\Omega^\pm \setminus \Gamma)$ be the solution to (1.3), (1.5) where $\Gamma \in C^2$. Let (X,Y) be a point on the interface with the local coordinates (ξ,η) as defined in (4.10). Assume also that $[u] = w$, $[\beta u] = q \in C^2(\Gamma)$, and $[\beta u_n] = [\beta u_\xi] = v \in C^1(\Gamma)$, then we have the following jump relations of the second order derivatives.*

$$[\beta u_{\eta\eta}] = -\chi''\,v + \frac{d^2q}{d\eta^2},\ [\beta u_{\xi\eta}] = \chi''\,\frac{dq}{d\eta} + \frac{dv}{d\eta},\ [\beta u_{\xi\xi}] = \chi''\,v - \frac{d^2q}{d\eta^2} + [f] + \frac{dw}{dt}. \tag{4.15}$$

Proof: From the governing equation (1.3), we have

$$[(\beta u_x)_x + (\beta u_y)_y] = [f] + [u_t]\,. \tag{4.16}$$

Applying Theorem 3.1 in [38], we get the jump relations above. □

Remark 4.1. *Usually we know $[u]$ and $[\beta u_n]$, but not $[\beta u]_\tau = [\beta u]_\eta = \frac{dq}{d\eta}$. To utilize the jump relations, we set $[\beta u]$ or $\frac{d[\beta u]}{d\tau} = \frac{d[\beta u]}{d\eta}$ (see next sub-section) as an augmented variable to* *decouple the two scales. The interface quantities have only one independent variable, say the arc-length s, thus we use $d(:)/d\eta$, for example, for the surface derivatives.*

With these jump relations, we can get the jump relations in the original coordinate directions as the following,

$$\begin{aligned}
[\beta u_{xx}] &= [\beta u_{\xi\xi}] \cos^2\theta - 2[\beta u_{\xi\eta}] \cos\theta \sin\theta + [\beta u_{\eta\eta}] \sin^2\theta, \\
[\beta u_{yy}] &= [\beta u_{\xi\xi}] \sin^2\theta + 2[\beta u_{\xi\eta}] \cos\theta \sin\theta + [\beta u_{\eta\eta}] \cos^2\theta.
\end{aligned} \tag{4.17}$$

4.1.2 Setting $d[\beta u]/d\tau$ as an augmented variable

Our second ADI method is to set $\frac{d[\beta u]}{d\tau} = q(s)$, the tangential derivative of u as an unknown coupled with the original problem

$$\begin{aligned}
& u_t = (\beta u_x)_x + (\beta u_y)_y - f, \qquad (x,y) \in \Omega^+ \cup \Omega^- \setminus \Gamma, \\
& \left.\frac{d[\beta u]}{d\tau}\right|_\Gamma = q, \qquad [\beta u_n]_\Gamma = v, \qquad (X,Y) \in \Gamma.
\end{aligned} \tag{4.18}$$

The solution $u(\mathbf{x},t)$ above is a functional of $q(s)$ and can be denoted as $u_q(\mathbf{x},t)$. The solution of the original problem corresponds to a particularly solution (q^*, $u_{q^*}(\mathbf{x},t)$) such that

$$\left[\frac{d\,[u_{q^*}]}{d\tau}\right]_\Gamma = \frac{dw}{ds}. \tag{4.19}$$

Note that $\frac{d[\beta u]}{d\tau} = [\beta \nabla u \cdot \boldsymbol{\tau}] = [\beta u_\tau]$ if β is a piecewise constant. While this formulation is only slightly different from the previous case, numerical results show dramatic differences in performance. Now assume we know the jump $\frac{d[\beta u]}{d\tau} = q$ and $[\beta u_n] = v$, then we have almost immediately

$$\begin{aligned}
[\beta u_x] &= [\beta u_\xi] \cos\theta - [\beta u_\eta] \sin\theta = v\,\cos\theta - q\,\sin\theta \\
[\beta u_y] &= [\beta u_\xi] \sin\theta + [\beta u_\eta] \cos\theta = v\,\sin\theta + q\,\cos\theta,
\end{aligned} \tag{4.20}$$

which are different from (4.14) in terms of the augmented variable and their surface derivatives.

It is advantageous to consider βu as a whole because the PDE in the spatial derivatives and the flux jump conditions are given in this scale. To get second order jump conditions, we have the following theorem.

Theorem 4.2. *Let $u(x,y) \in C^2(\Omega^\pm \setminus \Gamma)$ be the solution to (1.3) and (1.5), where $\Gamma \in C^2$. Let (X,Y) be a point on the interface with the local coordinates (ξ,η) as defined in (4.10). Assume also that $[u] = w \in C^2(\Gamma)$, $[\beta u]_\eta = q \in C^1(\Gamma)$ and $[\beta u]_\xi = v \in C^1(\Gamma)$, then we have the* *following jump relations of the second order derivatives.*

$$[\beta u_{\eta\eta}] = -\chi''\, v + \frac{dq}{d\eta}, \; [\beta u_{\xi\eta}] = \chi''\, q + \frac{dv}{d\eta}, \; [\beta u_{\xi\xi}] = \chi''\, v - \frac{dq}{d\eta} + [f] + \frac{dw}{dt}. \tag{4.21}$$

All those jump relations are somewhat similar. We only need to pay attention to which interface variables are known and which one is the augmented one.

4.2 The ADI algorithms using different augmented variables

There are two major steps in the algorithm design. One is the ADI method given an augmented variable $q(s)$; and another step is to get the system of equations for the augmented variable $q(s)$. For more details about an augmented strategy, we refer the reader to [38, 44].

The domain Ω is assumed to be rectangular, say $[a,b] \times [c,d]$. We use a uniform $M \times N$ grid with $x_i = a + ih,\ y_j = c + jh,\ h = (b-a)/M = (d-c)/N$, for simplicity of the presentation. We use $u_{ij}^k \approx u(x_i, y_j)$ as the finite difference solution at the grid point (x_i, y_j) and the time level t^k.

We use a level set representation for an interface Γ in the domain. That is, the interface is the zero level of a Lipschitz continuous function $\varphi(x,y)$. For example, for a unit circle interface $x^2 + y^2 = 1$, one of (and the ideal) level set functions is $\varphi(x,y) = \sqrt{x^2+y^2} - 1$. In general, the signed distance function is recommended for numerical computations. In discretization, the level set function is determined explicitly by its values at the grid points $\varphi_{ij} = \varphi(x_i, y_j)$, for example, for the unit circle interface, $\varphi_{ij} = \sqrt{x_i^2 + y_j^2} - 1$.

We classify a grid point (x_i, y_j) as a *regular* or *irregular* using the level set function. At a grid point (x_i, y_j), define

$$\begin{aligned} \varphi^{max} &= \max\left\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}, \\ \varphi^{min} &= \min\left\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}. \end{aligned} \tag{4.22}$$

If $\varphi^{max}\varphi^{min} > 0$, the grid point (x_i, y_j) is called *regular*, otherwise it is *irregular* meaning the interface cuts through the standard five point finite difference stencil. Note that, with the level set function representation, the interface can only cut the grid line between two neighboring grid points, for example, (x_{i-1}, y_j) and (x_i, y_j) once.

The ADI method is based on that described in [41] and can be written as

$$
\begin{aligned}
\frac{u_{ij}^{k+\frac{1}{2}} - u_{ij}^{k}}{\Delta t/2} &= \delta_{xx}^{h}\beta\, u_{ij}^{k+\frac{1}{2}} - (C_x)_{ij}^{k+\frac{1}{2}} - Q_{ij}^{k} - R_{ij}^{k} + \delta_{yy}^{h}\beta\, u_{ij}^{k} - (C_y)_{ij}^{k} - f_{ij}^{k+\frac{1}{2}}, \\
\frac{u_{ij}^{k+1} - u_{ij}^{k+\frac{1}{2}}}{\Delta t/2} &= \delta_{xx}^{h}\beta\, u_{ij}^{k+\frac{1}{2}} - (C_x)_{ij}^{k+\frac{1}{2}} - Q_{ij}^{k} - R_{ij}^{k} + \delta_{yy}^{h}\beta\, u_{ij}^{k+1} - (C_y)_{ij}^{k+1} - f_{ij}^{k+\frac{1}{2}},
\end{aligned}
\tag{4.23}
$$

where Δt is the time step size, $\delta_{xx}^{h}\beta u_{ij}^{k+1/2} - (C_x)_{ij}^{k+1/2}$ is a finite difference approximation to βu_{xx} at (x_i, y_j) at the time level $t^{k+1/2}$ and so on which are different for different choices of the augmented variable.

The equivalent one-step form. If we eliminate $u_{ij}^{k+1/2}$ in terms of u_{ij}^{k} and u_{ij}^{k+1}, we get the one-step form below for purpose of discussion of the consistency and stability but not for the implementation,

$$
\begin{aligned}
\frac{u_{ij}^{k+1} - u_{ij}^{k}}{\Delta t} \;=\; & \frac{1}{2}\delta_{xx}^{h}\left((\beta u)_{ij}^{k} + (\beta u)_{ij}^{k+1}\right) - (C_x)_{ij}^{k+\frac{1}{2}} - Q_{ij}^{k} - R_{ij}^{k} \\
& + \frac{1}{2}\left(\delta_{yy}^{h}(\beta u)_{ij}^{k} - (C_y)_{ij}^{k} + \delta_{yy}^{h}(\beta u)_{ij}^{k+1} - (C_y)_{ij}^{k+1}\right) \\
& + \frac{\Delta t}{4}\delta_{xx}^{h}\left(\delta_{yy}^{h}(\beta u)_{ij}^{k} - (C_y)_{ij}^{k} - \delta_{yy}^{h}(\beta u)_{ij}^{k+1} + (C_y)_{ij}^{k+1}\right).
\end{aligned}
\tag{4.24}
$$

The derivation is almost the same except for the factor β as in [41]. In the expression above, the right hand side in the first line is a discretization for $(\beta u_x)_x$ while the second line is for that of $(\beta u_y)_y$, in which the sum is an approximation for $\nabla \cdot (\beta \nabla u)$ at the time level $t^{k+1/2}$. Thus it is possible to determine the correction terms at each time level t^k and t^{k+1} or at the middle level $t^{k+1/2}$. Also as discussed in [41], if the solution is continuous $[u] = 0$, then we can simply set the correction terms $Q_{ij}^{k} = 0$ and $R_{ij}^{k} = 0$ in our ADI algorithms. Those two upper letters Q and R may be used to represent other quantities in this paper.

At regular grid points where the interface does not cut the five-point finite difference stencil, the standard ADI method is used, in which

$$
(C_x)_{ij}^{k+\frac{1}{2}} = (C_y)_{ij}^{k} = Q_{ij}^{k} = (C_y)_{ij}^{k+1} = R_{ij}^{k} = 0.
$$

At each irregular grid point where the interface cuts through either between the grid lines x_{i-1} and x_{i+1} or y_{j-1} and y_{j+1}, we need to determine the finite difference discretization and these correction terms.

Take Figure 4.1 as an example, and suppose that the interface cuts through the grid line $y = y_j$ at $(x^*, y_j) \in \Gamma$. Using the formula from [41, 38] and omitting the superscripts k for simplicity, we have the following lemma.

Lemma 4.1. *Assume that* $u \in C^2(\Omega^{\pm})$, i.e., $u(x,y,t)$ *is twice differentiable. Let* $x_i \le x^* < x_{i+1}$, *and* $(x_i, y_j) \in \Omega^-$, $(x_{i+1}, y_j) \in \Omega^+$ *then,*

$$\begin{aligned} \frac{u(x_{i-1},y_j) - 2u(x_i,y_j) + u(x_{i+1},y_j)}{h^2} &= u_{xx}^-(x^*,y_j) + \frac{[u]}{h^2} \\ &\quad + [u_x]\frac{(x_{i+1}-x^*)}{h^2} + [u_{xx}]\frac{(x_{i+1}-x^*)^2}{2h^2} + O(h) \\ &= u_{xx}^-(x^*,y_j) - (C_x)_{ij} + O(h). \end{aligned} \tag{4.25}$$

So the correction term C_x *is given by*

$$(C_x)_{ij} = -\left(\frac{[u]}{h^2} + [u_x]\frac{(x_{i+1}-x^*)}{h^2} + [u_{xx}]\frac{(x_{i+1}-x^*)^2}{2h^2}\right). \tag{4.26}$$

There is a similar formula for the grid point $(x_{i+1,j}, y_j)$ in the x-direction as well,

$$\begin{aligned} \frac{u(x_i,y_j) - 2u(x_{i+1},y_j) + u(x_{i+2},y_j)}{h^2} &= u_{xx}^+(x^*,y_j) - \frac{[u]}{h^2} \\ &\quad - [u_x]\frac{(x_i-x^*)}{h^2} - [u_{xx}]\frac{(x_i-x^*)^2}{2h^2} + O(h) \\ &= u_{xx}^+(x^*,y_j) - (C_x)_{i+1,j} + O(h). \end{aligned} \tag{4.27}$$

So the correction term C_x is given by

$$(C_x)_{i+1,j} = \frac{[u]}{h^2} + [u_x]\frac{(x_i-x^*)}{h^2} + [u_{xx}]\frac{(x_i-x^*)^2}{2h^2}. \tag{4.28}$$

The correction terms in (4.23) are determined mostly from the above formulas (4.26) and (4.28).

Remark 4.2. *If* β *is continuous across the interface, then we know* $[u] = w$ *and* $[u_n] = v/\beta$. *We have explicit expressions for* $[u], [u_x], [u_y], [u_{xx}], [u_{yy}]$. *The ADI method scheme is the classical one with modified right hand sides. When* β *is discontinuous, we can not directly get the dimension by dimension jump relations. The new ideas in this paper is to get those dimensional jump relations through a carefully chosen augmented variable along the interface.*

4.2.1 The ADI method using $[\beta u]$ as the augmented variable

The ADI method with a known $[\beta u] = q(s)$ is to discretize the original PDE directly. In the spatial discretization, we treat βu as a whole. That is, we consider βu as the unknown variable. With known values $[\beta u_n] = v$ and $[\beta u] = q$, the augmented variable, we can find $[\beta u]$, $[\beta u_x]$, $[\beta u_{xx}]$, and $[\beta u_y]$, $[\beta u_{yy}]$ from (4.14) and (4.17). Thus we can get the finite difference approximations for $(\beta u)_{xx}$ and $(\beta u)_{yy}$ easily. For example, assume that $(x_{i-1}, y_j) \in \Omega^-$, $(x_i, y_j) \in \Omega^-$, but $(x_{i+1}, y_j) \in \Omega^+$, then the finite difference discretization for approximating βu_{xx} is

$$\begin{aligned} &\frac{\beta^- u_{i-1,j} - 2\beta^- u_{ij} + \beta^+ u_{i+1,j}}{h^2} - (\tilde{C_x})_{ij} \approx (\beta u)^-_{xx} \qquad \text{where} \\ &(\tilde{C_x})_{ij} = \left(\frac{[\beta u]}{h^2} + [\beta u_x] \, \frac{(x_{i+1} - x^*)}{h^2} + [\beta u_{xx}] \, \frac{(x_{i+1} - x^*)^2}{2h^2} \right). \end{aligned} \tag{4.29}$$

The ADI discretization (the coefficients of the unknown u_{ij}^k's) remains the same with respect to βu with different correction terms.

The advantage of this ADI method is that finite difference scheme is the same as the classical one except that we treat βu as a whole.

The stability discussion

If the augmented variable is exact, then the modified ADI method is actually a Crank-Nicholson scheme with a perturbation term $\Delta t \, u_{xxyyt}\beta/4$ from (4.24) if $[u] = 0$. Thus the method would be asymptotically unconditionally stable. Nevertheless, the computed augmented variable can affect the stability. The most conservative estimate using the growth factor assuming that the augmented variable is at least first order accurate is that

$$\frac{\Delta t \, h}{h^2} \le C \frac{\beta_{min}}{\beta_{max}}, \quad \text{or} \quad \Delta t \le C \frac{\beta_{min}}{\beta_{max}} h, \tag{4.30}$$

where $C \le 5$ is the width of the interpolation scheme.

4.2.2 The ADI method using $d[\beta u]/d\tau$ as the augmented variable

The ADI method with a known $\frac{d[\beta u]}{d\tau} = q(s)$ is a better one in our opinion because it takes into account the two scales of two regions involved. It also allows larger time step size compared with the previous one. The ADI discretization is applied to the original PDE with modified finite difference coefficients in each coordinate direction. In the spatial discretization, we treat βu as a whole and use the known jump conditions $[u] = w$ and $[\beta u_n] = v$, and the augmented variable $\frac{d[\beta u]}{d\tau} = q$, in deriving the dimension by dimension finite difference discretization at grid points where the interface cuts the 3-point stencil. We know the first and second order derivative jump relations, $[\beta u_x]$ and $[\beta u_{xx}]$, $[\beta u_y]$ and $[\beta u_{yy}]$ from (4.20), (4.21), and (4.17). We derive the analytic expressions of the finite difference discretization below.

For example, assume that $(x_{i-1}, y_j) \in \Omega^-$, $(x_i, y_j) \in \Omega^-$, but $(x_{i+1}, y_j) \in \Omega^+$, then the finite difference discretization for approximating βu_{xx} at (x_i, y_j) is

$$\begin{aligned}
&\gamma_1(\beta^- u_{i-1,j}) + \gamma_2(\beta^- u_{i,j}) + \gamma_3(\beta^+ u_{i+1,j}) \approx (\beta u_x)_x^- - C_x, \\
&\hat{\gamma}_1(\beta^- u_{ij}) + \hat{\gamma}_2(\beta^+ u_{i+1,j}) + \hat{\gamma}_3(\beta^+ u_{i+2,j}) \approx (\beta u_x)_x^+ - \hat{C}_x,
\end{aligned} \tag{4.31}$$

assuming that the interface cuts the grid line y_j at x^*, or $(x^*, y_j) \in \Gamma$. Note we have omitted some index dependencies for simplification of notation. We explain only the derivation of the first finite difference approximation here. The coefficients are determined so that the linear combination minus the correction term (C_x) can approximate $(\beta u_x)_x$ at (x^*, y_j) in the most accurate way possible if the true solution is substituted in the finite difference formulas above. Thus with the Taylor expansion at (x^*, y_j) for each $u(x_l, y_j)$, $l = i-1, i, i+1$, we have

$$\begin{aligned}
&\gamma_1\beta^- u(x_{i-1}, y_j) + \gamma_2\beta^- u(x_i, y_j) + \gamma_3\beta^+ u(x_{i+1}, y_j) - C_x \\
&\quad = \gamma_1\beta^- \left(u^- + u_x^-(x_{j-1} - x^*) + u_{xx}^-(x_{j-1} - x^*)^2/2\right) \\
&\quad + \gamma_2\beta^- \left(u^- + u_x^-(x_j - x^*) + u_{xx}^-(x_j - x^*)^2/2\right) \\
&\quad + \gamma_3\beta^+ \left(u^+ + u_x^+(x_{j+1} - x^*) + u_{xx}^+(x_{j+1} - x^*)^2/2\right) - C_x + O\left(\max_{1\le l\le 3} |\gamma_l| h^3\right)
\end{aligned}$$

$$\begin{aligned}
= \quad & u^- \left(\beta^-\gamma_1 + \beta^-\gamma_2 + \beta^+\gamma_3 \right) + \gamma_3\,\beta^+\,[u] \\
& + (\beta u)_x^- \left((x_{j-1} - x^*)\,\gamma_1 + (x_j - x^*)\,\gamma_2 + (x_{j+1} - x^*)\gamma_3 \right) \\
& + \gamma_3\,(x_{j+1} - x^*)\,[\,(\beta u)_x] + \frac{1}{2}(\beta u)_{xx}^- \left((x^* - x_{j-1})^2\,\gamma_1 + (x^* - x_j)^2\,\gamma_2 \right. \\
& \left. + (x_{j+1} - x^*)^2\,\gamma_3\right) + \gamma_3\;[\,(\beta u)_{xx}\,]\,(x_{j+1} - x^*)^2/2 - C_x + O\left(\max_{1\le l\le 3} |\gamma_l| h^3\right).
\end{aligned}$$

Thus we get the following system of equations.

$$\left\{\begin{array}{lcl}
\beta^-\gamma_1 + \beta^-\gamma_2 + \beta^+\gamma_3 & = & 0 \\
(x_{j-1} - x^*)\,\gamma_1 + (x_j - x^*)\,\gamma_2 + (x_{j+1} - x^*)\gamma_3 & = & 0 \\
\frac{1}{2}(x^* - x_{j-1})^2\,\gamma_1 + \frac{1}{2}(x^* - x_j)^2\,\gamma_2 + \frac{1}{2}(x_{j+1} - x^*)^2\,\gamma_3 & = & 1,
\end{array}\right. \tag{4.32}$$

and the correction term

$$C_x = \gamma_3 \left(\beta^+\,[u] + (x_{j+1} - x^*)\,[\,(\beta u)_x] + \frac{1}{2}(x_{j+1} - x^*)^2\,[\,(\beta u)_{xx}\,] \right). \tag{4.33}$$

The solution to the system above has a closed form below

$$\left\{\begin{array}{l} \gamma_1 = 2\,\dfrac{\beta^- h - [\beta](x_j - x^*)}{h\,D_j} \\[2ex] \gamma_2 = -2\,\dfrac{\beta^-(x_{j+1} - x^*) - \beta^+(x_{j-1} - x^*)}{h\,D_j} \\[2ex] \gamma_3 = \dfrac{2\beta^-}{D_j}, \end{array}\right. \tag{4.34}$$

where

$$D_j \quad = \quad 2\beta^- h^2 + [\beta](x_{j-1} - x^*)(x_j - x^*). \tag{4.35}$$

The following theorem shows that the solution to the system of equations (4.32) is unique.

Theorem 4.3. *If $\beta^-\beta^+ > 0$, then $D_j > 0$. Then, there is a unique solution to the system of equations (4.32).*

Proof: Without loss of generality, we assume $\beta^- > 0$ and $\beta^+ > 0$. Notice that $(x_{j-1} - x^*)(x_j - x^*) \geq 0$, so if $[\beta] \geq 0$, then $D_j \geq 2\beta^- h^2 > 0$, hence the theorem is true. If $[\beta] < 0$, i.e., $\beta^- > \beta^+$, then since $\beta^-\beta^+ > 0$ we have

$$\left| \frac{\beta^- - \beta^+}{\beta^-} \right| < 1.$$

Hence

$$\begin{array}{rcl} h^2 + \dfrac{[\beta]}{2\beta^-}(x_{j-1} - x^*)(x_j - x^*) & > & h^2 - \dfrac{1}{2}(x_{j-1} - x^*)(x_j - x^*) \\[2ex] & > & h^2 - \dfrac{1}{2}(2h)h = 0, \end{array}$$

where we have used the fact that the interface point x^* is between x_j and x_{j+1}, that is, $x_j \leq x^* < x_{j+1}$. This completes the proof. □

Similarly, the system of equations for $\{\hat{\gamma}_k\}$'s is

$$\left\{\begin{array}{lcl} \beta^-\hat{\gamma}_1 + \beta^+\hat{\gamma}_2 + \beta^+\hat{\gamma}_3 & = & 0 \\[1ex] (x_j - x^*)\,\hat{\gamma}_1 + (x_{j+1} - x^*)\,\hat{\gamma}_2 + (x_{j+2} - x^*)\hat{\gamma}_3 & = & 0 \\[1ex] \dfrac{1}{2}(x^* - x_j)^2\,\hat{\gamma}_1 + \dfrac{1}{2}(x_{j+1} - x^*)^2\,\hat{\gamma}_2 + \dfrac{1}{2}(x_{j+2} - x^*)^2\,\hat{\gamma}_3 & = & 1. \end{array}\right. \tag{4.36}$$

And we have a closed form for the coefficients,

$$\left\{\begin{aligned} \hat{\gamma}_1 &= \frac{2\beta^+}{D_{j+1}} \\ \hat{\gamma}_2 &= -2\,\frac{\beta^-(x_{j+2}-x^*)-\beta^+(x_j-x^*)}{h\,D_{j+1}} \\ \hat{\gamma}_3 &= 2\,\frac{\beta^+h-[\beta](x_{j+1}-x^*)}{h\,D_{j+1}}, \end{aligned}\right. \tag{4.37}$$

where

$$D_{j+1} \;=\; 2\beta^+h^2-[\beta](x_{j+2}-x^*)(x_{j+1}-x^*). \tag{4.38}$$

We observe some sort of symmetry compared with $\{\gamma_k\}$'s as we can expect. With almost the same process, it is easy to show that the system has a unique solution. The correction term is

$$\hat{C}_x = -\widehat{\gamma}_1\left(\beta^-\,[u]+(x_j-x^*)\,[\,(\beta u)_x]+\frac{1}{2}(x_j-x^*)^2\,[\,(\beta u)_{xx}\,]\right). \tag{4.39}$$

The stability discussion

The stability of this ADI method is better than the first one because we have taken into account the two scales in constructing the ADI scheme. If the augmented variable is exact, then the modified ADI method is actually a Crank-Nicholson scheme with a perturbation term $\frac{\beta}{4}\,u_{xxyyt}\,\Delta t$ from (4.24) if $[u]=0$. Thus the method would be asymptotically unconditionally stable. Nevertheless, the computed augmented variable can affect the stability. The most conservative estimate using the growth factor assuming that the augmented variable is at least first order accurate is that

$$\frac{\Delta t\,h}{h^2}\le C, \quad \text{or} \quad \Delta t\le Ch, \tag{4.40}$$

where $C\le 5$ is the width of the interpolation scheme. In numerical experiments, the scheme does behave like asymptotically unconditionally stable. For small to modest jump ratios, or large jump ratios with fine mesh, the ADI method works nicely.

4.3 Discretization of the augmented equation and the Schur complement

The augmented strategy has been discussed in detail in [44, 38, 46]. The implementation of course depends on different applications. Here we briefly explain some key details for the augmented algorithms.

4.3.1 Discretization of an augmented variable

The augmented variable is defined at the *computed orthogonal projections* $\mathbf{X}_l = (X_l, Y_l)$, $l = 1, 2, \cdots, N_b$, of irregular grid points from one particular side (say $\varphi_{ij} > 0$) to avoid possible clustering, see [38] for more details.

4.3.2 The ADI method in the matrix-vector form given an augmented variable

The finite difference solution $\{u_{ij}^k\}$ form a vector $\mathbf{U}^k$ whose dimension is $O(NM)$, where M and N are the number of grid lines in the x- and y-directions with $M \sim N$, respectively. The discrete augmented variable $Q_l^{k^*}$, $l = 1, 2, \cdots, N_b$ also forms a vector $\mathbf{Q}^{k^*} \in R^{N_b}$, where $k^* = k + 1/2$, $N_b \sim O(N)$ is the number orthogonal projections of irregular grid points from one particular side, say $\varphi_{ij} > 0$.

Given $\mathbf{Q}^{k^*}$, from the ADI method (4.23), we get $\{u_{ij}^{k+1}\}$. The solution using the ADI method with known augmented variable $\mathbf{Q}^{k^*}$ in the matrix-vector form can be written as

$$A\,\mathbf{U}^{k+1} + B\mathbf{Q}^{k^*} = \mathbf{F}_1^k \tag{4.41}$$

for some vector $\mathbf{F}_1^k$ and sparse matrices A and B which do not need to be generated.

4.3.3 The discrete augmented equation in the matrix-vector form

We also need to discretize the augmented equation such as (4.13), or (4.19), to get a complete system for $\mathbf{U}^{k+1}$ and $\mathbf{Q}^{k^*}$. This step involves local interpolations.

For the first augmented approaches, that is, select $[\beta u]$ as the augmented variable, the interpolation is similar to that used in [44, 46]. With the augmented equation (4.13) as an example, we need to interpolate u_{ij}^{k+1} to get $U(\mathbf{X}_l)^{k+1}$ using $[u]^{k+1} = w^{k+1}$ and $[\beta u^{k+1}] = q^{k^*}$, here $k^* = k + 1$. We use the least squares interpolation schemes for interface problems, see Section 6.1.3 in [44].

The interpolation scheme for approximating $u(\mathbf{X}_l)^-$ can be written as

$$u(\mathbf{X}_l)^- = \sum_{k=0}^{k_s-1} \gamma_k\, u_{i^*+k,\, j^*+k} - C, \tag{4.42}$$

where k_s is the number of grid points involved in the interpolation scheme, (x_{i^*}, y_{j^*}) is the closest grid point to $\mathbf{X}_l$, C is a correction term.

The coefficients $\{\gamma_k\}$ are determined by minimizing the interpolation error of (4.42) when $u_{i^*+k,\, j^*+k}$ is substituted by the exact solution $u(x_{i^*+k}, y_{j^*+k})$. Using the local coordinate system

(4.10) centered at the point $\mathbf{X}_l$ or any point on the interface nearby, and denoting the local coordinates of (x_{i^*+k}, y_{j^*+k}) as (ξ_k, η_k), we have the following from the Taylor expansion at $\mathbf{X}_l = (X_l, Y_l)$, or $(0,0)$ in the local coordinates:

$$\begin{aligned} u(x_{i^*+k}, y_{j^*+k}) &= u(\xi_k, \eta_k) = u^{\pm} + \xi_k u_{\xi}^{\pm} + \eta_k u_{\eta}^{\pm} \\ &\quad + \frac{1}{2}\xi_k^2 u_{\xi\xi}^{\pm} + \xi_k \eta_k u_{\xi\eta}^{\pm} + \frac{1}{2}\eta_k^2 u_{\eta\eta}^{\pm} + O(h^3), \end{aligned} \tag{4.43}$$

where the '+' or '−' sign is chosen depending on whether (ξ_k, η_k) lies on the '+' or '−' side of Γ, $u^{\pm}$, $u_{\xi}^{\pm}$, $\cdots$, $u_{\eta\eta}^{\pm}$ are evaluated at the local coordinates $(0,0)$, or $\mathbf{X}_l = (X_l, Y_l)$ in the original coordinate system.

The coefficients $\{\gamma_k\}$'s satisfy the following linear system of equations.

$$\begin{aligned} &\sum_{k=0}^{k_s-1} \gamma_k = 1, &&\sum_{k=0}^{k_s-1} \gamma_k \, \xi_k = 0, \\ &\sum_{k=0}^{k_s-1} \gamma_k \, \eta_k = 0, &&\frac{1}{2}\sum_{k=0}^{k_s-1} \gamma_k \, \xi_k^2 = 0, \\ &\sum_{k=0}^{k_s-1} \gamma_k \, \xi_k \eta_k = 0, &&\frac{1}{2}\sum_{k=0}^{k_s-1} \gamma_k \, \eta_k^2 = 0. \end{aligned} \tag{4.44}$$

Once $\{\gamma_k\}$'s are computed, the correction term C is determined from the following,

$$C = a_2 \, [u] + a_4 \, q + a_6 \, [u_\eta] + a_8 \, [u_{\xi\xi}] + a_{10} \, [u_{\xi\eta}] + a_{12} \, [u_{\eta\eta}] \, , \tag{4.45}$$

where $a_2 = \sum_{k \in K^+} \gamma_k$, $a_4 = \sum_{k \in K^+} \xi_k \gamma_k$, $a_6 = \sum_{k \in K^+} \eta_k \gamma_k$, and so on, $k \in K^+$ means that the grid point involved in the interpolation $(x_{i^*+k}, y_{j^*+k}) \in \Omega^+$.

The discretization of the augmented equation (4.13) above can be written as

$$C \, \mathbf{U}^{k+1} + D\mathbf{Q}^{k^*} = \mathbf{F}_2^k \tag{4.46}$$

for some sparse matrices C and D. The solution $(\mathbf{U}^{k+1}, \mathbf{Q}^{k^*})$ satisfies the following system of equations,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{U}^{k+1} \\ \mathbf{Q}^{k^*} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1^k \\ \mathbf{F}_2^k \end{bmatrix}. \tag{4.47}$$

The Schur complement for $\mathbf{Q}^{k^*}$ is

$$(D - CA^{-1}B)\mathbf{Q}^{k^*} = \mathbf{F}_2^k - CA^{-1}\mathbf{F}_1^k = \bar{\mathbf{F}}^k, \qquad \text{or} \qquad S\mathbf{Q}^{k^*} = \bar{\mathbf{F}}^k, \tag{4.48}$$

where $A^{-1}\mathbf{F}^k$ is the result of one step of the ADI for a regular problem. Define the residual of the augmented equation given $\mathbf{Q}$ as

$$R(\mathbf{Q}) = C\,\mathbf{U}^{k+1} + D\mathbf{Q} - \mathbf{F}_2^k. \tag{4.49}$$

Note that $R(\mathbf{0}) = C\mathbf{U}^{k+1}(\mathbf{0}) - \mathbf{F}_2^k$, and from (4.41), we have $A^{-1}B\mathbf{Q} = A^{-1}\mathbf{F}_1^k - \mathbf{U}^{k+1}(\mathbf{Q})$, thus the matrix-vector multiplication $S\mathbf{Q}$ is

$$\begin{aligned} S\mathbf{Q} &= \left(D - CA^{-1}B\right)\mathbf{Q} = D\mathbf{Q} - CA^{-1}B\mathbf{Q} \\ &= D\mathbf{Q} + C\mathbf{U}^{k+1}(\mathbf{Q}) - \mathbf{F}_2^k + \mathbf{F}_2^k - CA^{-1}\mathbf{F}_1^k \\ &= R(\mathbf{Q}) - (C\mathbf{U}^{k+1}(\mathbf{0}) + D\mathbf{0} - \mathbf{F}_2^k) \\ &= R(\mathbf{Q}) - R(\mathbf{0}) \end{aligned} \tag{4.50}$$

since $\mathbf{U}^{k+1}(\mathbf{0}) = A^{-1}\mathbf{F}_1^k$. The right hand side of $S\mathbf{Q} = \bar{\mathbf{F}}^k$ can be computed from $-R(\mathbf{0})$ which corresponds to the residual of the augmented equation with zero value of the components of the augmented variable, which is obtained from one step ADI method.

For fixed interface Γ and time step Δt, the Schur complement matrix $S \in R^{N_b \times N_b}$ is fixed. So we form the matrix at the first step by using $\mathbf{Q} = \mathbf{e}_i$, $i = 1, 2, \cdots, N_b$. For moving interface problem, or adaptive time steps, it is recommended to use the GMRES iterative method, see for example [60].

4.3.4 The interpolation scheme when $d[\beta u]/d\tau$ is the augmented variable

Similar to the derivation of the ADI coefficients, the key to discretize the augmented equation $[\frac{du_q}{d\tau}]_\Gamma = q(s)$ is to treat $[\beta u]$ as one scale while enforcing the $[u] = w$ condition. The interpolation scheme for approximating $u(\mathbf{X}_l)^-$, for example, now is

$$u(\mathbf{X}_l)^- = \sum_{k=0}^{k_s-1} \gamma_k\,(\beta\,u)_{i^*+k,\,j^*+k} - C. \tag{4.51}$$

In other words, we interpolate $(\beta u)_{ij}$ instead of u_{ij} and the first equation in (4.44) now becomes

$$\sum_{k=0}^{k_s-1} \gamma_k \beta_{i^*+k,\,j^*+k} = 1, \tag{4.52}$$

while the rest of equations in (4.44) remain the same; and the correction term now is

$$C = \hat{a}_2\,[u] + a_4\,v + a_6\,q + a_8\,[\beta u_{\xi\xi}] + a_{10}\,\,[\beta u_{\xi\eta}] + a_{12}\,\,[\beta u_{\eta\eta}]\,, \tag{4.53}$$

where $\hat{a}_2 = \displaystyle\sum_{k=0}^{k_s-1} \gamma_k \, (\beta)_{i^*+k,\, j^*+k}$, and $v = [\beta u_n]$ is the original flux jump condition. The remaining a_i's have the same definition as before.

4.3.5 Outline of the new ADI methods

We outline the new ADI methods below.

Setup: Setup a grid $\{(x_i, y_j)\}$; the initial condition $\{u^0_{ij}\}$, the level set function $\{\varphi_{ij}\}$; classify grid points $\{(x_i, y_j)\}$ as regular or irregular using (4.22).

Compute the Schur complement matrix: At every irregular grid point (x_i, y_j) with $\varphi_{ij} > 0$, find the orthogonal projection $\mathbf{X}_l$ at which the augmented variable is defined. The process also generates an ordering of $\{\mathbf{X}_l\}$, $l = 1, 2, \cdots, N_b$. We also find and store the normal (tangential can be determined from the normal) and the curvature information.

For $i = 1, 2, \cdots, N_b$, set $\mathbf{Q}^{k^*} = \mathbf{e}_i$, the i-th unit vector, and apply the ADI method. The computed residual of the augmented equation is the i-th column of the Schur complement S. Carry out the $S = LU$ decomposition.

Apply the ADI method from t^k to t^{k+1}: Find the right hand side of the Schur complement $\bar{\mathbf{F}}^k$ by setting $\mathbf{Q}^{k^*} = \mathbf{0}$. Apply the stored LU decomposition to solve $S\mathbf{Q}^{k^*} = \bar{\mathbf{F}}^k$ to get the augmented variable $\mathbf{Q}^{k^*}$. Apply the modified ADI method with $\mathbf{Q}^{k^*}$ to get $\{u^{k+1}_{ij}\}$, and overwrite it to $\{u^k_{ij}\}$.

4.3.6 Convergence discussions

The derived ADI methods are obviously consistent. If the solution is a piecewise quadratic function, then the local truncation errors at all grid points are zero assuming the interface quantities such as the projection, normal and tangential directions, and curvature are exact. This is an indication of second order convergence if the ADI methods are also stable.

When we set $[\beta u]$ as the augmented variable, we get clean time step constraint $\Delta t \le Ch/\beta_{max}$. The constant C depends on the interpolation scheme and it is roughly between $5 \sim 10$.

We get better than expected stability when we set $(\frac{d[\beta u]}{d\tau})^{k+1/2}$ as the augmented variable. The ADI method behaves like asymptotically unconditionally stable. One possible cause may be the right scale in the algorithm which absorbs β into the formulation as a whole.

4.4 Numerical examples

We first validate our method through problems with known analytic solutions. Most of computations are done on a MAC Laptop with 2.6 Hz Intel Core i7 processor. Most of the results shown here are using $\frac{d[\beta u]}{d\tau}$ as the augmented variable unless specified differently since we think it is the better choice. We choose the time step size as $\Delta t = h/2$ or $\Delta t = h/(2C)$ with $C = \max\{\beta^+, \beta^-\}$.

4.4.1 An example from the literature

The first example is taken from [72]. This is a good testing example since the solution, its flux, and the source term, have two different scales and the problem is time invariant. The analytic solution is given by

$$u(x,y,t) = \cos(t) \begin{cases} \dfrac{(x^2+y^2)^3-1}{\beta^-} - \dfrac{3}{\beta^+} & \text{if } r \le 1, \\ -\dfrac{3}{\beta^+(x^2+y^2)} & \text{if } r > 1, \end{cases} \tag{4.54}$$

where $r = \sqrt{x^2+y^2}$. The source term is

$$f(x,y,t) = \begin{cases} \left(\dfrac{(x^2+y^2)^3-1}{\beta^-} - \dfrac{3}{\beta^+} \right) \sin t + 36(x^2+y^2)^2 \cos t & \text{if } r \le 1, \\ -\dfrac{3}{\beta^+(x^2+y^2)} \sin t - \dfrac{12}{(x^2+y^2)^2} \cos t & \text{if } r > 1. \end{cases} \tag{4.55}$$

In Table 4.1-4.3, we show numerical experiment results with the domain $[-2,\ 2] \times [-2,\ 2]$. Table 4.1 shows the results of the grid refinement analysis with modest jump ratios $\beta^- = 10$ and $\beta^+ = 2$ in (a), and $\beta^- = 2$ and $\beta^+ = 10$ in (b). Table 4.2 shows the results for large jump ratios $\beta^- = 1000$ and $\beta^+ = 1$ in (a), and $\beta^- = 1$ and $\beta^+ = 1000$ in (b). Table 4.3 shows the CPU time and the condition number of the Schur complement system with $\beta^- = 1000$ and $\beta^+ = 1$.

The first column in Table 4.1-4.3 is the mesh size in both x and y directions. The second column is the error of the computed solution in the maximum norm (pointwise). The third column is the approximate convergence order r obtained by (2.65). The fourth column is the error of the computed augmented variable $\frac{d[\beta u]}{d\tau}$ at the computed orthogonal projections from $\varphi_{ij} > 0$ side in the maximum norm (pointwise). We can obtain the gradient or directional derivative from each side of interface along any directions of the solution $u(x,y,t)$ using the augmented variable and the given flux jump condition $[\beta u_n] = v$. The fifth column is the approximate convergence order of the augmented variable. We observe second order convergence for all cases both for the solution and the augmented variable in the strongest norm. The computed accuracy (or the error) seems to be independent of the jump ratios. This is somewhat expected since the coefficient has been absorbed in the ADI method.

Figure 4.2 (a) shows a computed solution with $N = 64$, $\beta^- = 2$, $\beta^+ = 10$, and Figure 4.2 (b)

Table 4.1: A grid refinement analysis of the new ADI method at the final time $T = 1$ with modest jumps. (a) $\beta^- = 10$, $\beta^+ = 2$, (b) $\beta^- = 2$, $\beta^+ = 10$.

(a): $\beta^- = 10$; $\beta^+ = 2$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	3.6656e-02		2.4534e-03	
80	7.3944e-03	2.3095	6.5700e-03	1.9008
160	1.7855e-03	2.0501	1.6728e-03	1.9736
320	3.9268e-04	2.1849	4.5025e-04	1.8935

(b): $\beta^- = 2$; $\beta^+ = 10$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	1.1103e-02		7.1480e-03	
80	2.8390e-03	1.9675	1.6853e-03	2.0845
160	7.6865e-04	1.8850	4.8657e-04	1.7923
320	1.9389e-04	1.9871	1.1757e-04	2.0491

Table 4.2: A grid refinement analysis of the new ADI method at $T = 1$ with large jump ratios. (a) $\beta^- = 1000$, $\beta^+ = 1$, (b) $\beta^- = 1$, $\beta^+ = 1000$.

(a): $\beta^- = 1000$; $\beta^+ = 1$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	4.4455e-02		7.5757e-02	
80	1.0921e-02	2.0252	1.7310e-02	2.1298
160	2.5155e-03	2.1182	5.4003e-03	1.6805
320	5.9963e-04	2.0687	1.2742e-03	2.0835
640	1.0906e-04	2.4590	3.4965e-04	1.8656

(b): $\beta^- = 1$; $\beta^+ = 1000$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	2.1571e-02		9.4754e-05	
80	5.9738e-03	1.8524	2.2588e-05	2.0686
160	1.5718e-03	1.9262	6.3268e-06	1.8360
320	4.0456e-04	1.9580	1.5124e-06	2.0647
640	1.0248e-04	1.9811	3.9304e-07	1.9441

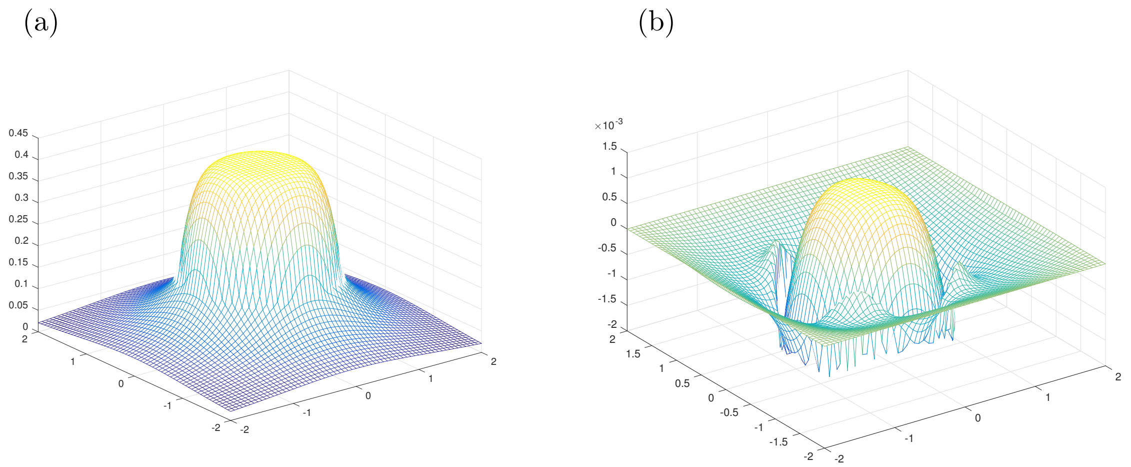

Figure 4.2: Plot of the computed solution for Example 4.4.1. (a), and the error (b) with $N = 64$, $\beta^- = 2$, $\beta^+ = 10$.

shows the corresponding error plot of the computed solution, which shows that the error is piecewise smooth and no large sparks around interface. Similar behaviors are observed with other computations as well.

In Table 4.3, we list the condition number of the Schur complement for one of the settings with $\beta^- = 1000$; $\beta^+ = 1$. The condition number is between $5 \sim 6$ and decreases a little as the mesh gets finer. Once again, this is true for all the test cases with different jump ratios.

The condition number of the Schur complement system when we use either $[u_n]$ or $[\beta u]$ is slightly larger when the jump ratio gets larger. In all of our testing examples with the jump ratio as large as $1 : 1000$ or $1000 : 1$, the condition number is under number 30.

Our computer code is not optimized, but it is still fast in terms of the CPU time from Table 4.3. The CPU time is roughly the same for different jump ratios.

Table 4.3: Condition number of the Schur complement matrix and the CPU time for the case of $\beta^- = 1000$; $\beta^+ = 1$.

N	40	80	160	320	640
$cond(S)$	6.5103	6.2896	6.1546	5.9063	5.8548
$CPU(s)$	0.45588	1.9964	10.100	61.148	438.26

To check the stability, we tried large Δt, for example, $\Delta t = 1$, $T = 1000$, $\beta^- : \beta^+ = 1000 : 1$; and $\Delta t = 100$, $T = 10^6$, $\beta^- : \beta^+ = 1000 : 1$. The code run to the final time with the energy $\max\{|u_{ij}^k|\}$ remains bounded in both cases. It looks like the developed ADI method is unconditionally stable.

Note that there are few ADI methods for interface problems in the literature to compare with except for the one in [72]. While our methods and the one in [72] are all second order accurate in space, our methods are also second order in time and allow large jump ratios, larger time step size. Note that $\Delta t = O(h^2)$ is used in [72] and the linear systems from the ADI method in [72] in each directional sweep are penta-diagonal instead of usual tri-diagonal which is still true in our ADI methods.

A grid refinement analysis with $[\beta u]$ as the augmented variable

Table 4.4: A grid refinement analysis of the new ADI method using the $[\beta u]$ as the augmented variable at $T = 1$ with large jump ratios. The second to fourth columns are the results for $\beta^- = 1000$, $\beta^+ = 1$. The average convergence rate is 2.257; The fourth and seventh columns are the condition numbers of the Schur complement matrix. The fifth to seventh columns are the results for $\beta^- = 1$, $\beta^+ = 1000$. The average convergence rate is 2.0778.

N	$\|E_N\|_\infty(1000:1)$	r	$cond(S)$	$\|E_N\|_\infty(1:1000)$	r	$cond(S)$
40	9.8746e-03		9.6930	1.7179e-02		10.056
80	1.9708e-03	2.3249	10.314	5.6197e-03	1.6121	7.389
160	4.4039e-04	2.1620	13.944	8.2220e-04	2.7729	10.347
320	9.0273e-05	2.2864	18.883	2.2835e-04	1.8482	10.880

In Table 4.4, we show the grid refinement results using $q = [\beta u]$ as the augmented variable for the same example with large jump ratios $\beta^- = 1000$, $\beta^+ = 1$ and $\beta^- = 1$, $\beta^+ = 1000$. Once again we see clearly second order convergence in the strongest norm. The average convergence rates in the infinity norm are 2.257 and 2.0778, respectively. The condition number of the Schur complement matrix is between 7 and 19. The advantage of this ADI is that it is relatively simpler and the finite difference coefficients remain the same as the classical one. The disadvantage compared with the ADI method with $[\beta u]_\tau$ is that the time step constraint depends on the jump ratio, $\Delta t \le Ch$, where $C \sim 5/\beta_{max}$. The constant C is between $5 \sim 10$ is from the width of the interpolation for the interface conditions.

Find the steady state solution

We can use the developed method to find the steady state solution of the interface problem numerically if it exists, or elliptic interface problems $\nabla \cdot \beta(\nabla u) - f$ using the PDE $u_t = \nabla \cdot (\beta \nabla u) = f$. In Table 4.5, we show the grid refinement analysis for the steady state solution of (4.54)-(4.55) with $\cos t$ replaced by number one and $\sin t$ by zero. In this example $\beta^- = 1$,

$\beta^+ = 1000$. The initial condition is chosen randomly. It seems that it is enough to have final time $T = 2$ to reach the steady state solution. Once again, we observe second order accuracy in the maximum norm for both of the solution and the augmented variable.

Table 4.5: The grid refinement analysis for the steady state solution with $T = 2$ with $\beta^- = 1$, $\beta^+ = 1000$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	1.7276 e-02		1.7908e-04	
80	4.7260 e-03	1.8701	4.2691e-05	2.0686
160	1.2523 e-03	1.9161	1.1732e-05	1.8636
320	3.3064 e-04	1.9212	2.8043e-06	2.0647
640	9.3791 e-05	1.8177	7.2879e-07	1.9441

4.4.2 An example with a variable dynamic flux jump condition

In the first example, $[\beta u]$ is a constant and the augmented variable $[\beta \nabla u \cdot \boldsymbol{\tau}] = 0$. So the example may not be general enough. In the second example, the true solution is

$$u(x,y,t) = g(t) \left\{ \begin{array}{ll} xy\left(x^2+y^2\right)^2 & \text{if } r \le 1, \\ xy\left(x^2+y^2\right) & \text{if } r > 1, \end{array} \right. \tag{4.56}$$

where $r = \sqrt{x^2+y^2}$. The source term $f(x,y,t)$, the initial condition at $t = 0$, and the Dirichlet boundary condition are determined from the true solution. This is a *general* example with a continuous solution, that is, $[u] = 0$, but non-homogeneous non-constant flux jump condition ($[\beta u_n] \neq 0$), and both $[\beta u]$ and $[\beta \nabla u \cdot \boldsymbol{\tau}]$ are varying along the interface,

$$\begin{array}{c} [\,\beta u_n\,] = g(t)\, xy\left(4\beta^+ - 6\beta^-\right), \\ \Big[\,\beta u\,\Big] = g(t)\, xy\left(\beta^+ - \beta^-\right), \qquad \Big[\,\beta \nabla u \cdot \boldsymbol{\tau}\,\Big] = g(t)\,\left(x^2 - y^2\right)\left(\beta^+ - \beta^-\right). \end{array} \tag{4.57}$$

Note that $[\beta u]$ and $[\beta \nabla u \cdot \boldsymbol{\tau}]$ are not free variables. In our numerical tests, we set $g(t) = \cos t$.

We first show the results of the grid refinement analysis with modest large jump ratios in Table 4.6 and large jump ratios in Table 4.7. The results agree with that in Example 1 with second order convergence for both of the solution (at all grid points) and the augmented variable (at some points on the interface).

Figure 4.3 shows two error plots with $N = 64$, $\beta^- = 10$, $\beta^+ = 2$ in the left plot, and $\beta^- = 0.033$, $\beta^+ = 33$ whose jump ratio is 1:1000 in the right plot. Once again, the errors are

Table 4.6: The grid refinement analysis for Example 2 with $T = 1$. (a): $\beta^- = 2$, $\beta^+ = 10$. The average convergence order for the solution and the augmented variable are 1.9898 and 2.1013 respectively. (b): $\beta^- = 10$, $\beta^+ = 2$. The average convergence order for the solution and the augmented variable are 2.4904 and 1.7774, respectively.

(a): $\beta^- = 10$; $\beta^+ = 2$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	7.0258e-05		1.4820e-02	
80	2.2003e-05	1.6749	1.5871e-03	3.2231
160	5.1093e-06	2.1065	8.8556e-04	0.8418
320	1.1214e-06	2.1878	1.8758e-04	2.2391

(b): $\beta^- = 2$; $\beta^+ = 10$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	3.6527e-04		1.0443e-02	
80	3.9449e-05	3.2109	2.8714e-03	1.8627
160	1.9328e-05	1.0293	1.0501e-03	1.4512
320	2.0585e-06	3.2310	2.5922e-04	2.0183

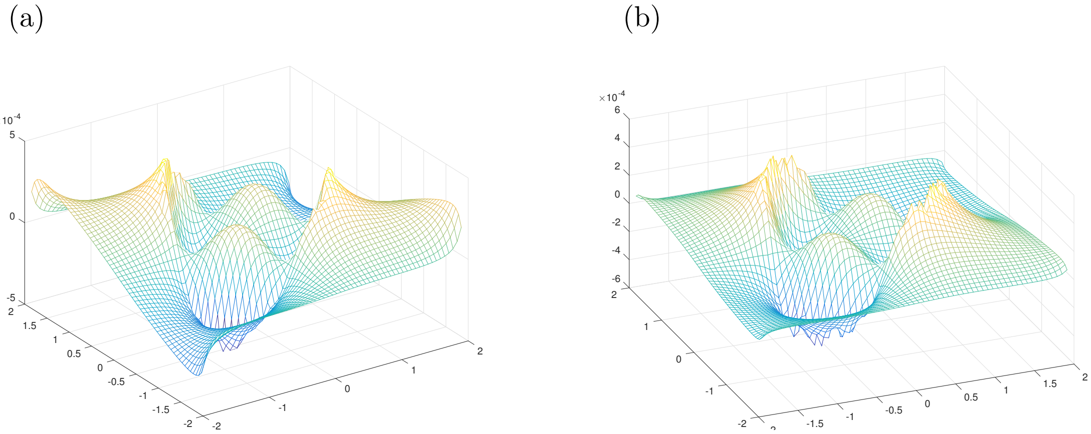

Figure 4.3: Plot of the errors with $N = 64$ and different jump ratios for Example 4.4.2. (a): $\beta^- = 10$, $\beta^+ = 2$; (b): $\beta^- = 0.033$, $\beta^+ = 33$ whose jump ratio is 1:1000.

piecewise smooth and no large sparks around the interface.

Next, we show the results with large jump ratios, 1 : 1000 and 1000 : 1. We re-scale the problem to get more balanced jump ratio without changing the nature of the problem. This can be done by multiplying the PDE by $\sqrt{\beta_{min}/\beta_{max}}$ or $\sqrt{\beta_{max}/\beta_{min}}$. In this way, we can get more balanced time step size. In Table 4.7, we show the grid refinement analysis results using the modified ADI method. The average convergence order once again is quadratic or better.

Table 4.7: The grid refinement analysis for Example 2 with $T = 1$ with large jump ratios 1 : 1000 and 1000 : 1. (a): $\beta^- = 0.033$, $\beta^+ = 33$. The average convergence order for the solution and the augmented variable are 2.2817 and 2.5749 respectively. (b): $\beta^- = 33$, $\beta^+ = 0.033$ so the jump ratio is 1000 : 1. The average convergence order for the solution and the augmented variable are 1.9381 and 1.9675, respectively.

(a): $\beta^- = 0.033$, $\beta^+ = 33$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	1.7240e-04		4.8628e-01	
80	2.7064e-05	2.6713	2.0254e-02	4.5855
160	7.1346e-06	1.9235	1.1348e-02	0.83576
320	1.4994e-06	2.2504	2.2989e-03	2.3034

(b): $\beta^- = 33$, $\beta^+ = 0.033$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
40	6.8045e-04		3.0187e-02	
80	1.8391e-04	1.8875	8.9924e-03	1.7471
160	6.4475e-05	1.5122	2.7157e-03	1.7274
320	1.2092e-05	2.4146	5.0469e-04	2.4279

A grid refinement analysis with very large jump ratios

We also tested very large jump ratios $\beta^-/\beta^+ = 1 : 10^5$ and $\beta^-/\beta^+ = 10^5 : 1$ as suggested by one of referees. Both ADI method work well if $\Delta t \leq Ch/\beta_{max}$, as shown in Table 4.8 using the second ADI method. The second ADI method does allow large time step size but the convergence behaves more erratically indicating the condition for asymptotic convergence may not true any more. Also, we show simulations of such large jump ratios for a flow problem in Figure 4.5.

Table 4.8: The grid refinement analysis for Example 4.4.2 with $T = 0.2$ with very large jump ratios $1 : 10^5$ in (a) and $10^5 : 1$ in (b). The average convergence order for the solution and the augmented variable are 2.0707 and 2.1739 respectively for (a) while they are 2.0005 and 1.7608, respectively for (b).

(a): $\beta^- = 1$, $\beta^+ = 10^5$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
32	1.1659e-04		7.1125e-02	
64	3.1996e-05	1.8655	1.2617e-02	2.4950
128	5.2947e-06	2.5953	3.0995e-03	2.0253
256	1.5727e-06	1.7513	7.7418e-04	2.0013

(b): $\beta^- = 10^5$, $\beta^+ = 1$.

N	$\|E_N\|_\infty$	r	$\|E_{aug}\|_\infty$	r
32	2.4321e-01		7.5076e-02	
64	3.9795e-02	2.6115	2.7540e-02	1.4468
128	8.1210e-03	2.2929	8.7106e-03	1.6607
256	3.7963e-03	1.0971	1.9290e-03	2.1750

4.4.3 An example in which the solution has a jump discontinuity

The developed ADI method in this paper can be applied to the case in which the solution has a finite jump across the interface without correction terms such as Q_{ij}^k and R_{ij}^k in (4.23). We would expect that the computed solution is super-linear convergent (nearly second order). We consider a general problem in which the analytic solution is

$$u(x, y, t) = \begin{cases} \cos(t) + e^{x^2+y^2}, & \text{if } r \leq 0.5, \\ \cos(t) + \sin(2x)\cos(2y), & \text{if } r > 0.5, \end{cases} \tag{4.58}$$

where $r = \sqrt{(x^2 + y^2)} = \frac{1}{2}$ is the interface in the domain $[-1, 1] \times [-1, 1]$. The source term, the flux jump condition, the initial and boundary conditions are determined from the analytic solution. The solution is not continuous across the interface,

$$[u] = w = \sin(2x)\cos(2y) - e^{1/4}.$$

In Table 4.9, we show grid refinement results for two cases with large jump ratios. The second and third columns are the results with $\beta^-/\beta^+ = 1 : 0.001$; while the fourth and fifth columns are the results with $\beta^-/\beta^+ = 0.001 : 1$. The average convergence orders are 1.8181 and 1.8877, respectively.

Table 4.9: A grid refinement analysis for Example 4.4.3 in which the solution has a finite jump discontinuity.

N	$\|E_n\|_\infty$	r	$\|E_n\|_\infty$	r
40	9.3882×10^{-3}		7.2805×10^{-4}	
80	3.1509×10^{-3}	1.5751	1.8623×10^{-4}	1.9670
160	7.6708×10^{-4}	2.0383	4.7239×10^{-5}	1.9790
320	2.1412×10^{-4}	1.8410	1.4268×10^{-5}	1.7171

4.4.4 An application example

We present an example to mimic a flow passing objects to show that our second ADI method can handle complicated geometries and multi-connected domains. In this example, we assume an incoming Poiseuille flow profile with $u = g(t)(4-y^2)$ which is used as the boundary condition in the domain $[-2,\, 2] \times\ [-2,\, 2]$. The permeability of the material (domain) is $1/\beta$ which is piecewise constant.

In Figure 4.4, we show the interface and contour plots of the computed solution with $g(t) = \cos t$ and a large jump ratio either $\beta^- : \beta^+ = 1/1000$ or $\beta^- : \beta^+ = 1000/1$. In the top plots, the permeability inside particles is large so flow get saturated while in the bottom plots the flow is hardly getting in due to small permeability. The interface is general with both convex and convex-concave. The results agree with physical reasoning and intuition.

We also tested a situation where the jump ratio is very large, $\beta^-/\beta^+ = 10^5 : 1$ and $\beta^-/\beta^+ = 1 : 10^5$. In this simulation, the Neumann BC, $\frac{\partial u}{\partial n} = 0$ is used for the right boundary. The convergence remains the same if the time step is $\Delta t \le Ch/\beta_{max}$, see Table 4.8. But for larger Δt, while the second ADI method still works, the convergence rate behaves more erratically indicating that the error constant is large.

4.4.5 A moving example

Finally, we show a moving interface example in the same setting as in the flow problem. Once again, the Neumann BC, $\frac{\partial u}{\partial n} = 0$ is used for the right boundary. The motion of interface is governed by the mean curvature with the normal velocity of the interface as $V_n = \alpha(\kappa - \kappa_0)$. The level set is evolved by the Hamilton-Jacobian equation $\varphi_t + V_n|\nabla\varphi| = 0$. We start with a six-star shaped interface $r = 1.2 + 0.4\sin(6\theta)$ in polar coordinates. We set $\kappa_0 = 1.2$. Thus the interface will relax to a circle. The coefficient is $\beta^+ = 50$ and $\beta^- = 1$. In Figure 4.6, we show the contour plots of the solution and the interface at different time. The result is as expected. In this case, the Schur complement matrix is not a constant matrix anymore and

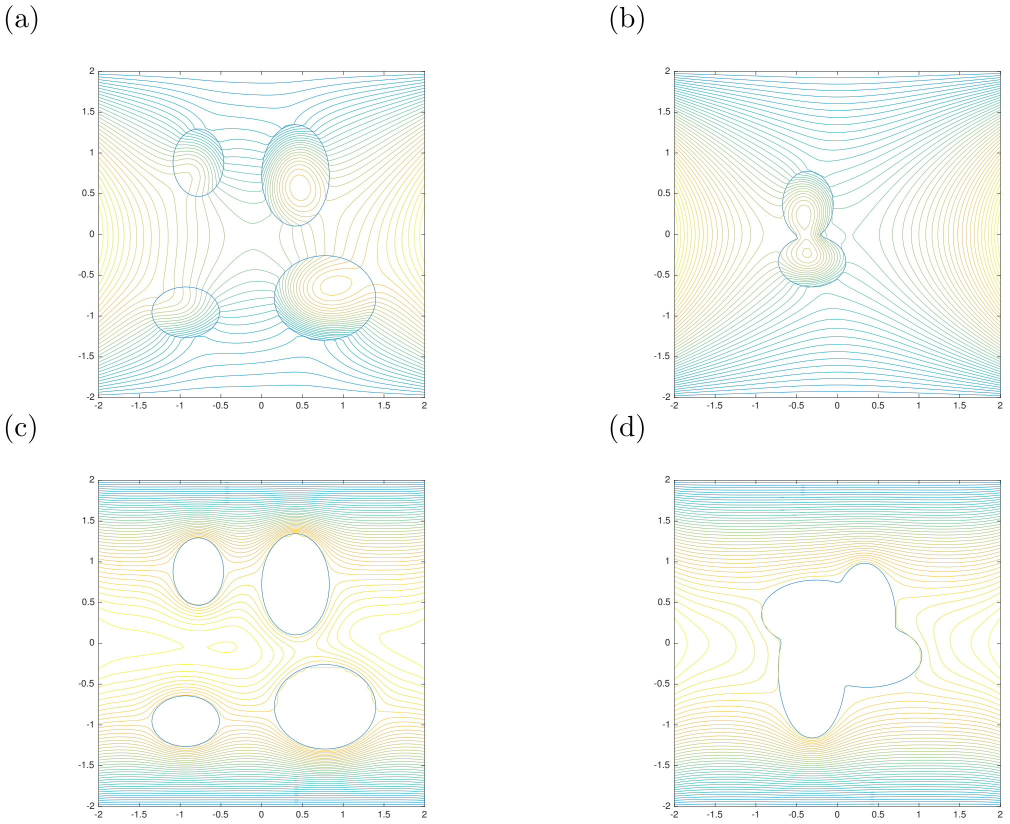

Figure 4.4: Snap contour plots ($T = 2\pi$) of the computed solution with $N = 80$, different jump ratios 1 : 1000 or 1000 : 1, and different geometries. (a) $\beta^- = 0.033$, $\beta^+ = 33$, (b) $\beta^- = 0.033$, $\beta^+ = 33$, (c) $\beta^- = 33$, $\beta^+ = 0.033$, and (d) $\beta^- = 100$, $\beta^+ = 0.1$. In the top plots, the permeability inside particles are large so flow get saturated while in the bottom plots the flow is hardly getting in due to small permeability.

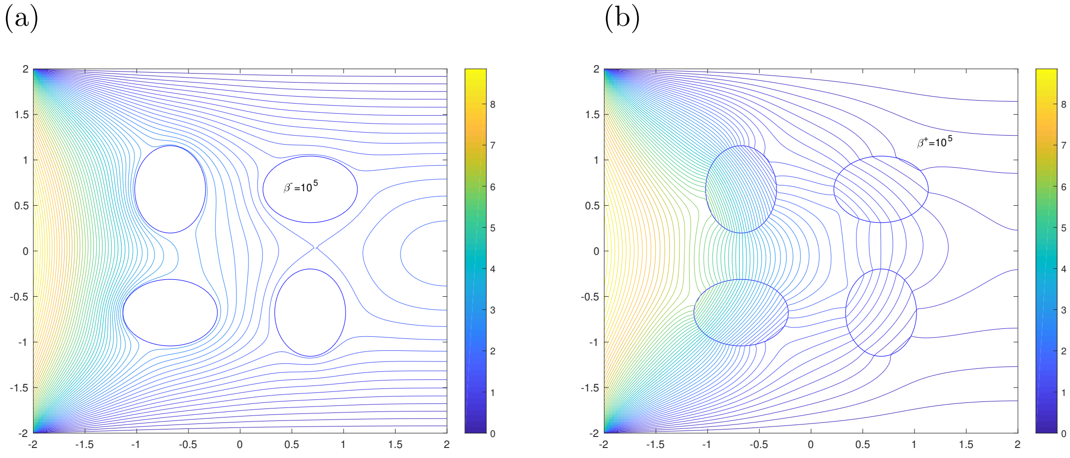

Figure 4.5: Contourplot of the solution in the flow situation with very large jump ratio $\beta^- : \beta^+ = 1/10^5$ or $\beta^- : \beta^+ = 10^5/1$.

its size also changes. We use the GMRES iterative method to solve the Schur complement system. Corresponding to the four different snap shots of the interface, the size of the Schur complement matrix is 462×462, 314×314, 264×264, 254×254, respectively, with condition number being $3.3200, 2.7468, 3.3265, 2.9068$. With the tolerance 10^{-6}, the average number of GMRES iterations is around 12.

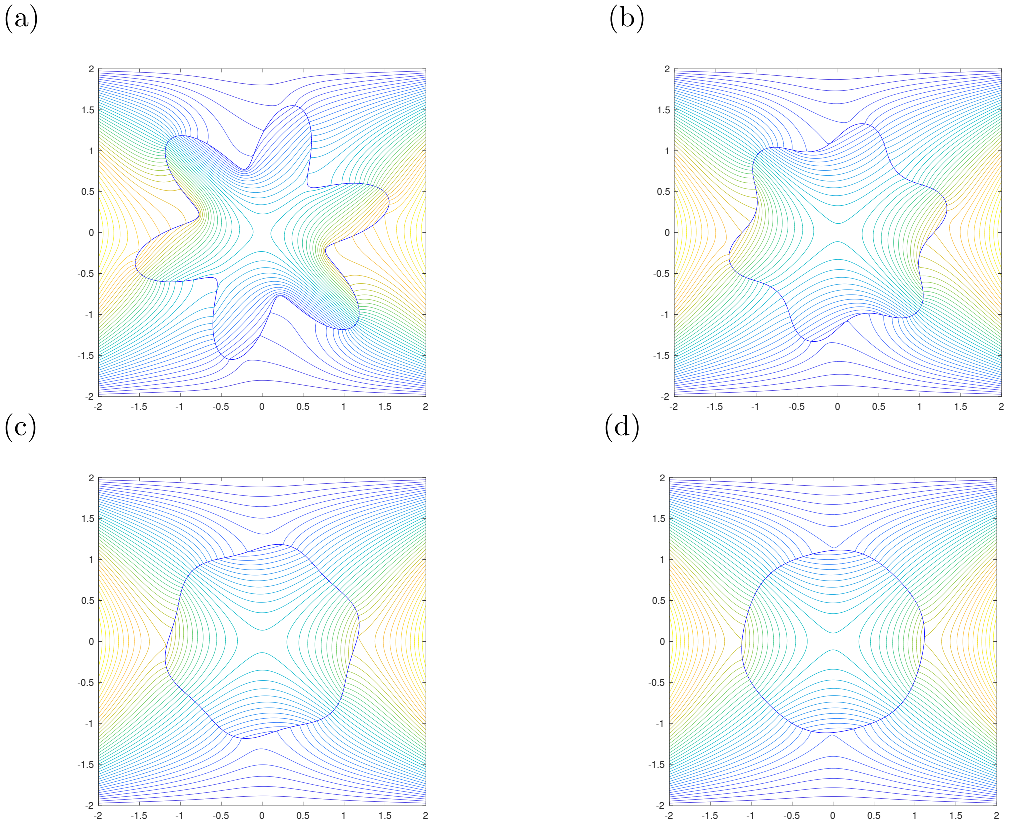

Figure 4.6: A moving interface driven by the mean curvature in the flow.

Chapter 5

A Direct IIM Approach for Two-phase Stokes Equations with Discontinuous Viscosity on Staggered Grids

In this chapter, we focus on solving the two-phase Stokes equations

$$\nabla p = \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{g} + \mathbf{F}, \quad \mathbf{x} \in \Omega, \tag{5.1}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}, \qquad \mathbf{x} \in \Omega, \tag{5.2}$$

$$\mathbf{u} \mid_{\partial\Omega} = \mathbf{u}_b, \tag{5.3}$$

where $\mathbf{u} = (u, v)^T$ is the fluid velocity, p is the fluid pressure, μ is the fluid viscosity, $\mathbf{u}_b$ is a Dirichlet boundary condition for the velocity, $\mathbf{g} = (g_1, g_2)^T$ is the body force and $\mathbf{F}$ is a singular source defined as

$$\mathbf{F} = \int_\Gamma \mathbf{f}(s)\delta(\mathbf{x} - \mathbf{X}(s))ds. \tag{5.4}$$

Here $\mathbf{X}(s)$ is a parametric form of the interface Γ with s being the parameter, for example, the arc-length, $\delta(\cdot)$ is the Dirac delta function, and $\mathbf{f} = (f_1, f_2)^T$ is the density function of the surface force along the interface. The problem is illustrated in Figure 1.3, where the domain Ω is divided into two subdomains, Ω^+ and Ω^-, by the interface Γ.

For the regularity requirement of the problem, we also assume that $\mu \geq \mu_0 > 0$, $\Gamma \in C^2$, and $\mathbf{f} \in C^1$. The divergence free condition (5.2) combined with divergence theorem leads to the compatibility condition for $\mathbf{u}_b$, i.e.

$$\int_{\partial\Omega} \mathbf{u}_b \cdot \mathbf{n}_b = 0, \tag{5.5}$$

where $\mathbf{n}_b$ is the unit normal vector pointing to the outside of $\partial\Omega$. The viscosity μ is assumed to be piecewise constant,

$$\mu(\mathbf{x}) = \begin{cases} \mu^+, & \mathbf{x} \in \Omega^+, \\ \mu^-, & \mathbf{x} \in \Omega^-. \end{cases} \tag{5.6}$$

Under this assumption, the incompressible Stokes equations (5.1)-(5.2) can be rewritten as

$$-\hat{u}_{xx} - \hat{u}_{yy} + p_x = g_1 \quad \mathbf{x} \in \Omega^{\pm}, \tag{5.7}$$

$$-\hat{v}_{xx} - \hat{v}_{yy} + p_y = g_2 \quad \mathbf{x} \in \Omega^{\pm}, \tag{5.8}$$

$$-\hat{u}_x - \hat{v}_y = 0 \quad \mathbf{x} \in \Omega^{\pm}, \tag{5.9}$$

where $(\hat{u}, \hat{v})^T$ is the transformed velocity field defined as

$$\hat{u} = \mu u, \quad \hat{v} = \mu v, \quad \hat{\mathbf{u}} = \mu \mathbf{u}. \tag{5.10}$$

Remark 5.1. *The viscosity μ is the coefficient for velocity terms but not for pressure and hence cannot be scaled as what is done for elliptic interface problems [46]. Instead we introduce the transformed velocity to incorporate the viscosity into the unknowns. This can help mitigate the scaling effect. The original velocity field (u, v) will be recovered once $(\hat{u}, \hat{v})$ is computed.*

5.1 Finite difference method using staggered grids

In this section, we will design a finite difference method which discretizes the above Stokes equations (5.7)-(5.9) on staggered grids.

5.1.1 Classification of grid points

As illustrated in Figure 5.1, the idea of MAC (Marker and Cell) Scheme is to place the unknowns (u, v, p) on different grid points. Specifically the pressure p is located in the center of each cell (formed by thick lines) and the horizontal velocity u on the middle points of vertical edges and the vertical velocity v on middle points of horizontal edges.

Assuming the domain is $\Omega = (a, b) \times (c, d)$, for simplification of discussion, we use the uniform mesh

$$x_i = a + ih, \quad i = 0, 1, \cdots M; \quad y_j = c + jh, \quad j = 0, 1, \cdots, N, \tag{5.11}$$

where h is the space step size. To be consistent with the grid indices used in (5.11), we keep using $u^{i,j}$, $v^{i,j}$, $p^{i,j}$ to represent the horizontal velocity, vertical velocity and pressure at the grid point (x_i, y_j) in the conventional way. However, $u^{i,j}$, $v^{i,j}$, $p^{i,j}$ are only defined on different disjoint sets of grid points. We classify a grid point (x_i, y_j) in the following sense,

$$\begin{aligned}
&(x_i, y_j) \text{ is a u-grid point if } (i \bmod 2) = 0 \text{ and } (j \bmod 2) = 1,\\
&(x_i, y_j) \text{ is a v-grid point if } (i \bmod 2) = 1 \text{ and } (j \bmod 2) = 0,\\
&(x_i, y_j) \text{ is a p-grid point if } (i \bmod 2) = 1 \text{ and } (j \bmod 2) = 1,
\end{aligned} \tag{5.12}$$

where "mod" is the modulus operator used to calculate arithmetic remainder. For example, we only place horizontal velocity u on u-grid points where the index i is even and j is odd, see Figure 5.1.

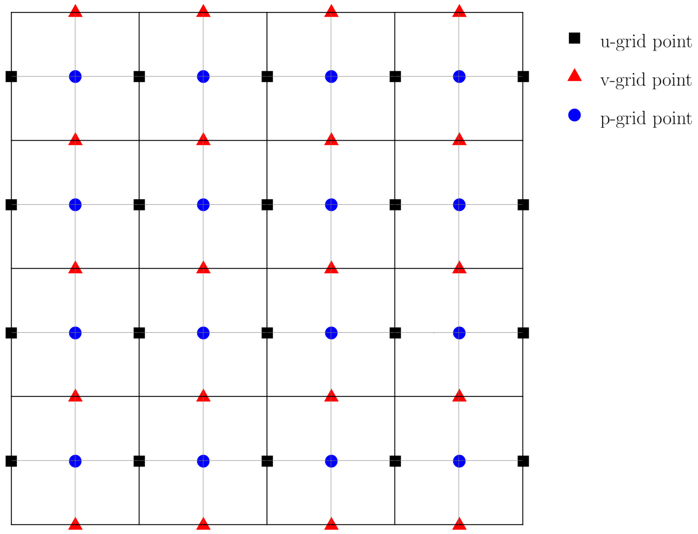

Figure 5.1: A diagram of the distribution of u, v, p-grid points inside a rectangular staggered grid mesh.

The interface Γ in this paper is represented by the zero level set of a Lipschitz continuous function $\phi(x, y)$, that is

$$\Gamma = \Big\{(x, y), \quad \phi(x, y) = 0, \quad (x, y) \in \Omega\Big\}. \tag{5.13}$$

The level set function $\phi(x, y)$ satisfies

$$\begin{aligned} \phi(x, y) < 0 &\quad \text{for } (x, y) \in \Omega^-, && (5.14)\\ \phi(x, y) = 0 &\quad \text{for } (x, y) \in \Gamma, && (5.15)\\ \phi(x, y) > 0 &\quad \text{for } (x, y) \in \Omega^+. && (5.16)\end{aligned}$$

In the neighborhood of the interface, we assume that $\phi(x, y) \in C^2$. In the implementation, the level set function is defined at the grid points as $\{\phi_{i,j}\}$. At a grid point (x_i, y_j), we define

$$\phi_{ij}^{max} = \begin{cases} \max\{\phi_{i-2,j}, \phi_{i,j}, \phi_{i+2,j}, \phi_{i,j-2}, \phi_{i,j+2}, \phi_{i-1,j}, \phi_{i+1,j}\}, & (x_i, y_j) \text{ is a u-grid point,}\\ \max\{\phi_{i-2,j}, \phi_{i,j}, \phi_{i+2,j}, \phi_{i,j-2}, \phi_{i,j+2}, \phi_{i,j-1}, \phi_{i,j+1}\}, & (x_i, y_j) \text{ is a v-grid point,}\\ \max\{\phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}\}, & (x_i, y_j) \text{ is a p-grid point,} \end{cases} \tag{5.17}$$

$$\phi_{ij}^{min} = \begin{cases} \min\{\phi_{i-2,j}, \phi_{i,j}, \phi_{i+2,j}, \phi_{i,j-2}, \phi_{i,j+2}, \phi_{i-1,j}, \phi_{i+1,j}\}, & (x_i, y_j) \text{ is a u-grid point,}\\ \min\{\phi_{i-2,j}, \phi_{i,j}, \phi_{i+2,j}, \phi_{i,j-2}, \phi_{i,j+2}, \phi_{i,j-1}, \phi_{i,j+1}\}, & (x_i, y_j) \text{ is a v-grid point,}\\ \min\{\phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}\}, & (x_i, y_j) \text{ is a p-grid point.} \end{cases} \tag{5.18}$$

A grid point (x_i, y_j) is called *regular* if $\phi_{ij}^{max}\phi_{ij}^{min} > 0$, otherwise it is called irregular. The interface information, such as the normal unit vector and signed curvature, etc., can be easily computed from

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{5.19}$$

$$\kappa = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}. \tag{5.20}$$

Remark 5.2. *Notice that for the grid mesh introduced in* (5.11)*, not every grid point is used as one can see from the classification definition* (5.12)*. The mesh interval size for the same type of grid points is actually $H = 2h$, which is used in finite difference scheme.*

5.1.2 The marker and cell scheme

The MAC scheme is to discretize the x-coordinate momentum equation (5.7) at vertical edges of cells, the y-coordinate momentum equation (5.8) at horizontal edges of cells, and the continuity equation (5.9) at cell centers using central difference schemes. In the following discussion, we will use notation $U_{i,j}$, $V_{i,j}$, $P_{i,j}$ $\hat{U}_{i,j}$, $\hat{V}_{i,j}$ to represent the numerical solutions of u, v, p, $\hat{u}$ and

$\hat{v}$ at the corresponding grid point (x_i, y_j) respectively. First we assume (x_i, y_j) is a regular grid point. Using the index system introduced in Section 5.1.1, the MAC scheme can be written as

$$\frac{4\hat{U}^{i,j} - \hat{U}^{i-2,j} - \hat{U}^{i+2,j} - \hat{U}^{i,j-2} - \hat{U}^{i,j+2}}{H^2} + \frac{P^{i+1,j} - P^{i-1,j}}{H} = g_1^{i,j}, \tag{5.21}$$

$$\frac{4\hat{V}^{i,j} - \hat{V}^{i-2,j} - \hat{V}^{i+2,j} - \hat{V}^{i,j-2} - \hat{V}^{i,j+2}}{H^2} + \frac{P^{i,j+1} - P^{i,j-1}}{H} = g_2^{i,j}, \tag{5.22}$$

$$-\frac{\hat{U}^{i+1,j} - \hat{U}^{i-1,j}}{H} - \frac{\hat{V}^{i,j+1} - \hat{V}^{i,j-1}}{H} = 0, \tag{5.23}$$

for a u-grid point, a v-grid point and a p-grid point, where $H = 2h$. See Figure 5.2 for an illustration. Next assume (x_i, y_j) is an irregular u-grid point. We will discretize the x-coordinate momentum equation (5.7) term by term. The finite difference approximations for $\hat{u}_{xx}$, $\hat{u}_{yy}$ and p_x have the following forms respectively.

$$\hat{u}_{xx} \approx \frac{\hat{U}_{i-2,j} - 2\hat{U}_{i,j} + \hat{U}_{i+2,j} + \hat{C}^u_{i,j}}{H^2}, \tag{5.24}$$

$$\hat{u}_{yy} \approx \frac{\hat{U}_{i,j-2} - 2\hat{U}_{i,j} + \hat{U}_{i,j+2} + \bar{C}^u_{i,j}}{H^2}, \tag{5.25}$$

$$p_x \approx \frac{P_{i+1,j} - P_{i-1,j} + \tilde{C}^u_{i,j}}{H}, \tag{5.26}$$

where $\hat{C}^u_{i,j}$, $\bar{C}^u_{i,j}$ and $\tilde{C}^u_{i,j}$ stand for some correction terms to be determined and the superscript u indicates they are used at u-grid points.

Before deriving the formulas for the correction terms, we introduce the notation $[\cdot]$ for a jump across the interface Γ and it is defined for an arbitrary function $q(\mathbf{X})$ along the interface at $\mathbf{X}$ by

$$[q] = [q]_\Gamma(\mathbf{X}) = q^+(\mathbf{X}) - q^-(\mathbf{X}), \tag{5.27}$$

where $q^+(\mathbf{X})$, and $q^-(\mathbf{X})$ are the limiting values of $q(\mathbf{X})$ from Ω^+ and Ω^- sides, respectively. Now let us first assume that the interface does not cut through the interval (x_{i-2}, x_{i+2}) along the line $y = y_j$, but cuts the grid line $x = x_i$ in the interval (y_j, y_{j+2}) at (x_i, y_j^*), see Figure 5.2a for an illustration. In this case, (x_i, y_j) is regular in the x-direction but irregular in the y-direction. Hence, we have zero x-direction correction terms and nonzero one for y-direction. Without loss of generality, we assume that $(x_i, y_j) \in \Omega^-$. By using Taylor expansion, one can derive the formula for the correction terms.

$$\hat{C}^u_{i,j} = \tilde{C}^u_{i,j} = 0, \tag{5.28}$$

$$\bar{C}^u_{i,j} = -[\hat{u}] - (y_{j+2} - y_j^*)[\hat{u}_y] - \frac{(y_{j+2} - y_j^*)^2}{2}[\hat{u}_{yy}]. \tag{5.29}$$

In this particular case, the corrected value $\hat{U}_{i,j+2} + \bar{C}^u_{i,j}$ in (5.22) can be interpreted as the smooth extension of the solution at (x_i, y_{j+2}) from Ω^- side. Next we assume the interface does not cut through the interval (y_{j-2}, y_{j+2}) along the line $x = x_i$, but cuts the grid line $y(x) = y_j$ in the interval (x_i, x_{i+2}) at (x_i^*, y_j), see Figure 5.2b for an illustration. This time, we assume $(x_i, y_j) \in \Omega^+$. One can then derive the following formula for the correction terms in a similar way.

$$\hat{C}^u_{i,j} = [\hat{u}] + (x_{i+2} - x_i^*)[\hat{u}_x] + \frac{(x_{i+2} - x_i^*)^2}{2}[\hat{u}_{xx}], \tag{5.30}$$

$$\bar{C}^u_{i,j} = 0, \tag{5.31}$$

$$\tilde{C}^u_{i,j} = \begin{cases} [p] + (x_{i+1} - x_i^*)[p_x], & \text{if } x_i^* \in (x_i, x_{i+1}) \\ 0, & \text{if } x_i^* \in (x_{i+1}, x_{i+2}), \end{cases} \tag{5.32}$$

where in (5.32), we intentionally choose open intervals to avoid confusion. It is possible to have x_i^* overlap the grid points. In the implementation, one can avoid this issue by simply categorizing those on-interface points as either inside or outside of the interface.

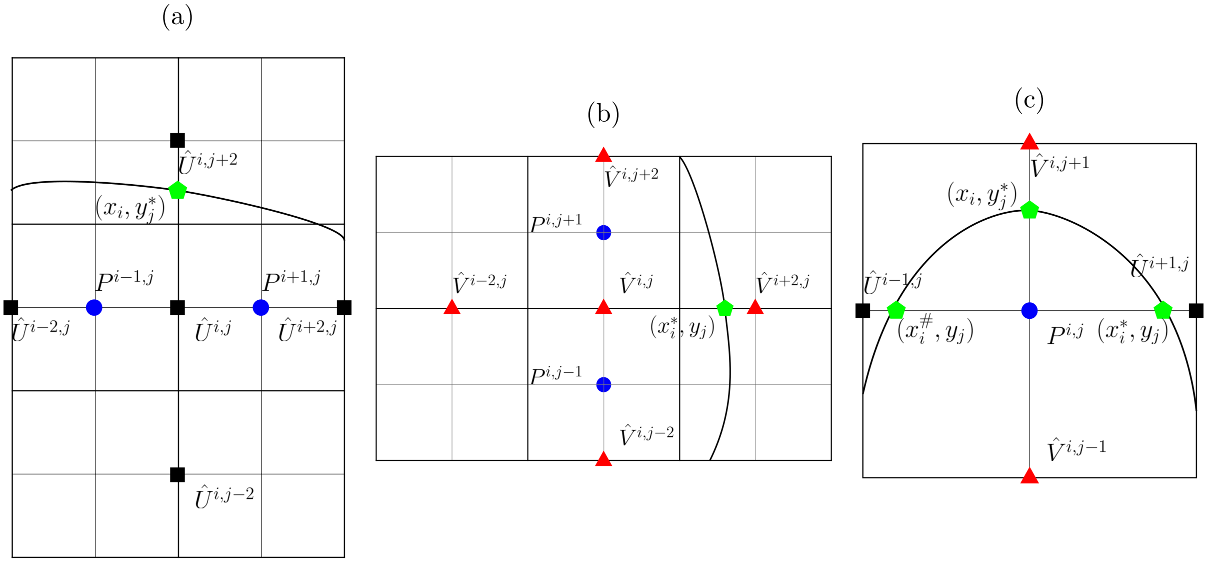

Figure 5.2: (a): MAC scheme for the x-direction momentum equation. (b): MAC scheme for the y-direction momentum equation. (c): MAC scheme for the continuity equation.

Now let us take a look at some irregular p-grid point $(x_i, y_j) \in \Omega^-$. The approximations for

$\hat{u}_x$ and $\hat{v}_y$ have the following forms

$$\hat{u}_x \approx \frac{\hat{U}_{i+1,j} - \hat{U}_{i-1,j} + \hat{C}^p_{i,j}}{H}, \tag{5.33}$$

$$\hat{v}_y \approx \frac{\hat{V}_{i,j+1} - \hat{V}_{i,j-1} + \bar{C}^p_{i,j}}{H}. \tag{5.34}$$

Consider a extreme case where the interface cuts through the interval (x_{i-1}, x_i) at $(x_i^{\#}, y_j)$, (x_i, x_{i+1}) at (x_i^*, y_j) and (y_j, y_{j+1}) at (x_i, y_j^*), see Figure 5.2c for an illustration. Then the formula for the correction terms are

$$\hat{C}^p_{i,j} = (x_{i-1} - x_i^{\#})[\hat{u}_x]_u^{\#} - (x_{i+1} - x_i^*)[\hat{u}_x]_u^* + \frac{(x_{i-1} - x_i^{\#})^2}{2}[\hat{u}_{xx}]_u^{\#} - \frac{(x_{i+1} - x_i^*)^2}{2}[\hat{u}_{xx}]_u^*, \tag{5.35}$$

$$\bar{C}^p_{i,j} = -(y_{j+1} - y_j^*)[\hat{v}_y]_v^* - \frac{(y_{j+1} - y_j^*)^2}{2}[\hat{v}_{yy}]_v^*, \tag{5.36}$$

where $[\cdot]_u^{\#}$, $[\cdot]_u^*$, $[\cdot]_v^*$ represents jumps at the intersection points $(x_i^{\#}, y_j)$, (x_i^*, y_j) and (x_i, y_j^*) respectively. The formula for the correction terms used in the y-coordinate momentum equation (5.8) and all the other possible situations can be derived in a similar way. Combining all the correction terms used at each grid point, the modified MAC schemes look like

$$\frac{4\hat{U}^{i,j} - \hat{U}^{i-2,j} - \hat{U}^{i+2,j} - \hat{U}^{i,j-2} - \hat{U}^{i,j+2}}{H^2} + \frac{P^{i+1,j} - P^{i-1,j}}{H} + C^u_{i,j} = g_1^{i,j}, \tag{5.37}$$

$$\frac{4\hat{V}^{i,j} - \hat{V}^{i-2,j} - \hat{V}^{i+2,j} - \hat{V}^{i,j-2} - \hat{V}^{i,j+2}}{H^2} + \frac{P^{i,j+1} - P^{i,j-1}}{H} + C^v_{i,j} = g_2^{i,j}, \tag{5.38}$$

$$-\frac{\hat{U}^{i+1,j} - \hat{U}^{i-1,j}}{H} - \frac{\hat{V}^{i,j+1} - \hat{V}^{i,j-1}}{H} + C^p_{i,j} = 0. \tag{5.39}$$

Then $(\hat{U}, \hat{V}, P)$ can be solved if the correction terms $C^u_{i,j}$, $C^v_{i,j}$, $C^p_{i,j}$ can be determined at each grid point.

Remark 5.3. *To get a second order finite difference stencil for the second derivative ($\hat{\mathbf{u}}_{xx}$, $\hat{\mathbf{u}}_{yy}$), the correction term needs the information of the third derivative jump. In the same way, a second order finite difference stencil for the first derivative ($\hat{\mathbf{u}}_x$, $\hat{\mathbf{u}}_y$, p_x, p_y) requires the information about second derivative jump. From the formulations of correction terms* (5.28)-(5.32) *and* (5.35)-(5.36)*, one can see that at the irregular grid points, the finite difference schemes for the momentum equations have first order truncation error while that for the continuity equation is second order. Empirically, one lower order of truncation error at the interface or the boundary will not affect overall second order accuracy of the solution.*

5.1.3 Jump relations and the coordinates transformation

To be self-contained, the following contents are presented and cited from [35, 38]. Assume that the interface in the parametric form is

$$\Gamma = \{(X(s), Y(s)), \quad X(s) \in C^2, Y(s) \in C^2\}, \tag{5.40}$$

where s is a parameter, for example, the arc-length. At a point of the interface (X, Y), the local coordinate system in the normal and tangential directions is defined as

$$\begin{cases} \xi = (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta, \end{cases} \tag{5.41}$$

where θ is the angle between the x-axis and the normal direction, pointing to the Ω^+ subdomain, see Figure 5.3 for an illustration. Under the new coordinates system, the interface can be parameterized by

$$\xi = \chi(\eta) \quad \text{with} \quad \chi(0) = 0, \quad \chi'(0) = 0. \tag{5.42}$$

The formula for the signed curvature $\kappa = \chi''(0)$ of the interface at (X, Y) is given by (5.20). Given the angle θ, the unit normal and tangential vector at the point (X, Y) can be expressed as

$$\mathbf{n} = (\cos\theta, \sin\theta), \quad \boldsymbol{\tau} = (-\sin\theta, \cos\theta). \tag{5.43}$$

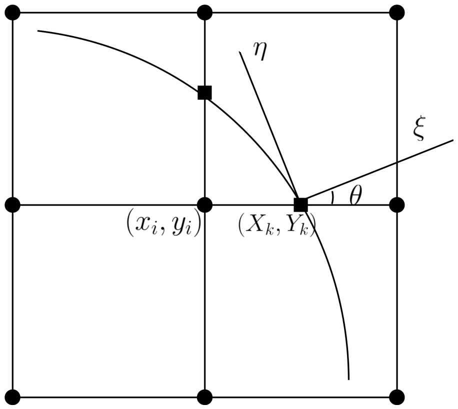

Figure 5.3: A diagram of an irregular grid point (x_i, y_i), its control point (X_k, Y_k) on the x-axis, and the local coordinates at (X_k, Y_k) in the normal and tangential directions, where θ is the angle between the x-axis and the normal direction.

Given the continuity conditions $[\mathbf{u}] = \mathbf{0}$, $\nabla \cdot \mathbf{u} = 0$ and momentum equations (5.7)-(5.8), one can derive the following jump relations at a point (X, Y) on the interface which are necessary to derive the accurate finite difference method.

Theorem 5.1. *For Stokes equations* (5.1) *with piecewise constant coefficient* (5.6)*, given the interface information,* $\kappa = \chi''(0)$*, and the continuity conditions* $[\mathbf{u}] = \mathbf{0}$*,* $\nabla \cdot \mathbf{u} = 0$*, one can derive the following jump relations for the velocity* $\mathbf{u}$ *in the local coordinates.*

$$[\mathbf{u}] = 0, \tag{5.44}$$

$$[\mathbf{u}_\eta] = 0, \tag{5.45}$$

$$[\mu \mathbf{u}_\eta] \cdot \boldsymbol{\tau} + [\mu \mathbf{u}_\xi] \cdot \mathbf{n} = 0, \tag{5.46}$$

$$[\mu \mathbf{u}_\eta] \cdot \mathbf{n} + [\mu \mathbf{u}_\xi] \cdot \boldsymbol{\tau} + \hat{f}_2 = 0, \tag{5.47}$$

$$[\mathbf{u}_{\eta\eta}] = -\kappa [\mathbf{u}_\xi], \tag{5.48}$$

$$[\mu \mathbf{u}_{\eta\eta}] \cdot \boldsymbol{\tau} + [\mu \mathbf{u}_{\eta\xi}] \cdot \mathbf{n} = 0, \tag{5.49}$$

$$[\mu \mathbf{u}_{\eta\xi}] \cdot \boldsymbol{\tau} + [\mu \mathbf{u}_{\xi\xi}] \cdot \mathbf{n} = 0, \tag{5.50}$$

$$[\mu \mathbf{u}_{\eta\eta}] \cdot \mathbf{n} + [\mu \mathbf{u}_{\eta\xi}] \cdot \boldsymbol{\tau} = 4\kappa [\mu \mathbf{u}_\eta] \cdot \boldsymbol{\tau} - \hat{f}_2', \tag{5.51}$$

$$3[\mu \mathbf{u}_{\eta\xi}] \cdot \mathbf{n} - [\mu \mathbf{u}_{\xi\xi}] \cdot \boldsymbol{\tau} = [\mathbf{g}] \cdot \boldsymbol{\tau} - \hat{f}_1' - 2\kappa \hat{f}_2, \tag{5.52}$$

where $\hat{f}_1 = \mathbf{f} \cdot \mathbf{n}$ *and* $\hat{f}_2 = \mathbf{f} \cdot \boldsymbol{\tau}$ *are used to represent the force densities in the normal and tangential directions, respectively. The jump relations for the pressure* p *are more convenient to be expressed in terms of jumps of the velocity* $\mathbf{u}$ *in the global coordinates and are given as the following*

$$[p] = 2\left[\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] \cdot \mathbf{n} + \hat{f}_1, \tag{5.53}$$

$$[p_x] = [\mu u_{xx}] + [\mu u_{yy}] + [g_1], \tag{5.54}$$

$$[p_y] = [\mu v_{xx}] + [\mu v_{yy}] + [g_2], \tag{5.55}$$

which are proved in Theorem (10.2) in Section 10.1 in [38]. After some algebraic manipulations, it turns out that given $[\hat{\mathbf{u}}]$, $[\hat{\mathbf{u}}_\eta]$, and $[\hat{\mathbf{u}}_{\eta\eta}]$, all the jump relations for the scaled velocity $\hat{\mathbf{u}}$ will be known and are given below

$$[\hat{u}_\xi] = 2sc[\hat{u}_\eta] - (c^2 - s^2)[\hat{v}_\eta] + s\hat{f}_2, \tag{5.56}$$

$$[\hat{v}_\xi] = -(c^2 - s^2)[\hat{u}_\eta] - 2sc[\hat{v}_\eta] - c\hat{f}_2, \tag{5.57}$$

$$[\hat{u}_{\eta\xi}] = cD_1 - sD_2, \quad [\hat{u}_{\xi\xi}] = -sD_1 - cD_2 + sD_3, \tag{5.58}$$

$$[\hat{v}_{\eta\xi}] = sD_1 + cD_2, \quad [\hat{v}_{\xi\xi}] = cD_1 - sD_2 - cD_3, \tag{5.59}$$

where s, c stand for $\sin\theta$ and $\cos\theta$ respectively, which are defined in (5.41). The terms D_1-D_3 are expressed in terms of $[\hat{\mathbf{u}}_\eta]$, and $[\hat{\mathbf{u}}_{\eta\eta}]$ only and are given below

$$D_1 = -[\hat{\mathbf{u}}_{\eta\eta}] \cdot \boldsymbol{\tau}, \tag{5.60}$$

$$D_2 = -[\hat{\mathbf{u}}_{\eta\eta}] \cdot \mathbf{n} + 4\kappa[\hat{\mathbf{u}}_\eta]\boldsymbol{\tau} - \hat{f}_2', \tag{5.61}$$

$$D_3 = 2[\hat{\mathbf{u}}_{\eta\eta}] \cdot \boldsymbol{\tau} + [\mathbf{g}] \cdot \boldsymbol{\tau} - \hat{f}_1' - 2\kappa\hat{f}_2. \tag{5.62}$$

Now the only unknowns are $[\hat{\mathbf{u}}]$, $[\hat{\mathbf{u}}_\eta]$ and $[\hat{\mathbf{u}}_{\eta\eta}]$. Although we do not know the exact values for these terms, fortunately, they can be formulated in terms of the limiting values and derivatives of $\hat{\mathbf{u}}$ from only one side of the interface, i.e.

$$[\hat{u}] = (1/\rho - 1)\hat{u}^-, \quad [\hat{v}] = (1/\rho - 1)\hat{v}^-, \tag{5.63}$$

$$[\hat{u}_\eta] = (1/\rho - 1)\hat{u}_\eta^-, \quad [\hat{v}_\eta] = (1/\rho - 1)\hat{v}_\eta^-, \tag{5.64}$$

$$[\hat{u}_{\eta\eta}] = (1/\rho - 1)\hat{u}_{\eta\eta}^- + \kappa(1/\rho - 1)\hat{u}_\xi^- - \kappa[\hat{u}_\xi], \tag{5.65}$$

$$[\hat{v}_{\eta\eta}] = (1/\rho - 1)\hat{v}_{\eta\eta}^- + \kappa(1/\rho - 1)\hat{v}_\xi^- - \kappa[\hat{v}_\xi], \tag{5.66}$$

where $\rho = \mu^-/\mu^+$ is the jump ratio. This makes it possible to write down the Taylor expansion of a discontinuous function about a limiting jump point from another side of the interface by utilizing the jump relations and hence derive the interpolation schemes for the correction terms. Finally, since we are only interested in the jumps defined in global x-y coordinates, the coordinate transformation matrices are provided below

$$\begin{bmatrix} T_\eta \\ T_\xi \end{bmatrix} = \begin{bmatrix} -s & c \\ c & s \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \quad \begin{bmatrix} T_{\eta\eta} \\ T_{\eta\xi} \\ T_{\xi\xi} \end{bmatrix} = \begin{bmatrix} s^2 & -2sc & c^2 \\ -sc & c^2 - s^2 & sc \\ c^2 & 2sc & s^2 \end{bmatrix} \begin{bmatrix} T_{xx} \\ T_{xy} \\ T_{yy} \end{bmatrix}, \tag{5.67}$$

where T is a function defined on Ω, s and c stand for $\sin\theta$ and $\cos\theta$. Notice that both the transformation matrices are involutory, which means they are their own inverses. Hence, they can be applied for the transformation from local into Cartesian coordinates as well.

Remark 5.4. *The idea of the IIM is to embed the jump relations into the approximation schemes for the correction terms. It is known that an interpolation scheme can be derived by the method of undetermined coefficients using Taylor expansion. In this approach, the solution and its derivatives at a certain point are regarded as a basis to form a system of equations, from which the coefficients of the interpolation scheme can be solved. However, if one expands a variable, let us say $\hat{u}(x, y)$, about the point on the interface, there would be two possible sets of basis to use based on whether (x, y) lies on the "+" or "−" side of the interface, say $\{\hat{u}^-$, $\hat{u}_x^-$, $\hat{u}_y^-$, $\hat{u}_{xx}^-$, $\hat{u}_{xy}^-$, $\hat{u}_{yy}^-\}$ or $\{\hat{u}^+$, $\hat{u}_x^+$, $\hat{u}_y^+$, $\hat{u}_{xx}^+$, $\hat{u}_{xy}^+$, $\hat{u}_{yy}^+\}$. This is where the jump relations play* *a role. One can apply the jump relations to convert one basis into another one. The details are explained in the next section.*

5.1.4 The approximation of correction terms

In this section, we describe how to interpolate the correction terms at the irregular grid points. Here we derive the scheme for $C^u_{i,j}$ used in (5.37) as an example. We will use the jump relations explained in Section 5.1.3 to rewrite the correction term $C^u_{i,j}$ and the Taylor expansions of the solution $(\hat{u}, \hat{v}, p)$ at irregular grid points in terms of the limiting values $\hat{u}^-$, $\hat{u}^-_x$, $\hat{u}^-_y$, $\hat{u}^-_{xx}$, $\hat{u}^-_{xy}$, $\hat{u}^-_{yy}$ and $\hat{v}^-$, $\hat{v}^-_x$, $\hat{v}^-_y$, $\hat{v}^-_{xx}$, $\hat{v}^-_{xy}$, $\hat{v}^-_{yy}$, p^-. Using the method of undetermined coefficients, a compact 33-point interpolation scheme can be derived.

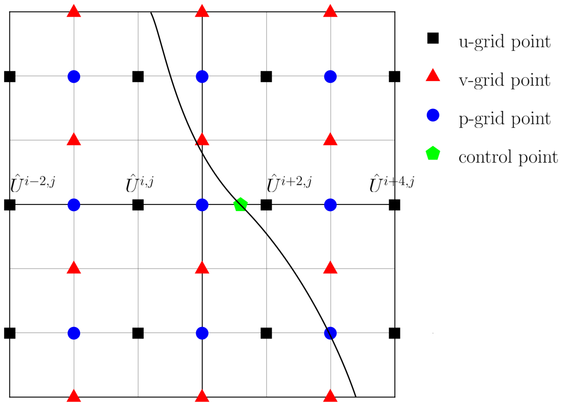

Figure 5.4: A diagram of the distribution of u, v, p-grid points involved in the interpolation scheme for the correction terms.

Let (x_i, y_j) be an irregular u-grid point. Without loss of generality, we assume the interface cuts through the interval (x_i, x_{i+2}) along the line $y = y_j$ and intersects at the point (x^*_i, y_j). The interpolation scheme for $C^u_{i,j}$ involves total thirty-three grid points and has the form

$$C^u_{i,j} = \sum_{i^u_k, j^u_k} \gamma_{i^u_k, j^u_k} \hat{U}_{i+i^u_k, j+j^u_k} + \sum_{i^v_k, j^v_k} \gamma_{i^v_k, j^v_k} \hat{V}_{i+i^v_k, j+j^v_k} + \sum_{i^p_k, j^p_k} h \cdot \gamma_{i^p_k, j^p_k} P_{i+i^p_k, j+j^p_k} + \gamma_c, \tag{5.68}$$

where

$$i^u_k = 0, \pm 2, 4, \quad j^u_k = 0, \pm 2, \quad i^v_k = \pm 1, 3, \quad j^v_k = \pm 1, \pm 3, \quad i^p_k = \pm 1, 3, \quad j^p_k = 0, \pm 2, \tag{5.69}$$

see Figure 5.4 for an illustration. The interpolation scheme that we proposed involves twelve u-grid points, twelve v-grid points and nine p-grid points, which contains enough points for obtaining second order accuracy. Note that we put weight h for p-grid points (see Remark 5.5). The Taylor expansions of $\hat{u}(x_{i+i_k^u}, y_{j+j_k^u})$ and $p(x_{i+i_k^p}, y_{j+j_k^p})$ about (x_i^*, y_j) have the forms

$$\begin{aligned}\hat{u}(x_{i+i_k^u}, y_{j+j_k^u}) &= \hat{u}^{\pm} + (x_{i+i_k^u} - x_i^*)\hat{u}_x^{\pm} + (y_{j+j_k^u} - y_j)\hat{u}_y^{\pm} + \frac{1}{2}(x_{i+i_k^u} - x_i^*)^2\hat{u}_{xx}^{\pm} \\ &+ \frac{1}{2}(y_{j+j_k^u} - y_j)^2\hat{u}_{yy}^{\pm} + (x_{i+i_k^u} - x_i^*)(y_{j+j_k^u} - y_j)\hat{u}_{xy}^{\pm} + O(h^3),\end{aligned} \tag{5.70}$$

and

$$p(x_{i+i_k^p}, y_{j+j_k^p}) = p^{\pm} + (x_{i+i_k^p} - x_i^*)p_x^{\pm} + (y_{j+j_k^p} - y_j)p_y^{\pm} + O(h^3), \tag{5.71}$$

where the "+" or "−" sign is chosen depending on whether $(x_{i+i_k^u}, y_{j+j_k^u})$ and $(x_{i+i_k^p}, y_{j+j_k^p})$ lie on the "+" or "−" side of Γ. Here the Taylor expansion of $\hat{v}(x_{i+i_k^v}, y_{j+j_k^v})$ is similar to (5.70) and hence is ignored for simplicity. Note that the right-hand side of the above equations can be expressed in terms of $\hat{u}^-$, $\hat{u}_x^-$, $\hat{u}_y^-$, $\hat{u}_{xx}^-$, $\hat{u}_{xy}^-$, $\hat{u}_{yy}^-$ and $\hat{v}^-$, $\hat{v}_x^-$, $\hat{v}_y^-$, $\hat{v}_{xx}^-$, $\hat{v}_{xy}^-$, $\hat{v}_{yy}^-$, p^- by using the jump relations (5.56)-(5.59) and (5.63)-(5.66). Take $\hat{u}(x_{i+i_k^u}, y_{j+j_k^u})$ for example, we will have the following form

$$\begin{aligned}\hat{u}(x_{i+i_k^u}, y_{j+j_k^u}) &= c_1\hat{u}^- + c_2\hat{u}_x^- + c_3\hat{u}_y^- + c_4\hat{u}_{xx}^- + c_5\hat{u}_{xy}^- + c_6\hat{u}_{yy}^- + c_7\hat{v}^- \\ &+ c_8\hat{v}_x^- + c_9\hat{v}_y^- + c_{10}\hat{v}_{xx}^- + c_{11}\hat{v}_{xy}^- + c_{12}\hat{v}_{yy}^- + c_{13}p^- + c_{14},\end{aligned} \tag{5.72}$$

where the coefficients c_1-c_{14} are determined by the jump relations. Applying the divergence-free condition (5.9) and replacing $\hat{\mathbf{u}}$, its derivatives, and p by the numerical solution, the above equation (5.72) can be simplified as

$$\begin{aligned}\hat{U}(x_{i+i_k^u}, y_{j+j_k^u}) &= c^1_{i_k^u,j_k^u}\hat{U}^- + c^2_{i_k^u,j_k^u}\hat{U}_x^- + c^3_{i_k^u,j_k^u}\hat{U}_y^- + c^4_{i_k^u,j_k^u}\hat{U}_{xx}^- + c^5_{i_k^u,j_k^u}\hat{U}_{xy}^- + c^6_{i_k^u,j_k^u}\hat{U}_{yy}^- \\ &+ c^7_{i_k^u,j_k^u}\hat{V}^- + c^8_{i_k^u,j_k^u}\hat{V}_x^- + c^9_{i_k^u,j_k^u}\hat{V}_{xx}^- + c^{10}_{i_k^u,j_k^u}P^- + c^{11}_{i_k^u,j_k^u},\end{aligned} \tag{5.73}$$

where

$$\begin{aligned}&c^2_{i_k^u,j_k^u} = c_2 - c_9, \quad c^4_{i_k^u,j_k^u} = c_4 - c_{11}, \quad c^5_{i_k^u,j_k^u} = c_5 - c_{12}, \\ &c^9_{i_k^u,j_k^u} = c_{10}, \quad c^{10}_{i_k^u,j_k^u} = c_{13}, \quad c^{11}_{i_k^u,j_k^u} = c_{14}, \quad c^l_{i_k^u,j_k^u} = c_l \text{ for the others.}\end{aligned} \tag{5.74}$$

Similarly, $\hat{V}(x_{i+i_k^v}, y_{j+j_k^v})$, $P(x_{i+i_k^p}, y_{j+j_k^p})$ have the same form as equation (5.73) but with different coefficient set $\{c^l_{i_k^v,j_k^v}\}$ and $\{c^l_{i_k^p,j_k^p}\}$ respectively. Using the same approach, the correction term $C^u_{i,j}$ given by (5.28)-(5.29) or (5.30)-(5.32) can be reformulated as

$$\begin{aligned}C^u_{i,j} &= a_1^u\hat{U}^- + a_2^u\hat{U}_x^- + a_3^u\hat{U}_y^- + a_4^u\hat{U}_{xx}^- + a_5^u\hat{U}_{xy}^- + a_6^u\hat{U}_{yy}^- \\ &+ a_7^u\hat{V}^- + a_8^u\hat{V}_x^- + a_9^u\hat{V}_{xx}^- + a_{10}^uP^- + a_{11}^u.\end{aligned} \tag{5.75}$$

Now substituting $\hat{U}(x_{i+i_k^u}, y_{j+j_k^u})$, $\hat{V}(x_{i+i_k^v}, y_{j+j_k^v})$, and $P(x_{i+i_k^p}, y_{j+j_k^p})$ specified in (5.73) back into (5.68) and setting this equation equal to (5.75) yields the following system of equations

$$\begin{array}{c} \\ u^- \\ u_x^- \\ u_y^- \\ u_{xx}^- \\ u_{xy}^- \\ u_{yy}^- \\ v^- \\ v_x^- \\ v_{xx}^- \\ p^- \end{array}
\begin{array}{c} \begin{array}{ccccc} \hat{u}_{i-2,j-2} & \cdots & \hat{v}_{i+1,j+1} & \cdots & h\cdot p_{i+3,j+2} \end{array} \\
\begin{bmatrix}
c^1_{-2,-2} & \cdots & c^1_{1,1} & \cdots & h\cdot c^1_{3,2} \\
c^2_{-2,-2} & \cdots & c^2_{1,1} & \cdots & h\cdot c^2_{3,2} \\
c^3_{-2,-2} & \cdots & c^3_{1,1} & \cdots & h\cdot c^3_{3,2} \\
c^4_{-2,-2} & \cdots & c^4_{1,1} & \cdots & h\cdot c^4_{3,2} \\
c^5_{-2,-2} & \cdots & c^5_{1,1} & \cdots & h\cdot c^5_{3,2} \\
c^6_{-2,-2} & \cdots & c^6_{1,1} & \cdots & h\cdot c^6_{3,2} \\
c^7_{-2,-2} & \cdots & c^7_{1,1} & \cdots & h\cdot c^7_{3,2} \\
c^8_{-2,-2} & \cdots & c^8_{1,1} & \cdots & h\cdot c^8_{3,2} \\
c^9_{-2,-2} & \cdots & c^9_{1,1} & \cdots & h\cdot c^9_{3,2} \\
c^{10}_{-2,-2} & \cdots & c^{10}_{1,1} & \cdots & h\cdot c^{10}_{3,2}
\end{bmatrix} \end{array}
\begin{bmatrix} \gamma_{-2,-2} \\ \vdots \\ \gamma_{1,1} \\ \vdots \\ \gamma_{3,2} \end{bmatrix}
=
\begin{bmatrix} a^u_1 \\ a^u_2 \\ a^u_3 \\ a^u_4 \\ a^u_5 \\ a^u_6 \\ a^u_7 \\ a^u_8 \\ a^u_9 \\ a^u_{10} \end{bmatrix}. \tag{5.76}$$

The linear system (5.76) contains 9 equations with 33 unknown variables and is underdetermined. There are infinitely many solutions for this system. One can use singular value decomposition (SVD) to get the solution $\{\gamma_{i_k,j_k}\}$ and set

$$\gamma_c = a^u_{11} - \sum_{i_k,j_k} c^{11}_{i_k,j_k} \cdot \gamma_{i_k,j_k}, \tag{5.77}$$

where (i_k, j_k) takes all the values from (i_k^u, j_k^u), (i_k^v, j_k^v), and (i_k^p, j_k^p). Note that using more points than necessary for second order accuracy in (5.76) and then applying SVD are important for accuracy of gradient and the stability. In this way, the SVD solution not only satisfies the equations, but also has the minimum 2-norm among all possible solutions. In other words, the magnitude of the interpolation coefficients are well balanced.

For the other situations, one can first find the closest p-grid point to the control point (intersection point) and use nearby thirty-three points (see Figure 5.4) to derive the interpolation scheme in the similar way. Up to this point, the finite difference scheme for the Stokes equations (5.7)-(5.9) is fully determined

Remark 5.5. *Notice that we add a factor of h to the coefficients of $\{P_{i+i_k^p,j+j_k^p}\}$ in the interpolation scheme* (5.68)*. This is used to control the size of the coefficients $\{h\gamma_{i_k^p,j_k^p}\}$ for the p-grid points. Since the system* (5.76) *is solved by SVD and all the coefficients $\{\gamma_{i_k,j_k}\}$ have the same level of magnitude, the real coefficients of $\{P_{i+i_k^p,j+j_k^p}\}$ in the interpolation scheme are actually scaled down by a factor of h. This is consistent with the original MAC scheme where the coefficients for $\hat{U}$ are $O(1/H^2)$ and those for P are $O(1/H)$ in the momentum equations.*

5.1.5 Solving the discrete Stokes equations

The discrete Stokes equations can be written as the following matrix-vector form,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}} \\ P \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}. \tag{5.78}$$

Our goal in this section is to design an efficient solver for this linear system. Notice that the coefficient matrix is rank-one deficient since no boundary condition for pressure p is used. Hence the system (5.78) cannot be solved by regular linear solvers. If we simplify our problem by making $\mu_1 = \mu_2$, then in (5.78) we will have $C = B^T$, $D = \mathbf{0}$, A is block diagonal with two blocks $A1$, $A2$, and $A1$, $A2$, CB are negative discrete Laplacian for $-\Delta\hat{u}$, $-\Delta\hat{v}$, and $-\Delta p$ respectively. In this case, the coefficient matrix is the same as what is obtained from the standard discretization using MAC scheme for the saddle point problem, and the system can be solved by Uzawa method. This motivates us to design a Uzawa type method to solve our problem.

The idea of the Uzawa method is to use Richardson method to solve the Schur complement system

$$SP = G, \tag{5.79}$$

where $S = CA^{-1}B - D$, and $G = CA^{-1}F_1 - F_2$. And then use the computed pressure P to update the velocity $\hat{\mathbf{U}}$ by solving the momentum equations. One iteration of Uzawa method has the form

$$\hat{\mathbf{U}}^{k+1} = A^{-1}(F_1 - BP^k), \tag{5.80}$$

$$\begin{aligned} P^{k+1} &= P^k + w(G - SP^k) \\ &= P^k + w(C\hat{\mathbf{U}}^{k+1} - F_2 + DP^k), \end{aligned} \tag{5.81}$$

where w is some positive real number. The iteration starts with a given initial vector P^0 and is stopped when $\|P^{k+1} - P^k\|$ is small enough. The error equation for Richardson method is

$$P - P^{k+1} = (I - wS)(P - P^k), \tag{5.82}$$

and the convergence result is given below.

Theorem 5.2. *Suppose S is diagonalizable and that $\{\lambda_j\}$ are the eigenvalues of S. The Richardson iteration* (5.81) *converges if and only if $|1 - w\lambda_j| < 1$ for all eigenvalues λ_j. In addition, if all the eigenvalues of S are positive real numbers, then the solution will converge if*

$0 < w < 2/\lambda_{max}(S)$*. The optimal choice of* w *is*

$$w_{opt} = \frac{2}{\lambda_{min}(S) + \lambda_{max}(S)}, \tag{5.83}$$

and the corresponding convergence rate is

$$\|P - P^n\| \leq \left(\frac{\kappa(S) - 1}{\kappa(S) + 1} \right)^n \|P - P^0\|, \tag{5.84}$$

where $\kappa(S)$ *is the condition number of the matrix* S*.*

The proof of the theorem above can be found in most textbooks about numerical analysis, for example see Chapter 1 in [29]. In our problem, we do not know much information about the Schur complement S since its definition involves A^{-1}. Fortunately, for the simplified problem with $\mu^+ = \mu^-$, S is the same as what is obtained from the standard discretization using MAC scheme and is symmetric positive definite. We know that the correction terms used in (5.37)-(5.39) depend continuously on the jump ratio $\rho = \mu^-/\mu^+$ and the eigenvalues of S depend continuously on its entries. So, if the corrections terms can be regarded as small perturbations for the coefficient matrix, one can expect that the real part of λ_j is still positive for all eigenvalues of S. We summarize our conjecture as the following.

Conjecture 5.1. *The eigenvalues* $\{\lambda_j\}$ *of the Schur complement* S *(excluding the zero eigenvalue) defined in* (5.79) *are located in the right half plane, i.e.* $Re(\lambda_j) > 0$ *for all eigenvalues.*

Note that there is a zero eigenvalue for the Schur complement because the pressure is not unique and can differ by a constant. We will verify the above conjecture in the numerical tests. The rigorous theoretical proof is still an open question. Under this assumption, we have the following theorem.

Theorem 5.3. *Suppose* S *is diagonalizable and that* $\{\lambda_j\}$ *are the eigenvalues of* S *(excluding the zero eigenvalue) located in the right half plane, i.e.* $Re(\lambda_j) > 0$*. The Richardson iteration* (5.81) *converges if and only if*

$$0 < w < \frac{2Re(\lambda_j)}{|\lambda_j|^2} \quad \text{for all } \lambda_j. \tag{5.85}$$

This result simply comes from the condition $|1 - w\lambda_j| < 1$ in Theorem 5.2. In practice, the imaginary part of the eigenvalue of S has a small magnitude compared with its real part, and the equation (5.83) is still a solid choice for w. The largest eigenvalue $\lambda_{max}(S)$ can be approximated by running power iterations, but finding the smallest eigenvalue $\lambda_{min}(S)$ is expensive. A more practical choice for w is

$$w = \frac{2}{1+\lambda_{max}(S)}, \tag{5.86}$$

where 1 is the largest eigenvalue of S in the standard discretization using MAC scheme.

Notice that in each Uzawa iteration, the equation (5.80) involves solving the system $A^{-1}(F_1 - BP^k)$. This can be computed by Applying GMRES iterative method. Recall that $(-A)$ is the perturbed matrix of two-block Laplacian matrix. Hence the discrete Laplacian will be an ideal preconditioner and can be solved by fast Fourier transforms in $O(N\log(N))$ or multigrid methods in $O(N)$.

Remark 5.6. *The solver we designed in this section consists of an outer Uzawa iteration and inner GMRES iteration. In practice, one can set a small relative tolerance for GMRES method to avoid computational cost, which is similar to inexact Uzawa solvers [6].*

Remark 5.7. *The numerical solution of pressure P solved from Uzawa method differs from the exact solution by a constant due to the rank-one deficiency of the coefficient matrix.*

5.1.6 An outline of the algorithm

In this section, we give an outline of our algorithm.

Step 1: Generate a uniform mesh on the domain $\Omega = (a,b)\times(c,d)$ and use the level set function to represent the interface.

Step 2: Determine the irregular grid points using equation (5.17)-(5.18) and calculate the coordinates of the control points which are the intersection points of interface and grid lines.

Step 3: Use the standard MAC schemes to discretize momentum equations (5.7)-(5.8) and continuity equation (5.9) at the regular grid points.

Step 4: Solve the linear system (5.76) to determine the interpolation schemes for the correction terms $C^u_{i,j}$, $C^v_{i,j}$ and $C^p_{i,j}$ used in (5.37)-(5.39).

Step 5: Use the Uzawa method described in Section 5.1.5 to solve the system of linear equations (5.78) to get second order numerical solutions for velocity and pressure.

5.2 Numerical examples

We present a variety of numerical experiments to show the performance of the new direct IIM approach for solving Stokes equations on staggered grids. All the examples are computed with double precision and are performed on a laptop with Intel(R) Core(TM) i7-4650U CPU, 1.70GHz, 8.00 GB memory. We present errors in L^∞ norm in the following way,

$$E(\mathbf{u}) = \frac{1}{2}(\max_{i,j} |U^{i,j} - u(x_i, y_j)| + \max_{i,j} |V^{i,j} - v(x_i, y_j)|), \tag{5.87}$$

$$E(p) = \max_{i,j} \left[(P^{i,j} - \text{mean}(P)) - (p(x_i, y_j) - \text{mean}(p))\right], \tag{5.88}$$

where mean$(\cdot)$ represents the arithmetic mean of the values used on the given grid points. The convergence order r is estimated by (2.65). We use the Uzawa method to solve the discrete linear system with the choice of w defined in (5.86). In each Uzawa iteration, A^{-1} is solved by GMRES iterative method preconditioned by the two-block Laplacian matrix. The preconditioning step is completed by Algebraic Multigrid (AMG) solver. The absolute tolerance is set to be 10^{-6} for the Uzawa method while for GMRES solver, we set the relative tolerance as 10^{-1}. The initial value is $\mathbf{0}$ in all computations. In all tables listed in this section, we use ”N_U” and ”N_G” to represent the number of Uzawa and GMRES iterations respectively, ”N” the number of mesh intervals for the same type of grid points in each direction of the rectangular domain (half number of intervals in the big mesh defined in (5.11)), and ”λ_S”, ”λ_A” the smallest eigenvalue of the Schur complement S (excluding the zero eigenvalue) and matrix A respectively. In the following examples, a grid refinement analysis is performed to demonstrate the second order convergence for both velocity $\mathbf{u}$ and pressure p. We also conduct an eigenvalue analysis to show that the inverse of the Schur complement S and the inverse of the block A in the coefficient matrix is norm-bounded in L^2 norm and hence the method is stable.

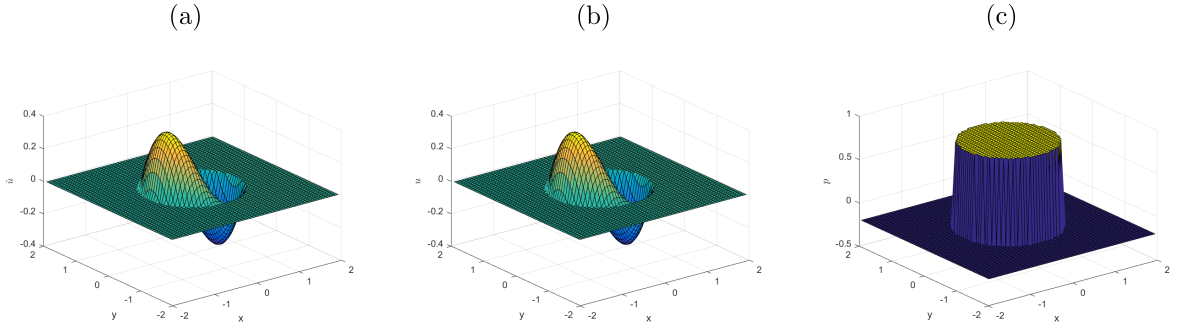

Figure 5.5: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.1 with jump ratio $\mu^- = 1$ and $\mu^+ = 0.5$ when $N = 64$ and $\Omega = [-2, 2] \times [-2, 2]$.

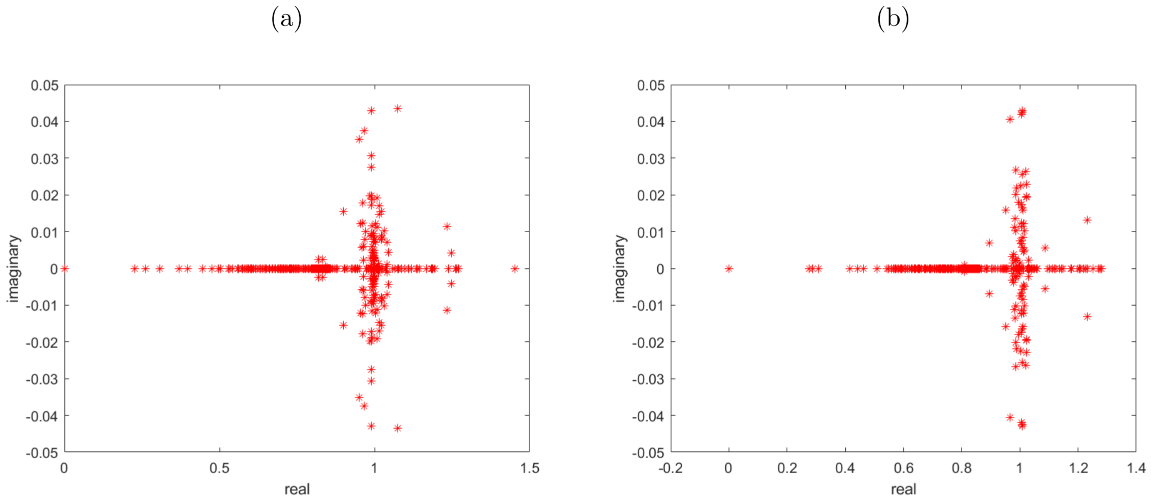

Figure 5.6: The eigenvalue plots of the Schur complement S in Example 5.1 when $N = 64$ and $\Omega = [-2, 2] \times [-2, 2]$. (a) $\mu^- = 1$ and $\mu^+ = 0.5$. (b): $\mu^- = 0.5$ and $\mu^+ = 1$.

Table 5.1: A grid refinement analysis for Example 5.1 with $\mu^- = 1$, $\mu^+ = 0.5$ and $\Omega = [-2, 2] \times [-2, 2]$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	8.87×10^{-3}		5.72×10^{-3}		27	8	0.25	0.75
64	2.12×10^{-3}	2.06	1.89×10^{-3}	1.60	33	12	0.23	0.75
128	5.49×10^{-4}	1.95	4.68×10^{-4}	2.01	37	18	0.21	0.76
256	1.46×10^{-4}	1.91	2.15×10^{-4}	1.12	41	22	0.21	0.76
512	3.30×10^{-5}	2.15	2.85×10^{-5}	2.92	45	32	0.21	0.76

Example 5.1. *Consider the example with the following exact solution:*

$$u(\mathbf{x}) = \begin{cases} y(x^2 + y^2 - 1) & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.89}$$

$$v(\mathbf{x}) = \begin{cases} -x(x^2 + y^2 - 1) & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.90}$$

$$p(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.91}$$

Table 5.2: A grid refinement analysis for Example 5.1 with $\mu^- = 0.5$, $\mu^+ = 1$ and $\Omega = [-2,2] \times [-2,2]$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	2.89×10^{-3}		3.33×10^{-3}		30	8	0.31	1.62
64	8.49×10^{-4}	1.77	1.51×10^{-3}	1.14	36	11	0.28	1.65
128	1.95×10^{-4}	2.12	4.71×10^{-4}	1.68	41	16	0.25	1.66
256	5.13×10^{-5}	1.93	1.08×10^{-4}	2.12	46	23	0.25	1.67
512	1.29×10^{-5}	1.99	2.83×10^{-5}	1.93	51	33	0.25	1.67

with viscosity

$$\mu(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \Omega^-, \\ \frac{1}{2} & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \textit{or} \quad \mu(\mathbf{x}) = \begin{cases} \frac{1}{2} & \textit{if } \mathbf{x} \in \Omega^-, \\ 1 & \textit{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.92}$$

The bounded external forcing term $\mathbf{g}$ *is given by*

$$g_1(\mathbf{x}) = \begin{cases} -8y & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \textit{and} \quad g_2(\mathbf{x}) = \begin{cases} 8x & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.93}$$

which has a finite jump across the interface. The interface is the unit circle which is represented by the zero level set of $\phi(x,y) = \sqrt{x^2+y^2} - 1$ *and* $\Omega = [-2,2] \times [-2,2]$*. The normal and tangential force densities can be computed by the analytical solutions and jump relations* (5.47) *and* (5.53)*.*

Table 5.3: A grid refinement analysis for Example 5.1 with $\mu^- = 1$, $\mu^+ = 0.5$ and $\Omega = [-1.99, 1.99] \times [-1.99, 1.99]$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	9.91×10^{-3}		5.12×10^{-3}		22	9	0.24	0.75
64	2.15×10^{-3}	2.20	2.71×10^{-3}	0.92	27	12	0.23	0.76
128	5.96×10^{-4}	1.85	4.00×10^{-4}	2.76	30	17	0.21	0.76
256	1.33×10^{-4}	2.16	1.90×10^{-4}	1.07	37	24	0.21	0.76
512	3.11×10^{-5}	2.10	2.49×10^{-5}	2.93	45	32	0.21	0.76

This is a simple example with a mild jump in the coefficient μ along the interface taken from [39]. We present a grid refinement analysis in Table 5.1-5.2. In the tables, the second

Table 5.4: A grid refinement analysis for Example 5.1 with $\mu^- = 0.5$, $\mu^+ = 1$ and $\Omega = [-1.99, 1.99] \times [-1.99, 1.99]$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	3.05×10^{-3}		3.27×10^{-3}		27	8	0.30	1.63
64	8.42×10^{-4}	1.86	1.36×10^{-3}	1.27	32	11	0.27	1.65
128	2.07×10^{-4}	2.02	4.14×10^{-4}	1.72	37	16	0.25	1.66
256	4.62×10^{-5}	2.16	1.07×10^{-4}	1.95	42	23	0.25	1.67
512	1.28×10^{-5}	1.85	2.80×10^{-5}	1.93	48	34	0.25	1.67

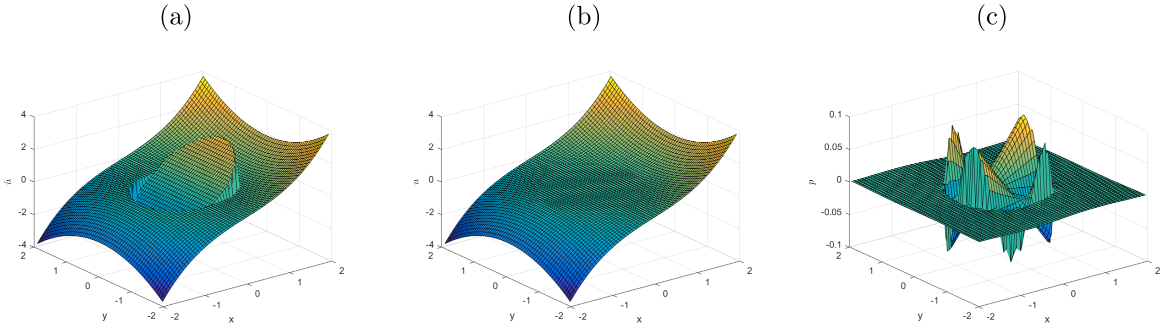

Figure 5.7: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.2 with jump ratio $\mu^- = 10$ and $\mu^+ = 1$ when $N = 64$.

column is the maximum error of the velocity $\mathbf{u}$ while the third column is the approximate convergence order. The fourth column is the error of the pressure p and the fifth column is the corresponding approximate convergence order. The sixth and seventh columns represent the number of Uzawa and GMRES iterations respectively. The last two columns are the smallest eigenvalue of the Schur complement S and that of the block matrix A in the coefficient matrix. Compared with the previous augmented method in [39], we can observe that both methods achieved average second order accuracy in the velocity and the pressure. However, unlike the augmented approach, the main advantage of our direct IIM method is that we do not need to set up the augmented variable and the additional Schur complement equation, which makes the implementation much easier. The eigenvalue plots of the Schur complement S are given in Figure 5.6. One can observe that all the eigenvalues are located in the right half plane and the smallest one is bounded away as we increase the mesh size, which agrees with our conjecture 5.1. Combined with the fact that the smallest eigenvalue of the block matrix A is also bounded away from 0, one can ensure that both the Uzawa and GMRES methods will converge to the exact solutions and the method is stable at least in L_2 norm.

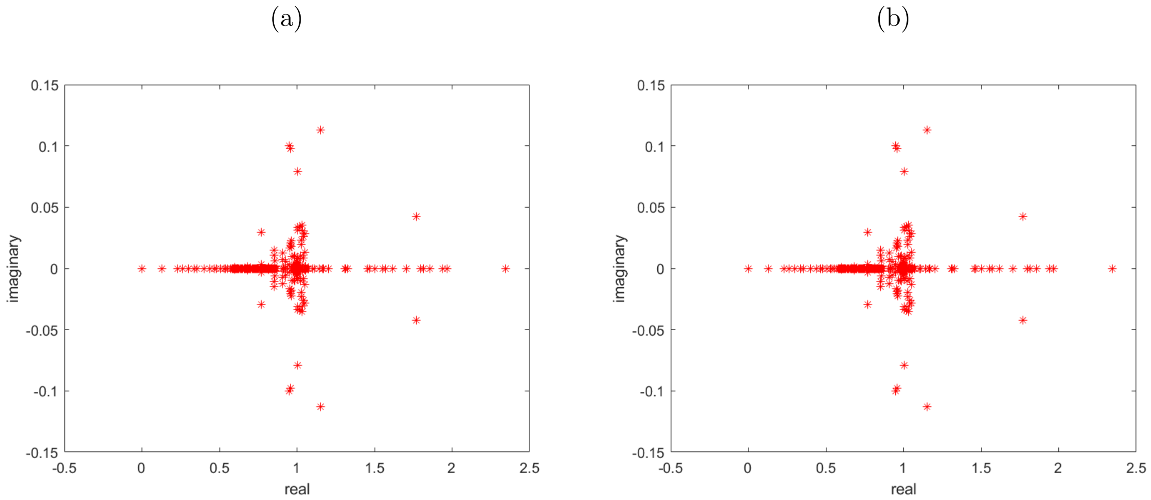

Figure 5.8: The eigenvalue plots of the Schur complement S in Example 5.2 when $N = 64$. (a) $\mu^- = 10$ and $\mu^+ = 1$. (b): $\mu^- = 1$ and $\mu^+ = 10$.

Table 5.5: A grid refinement analysis for Example 5.2 with $\mu^- = 10$ and $\mu^+ = 1$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	1.41×10^{-2}		1.70×10^{-2}		31	14	0.14	0.22
64	3.26×10^{-3}	2.11	5.22×10^{-3}	1.70	34	22	0.13	0.21
128	8.39×10^{-4}	1.96	1.49×10^{-3}	1.81	30	33	0.13	0.21
256	2.26×10^{-4}	1.89	5.34×10^{-4}	1.48	34	43	0.13	0.21
512	5.11×10^{-5}	2.14	1.10×10^{-4}	2.28	35	65	0.13	0.21

Notice that in Table 5.1-5.2, the convergence rate for the pressure oscillates around number two. This phenomenon has been observed and explained in the literature, see for example [45]. This is because the global errors obtained from the IIM depend on the relative location of the underlying grid and the interface. This is can be seen from the Taylor expansion that the largest error at a grid point behaves like $O((\mathbf{x}_{ij} - \mathbf{X}^*_{ij})^2)$ instead of $O(h^2)$, where $\mathbf{x}_{ij}$ is the grid point that has the largest error, and $\mathbf{X}^*_{ij}$ is the corresponding intersection point on the interface near $\mathbf{x}_{ij}$. We also carried out a further study on the convergence orders by perturbing the domain (or equally the mesh) for the same problem. We perturb the mesh slightly by choosing $\Omega = [-1.99, 1.99] \times [-1.99, 1.99]$. The results of the grid refinement analysis are listed in Table 5.3-5.4. The results again confirmed the explanations above that the errors do not decline monotonically but the average convergence rates of the velocity and pressure remain to be second order accurate.

Table 5.6: A grid refinement analysis for Example 5.2 with $\mu^- = 1$ and $\mu^+ = 10$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	3.18×10^{-3}		6.82×10^{-2}		10	20	0.26	1.80
64	8.19×10^{-4}	1.96	2.35×10^{-2}	1.54	25	15	0.24	1.84
128	1.63×10^{-4}	2.07	4.48×10^{-3}	2.39	31	22	0.23	1.89
256	3.75×10^{-5}	1.93	1.19×10^{-3}	1.91	33	35	0.23	1.89
512	1.04×10^{-5}	1.85	3.31×10^{-4}	1.85	34	53	0.23	1.89

Example 5.2. *In the previous example, the force density is a constant. In this example, we construct the exact solutions in such a way that all the jumps and their derivatives along the interface are nonconstant functions. The exact velocity and pressure are given by*

$$u(\mathbf{x}) = \begin{cases} \dfrac{y}{4} & \textit{if } \mathbf{x} \in \Omega^-, \\[2ex] \dfrac{y}{4}(x^2+y^2) & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.94}$$

$$v(\mathbf{x}) = \begin{cases} -\dfrac{x}{4}(1-x^2) & \textit{if } \mathbf{x} \in \Omega^-, \\[2ex] -\dfrac{xy^2}{4} & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.95}$$

$$p(\mathbf{x}) = \begin{cases} \left(-\dfrac{3}{4}x^3 + \dfrac{3}{8}x\right) y & \textit{if } \mathbf{x} \in \Omega^-, \\[2ex] 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.96}$$

with viscosity

$$\mu(\mathbf{x}) = \begin{cases} 10 & \textit{if } \mathbf{x} \in \Omega^-, \\ 1 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \textit{or} \quad \mu(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \Omega^-, \\ 10 & \textit{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.97}$$

The bounded external forcing term $\mathbf{g}$ *is given by*

$$g_1(\mathbf{x}) = \begin{cases} \left(-\frac{9}{4}x^2 + \frac{3}{8}\right) y & \textit{if } \mathbf{x} \in \Omega^-, \\ -2\mu^+ y & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \textit{and} \quad g_2(\mathbf{x}) = \begin{cases} -\frac{3}{4}x^3 + \frac{3}{8}x - \frac{3\mu^-}{2}x & \textit{if } \mathbf{x} \in \Omega^-, \\ \frac{\mu^+}{2}x & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.98}$$

which is discontinuous across the interface. The interface is still the unit circle and $\Omega = [-2,2] \times [-2,2]$. *The normal and tangential force densities can be computed by the analytical solutions and jump relations* (5.47) *and* (5.53).

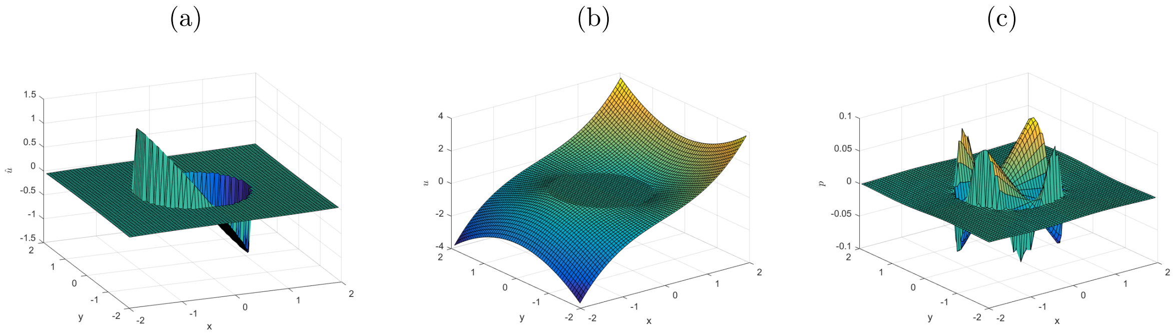

Figure 5.9: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.2 with jump ratio $\mu^- = 10$ and $\mu^+ = 0.01$ when $N = 64$.

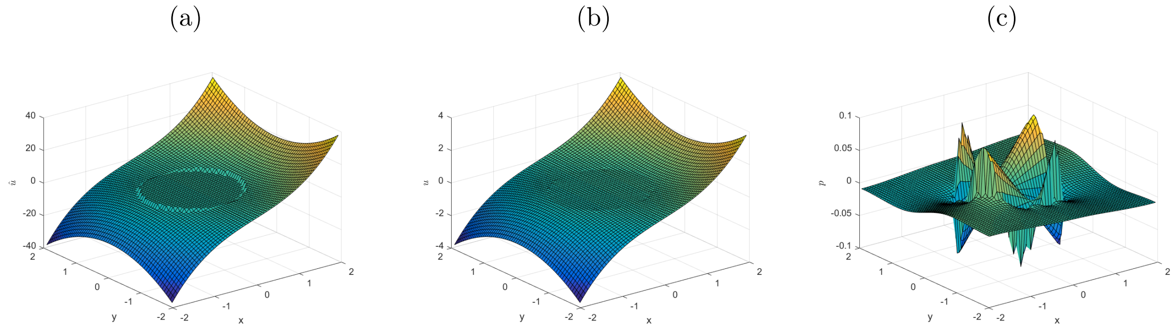

Figure 5.10: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.2 with jump ratio $\mu^- = 0.01$ and $\mu^+ = 10$ when $N = 64$.

This is a general example with a medium jump in the viscosity along the unit circle. The results for the grid refinement analysis are given in Table 5.5-5.6. The plots of the solutions and the distribution of the eigenvalues of S are shown in Figure 5.7 and Figure 5.8. Again, we observe second order accurate solutions for both the velocity and pressure. The eigenvalues of the Schur complement and block matrix A are located in the right half plane and their smallest one are bounded away from zero, which implies the stability of our method.

Now we use the same exact solution and interface but with a large jump in the coefficient μ along the interface.

$$\mu(\mathbf{x}) = \begin{cases} 10 & \text{if } \mathbf{x} \in \Omega^-, \\ 0.01 & \text{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \text{or} \quad \mu(\mathbf{x}) = \begin{cases} 0.01 & \text{if } \mathbf{x} \in \Omega^-, \\ 10 & \text{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.99}$$

Notice that the coefficient μ has a large jump size along the interface. Nevertheless, all the nice

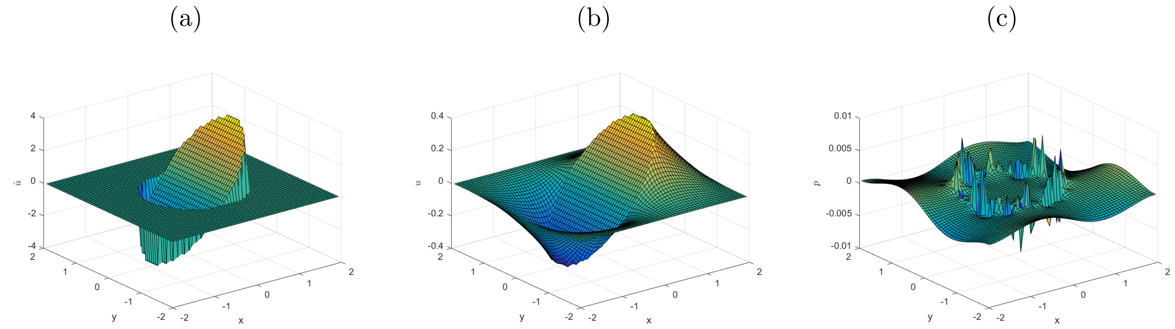

Figure 5.11: The error plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.2 with jump ratio $\mu^- = 10$ and $\mu^+ = 0.01$ when $N = 64$.

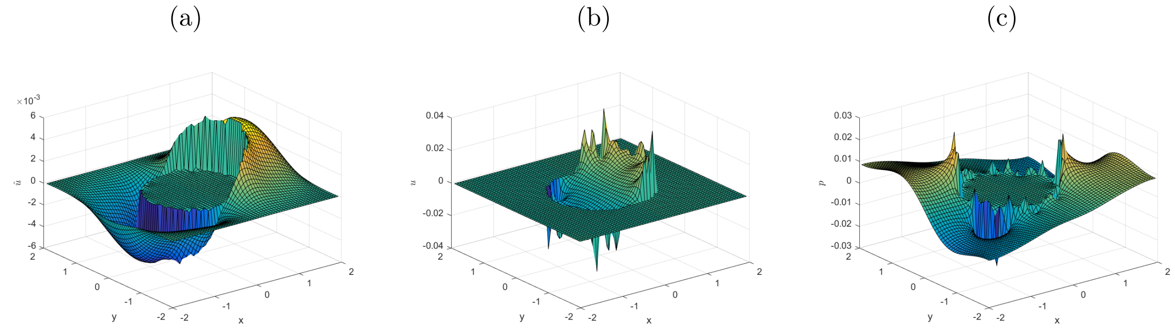

Figure 5.12: The error plots of the transformed x-component velocity $\hat{u}$, the $x$component velocity u and the pressure p for Example 5.2 with jump ratio $\mu^- = 0.01$ and $\mu^+ = 10$ when $N = 64$.

features are the same as in the previous examples. The results of the grid refinement analysis are given in Table 5.7-5.8 and the plots for the solution, error and eigenvalue distribution are given in Figure 5.9-5.10, 5.11-5.12 and 5.13. We do observe that the smallest eigenvalue of A is close to zero (but bounded away from it) when $\mu^- = 10$ and $\mu^+ = 0.01$, which causes difficulty in solving the system by GMRES method. From the solution plots, we also observe that the transformed velocity $\hat{u}$ has a relatively large jump across the interface for the case with $\rho = 1000$ compared with the case where $\rho = 0.001$. As a result, the magnitude of the error for the transformed velocity is relatively larger when $\rho = 1000$ compared with that in the second case. This explains why it takes more computational efforts to solve the problem and needs a finer grid to resolve the pressure for the first case.

Example 5.3. *In this example, the interface is an elliptic shape, which is represented by the*

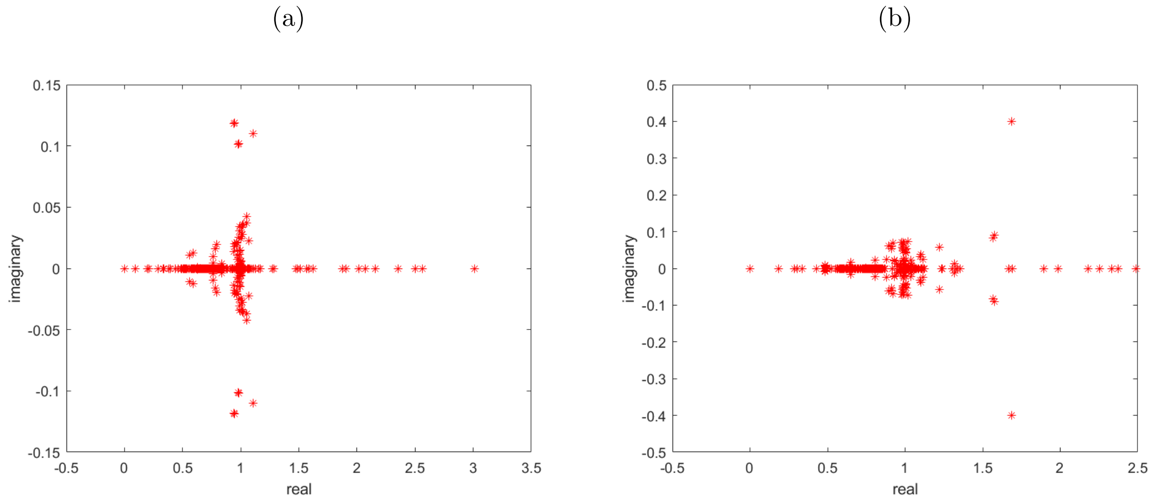

Figure 5.13: The eigenvalue plots of the Schur complement S in Example 5.2 when $N = 64$. (a) $\mu^- = 10$ and $\mu^+ = 0.01$. (b): $\mu^- = 0.01$ and $\mu^+ = 10$.

Table 5.7: A grid refinement analysis for Example 5.2 with $\mu^- = 10$ and $\mu^+ = 0.01$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	1.56×10^{-0}		1.68×10^{-2}		42	22	0.10	0.003
64	3.86×10^{-1}	2.01	6.71×10^{-3}	1.32	40	36	0.09	0.003
128	9.82×10^{-2}	1.97	2.07×10^{-3}	1.70	21	55	0.09	0.003
256	2.64×10^{-2}	1.90	6.39×10^{-4}	1.70	26	74	0.09	0.003
512	6.09×10^{-3}	2.12	1.48×10^{-4}	2.11	33	101	0.09	0.003

zero level set function

$$\phi(x, y) = \sqrt{\frac{x^2}{1^2} + \frac{y^2}{0.5^2}} - 1. \tag{5.100}$$

Table 5.8: A grid refinement analysis for Example 5.2 with $\mu^- = 0.01$ and $\mu^+ = 10$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	1.62×10^{-1}		8.19×10^{-2}		22	10	0.19	1.67
64	3.52×10^{-2}	2.20	2.64×10^{-2}	1.63	32	15	0.18	1.71
128	3.39×10^{-3}	3.38	5.22×10^{-3}	2.34	34	25	0.18	1.75
256	5.91×10^{-4}	2.52	1.51×10^{-3}	1.79	47	35	0.18	1.75
512	7.74×10^{-5}	2.93	3.56×10^{-4}	2.08	51	52	0.18	1.75

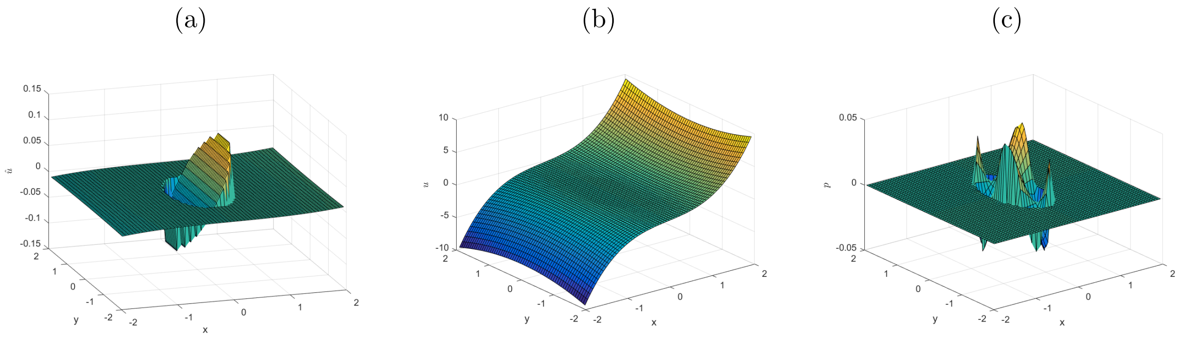

Figure 5.14: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.3 with jump ratio $\mu^- = 1$ and $\mu^+ = 0.001$ when N = 64.

The exact velocity and pressure are given by

$$u(\mathbf{x}) = \begin{cases} \dfrac{y}{4} & \textit{if } \mathbf{x} \in \Omega^-, \\ \dfrac{y}{4}(x^2 + 4y^2) & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.101}$$

$$v(\mathbf{x}) = \begin{cases} -\dfrac{x}{16}(1.0 - x^2) & \textit{if } \mathbf{x} \in \Omega^-, \\ -\dfrac{xy^2}{4} & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.102}$$

$$p(\mathbf{x}) = \begin{cases} \left(-\dfrac{3}{4}x^3 + \dfrac{3}{8}x\right) y & \textit{if } \mathbf{x} \in \Omega^-, \\ 0 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \tag{5.103}$$

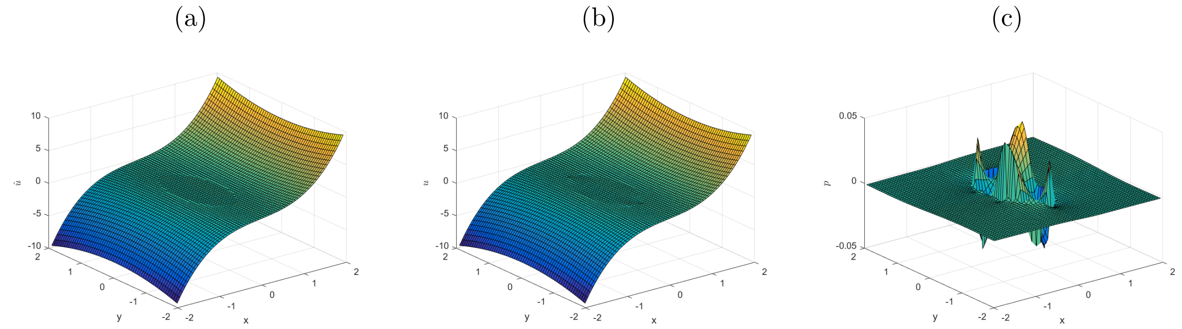

Figure 5.15: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.3 with jump ratio $\mu^- = 0.001$ and $\mu^+ = 1$ when N = 64.

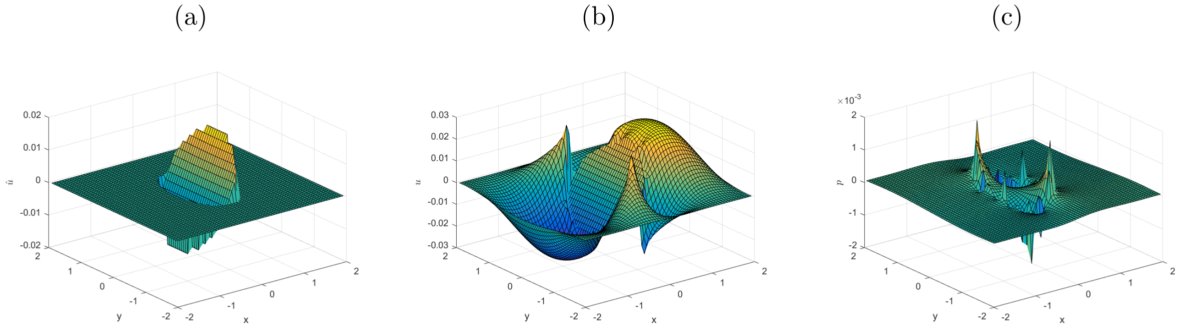

Figure 5.16: The error plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.3 with jump ratio $\mu^- = 1$ and $\mu^+ = 0.001$ when N = 64.

with viscosity

$$\mu(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \Omega^-, \\ 0.001 & \textit{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \textit{or} \quad \mu(\mathbf{x}) = \begin{cases} 0.001 & \textit{if } \mathbf{x} \in \Omega^-, \\ 1 & \textit{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.104}$$

The bounded external forcing term $\mathbf{g}$ *is given by equation* (5.98) *and* $\Omega = [-2, 2] \times [-2, 2]$. *The normal and tangential force densities can be computed by the analytical solutions and jump relations* (5.47) *and* (5.53).

This is a general example with a large jump in the viscosity along an ellipse. Once again, when the mesh get sufficiently fine, we observe second order convergence for the velocity $\mathbf{u}$. However, the pressure seems only sup-linear for the case where $\rho = 1000$ and one needs a finer mesh to resolve it, see Table 5.9. This is in part due to the large jump size in the transformed velocity $\hat{u}$ along the interface as shown in Figure 5.14. In contrast, the transformed velocity $\hat{u}$

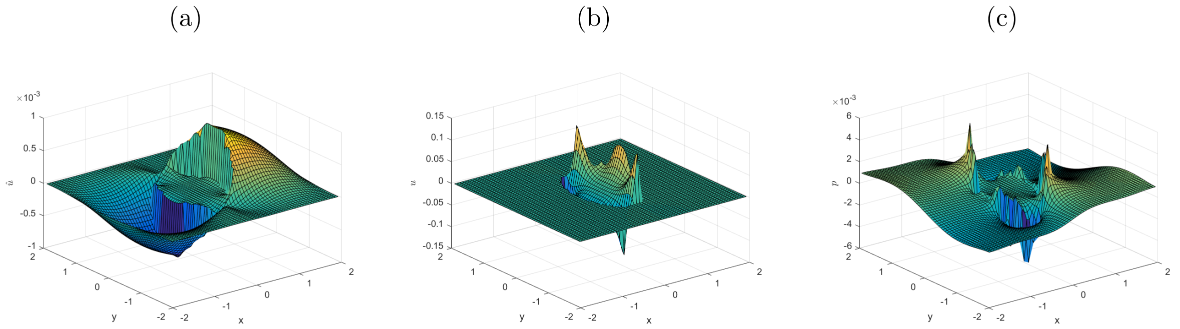

Figure 5.17: The error plots o thef transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.3 with jump ratio $\mu^- = 0.001$ and $\mu^+ = 1$ when $N = 64$.

Table 5.9: A grid refinement analysis for Example 5.3 with $\mu^- = 1$ and $\mu^+ = 0.001$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	2.15×10^{-1}		4.12×10^{-3}		32	30	0.20	0.003
64	4.16×10^{-2}	2.37	1.81×10^{-3}	1.19	35	38	0.18	0.003
128	9.67×10^{-3}	2.11	6.37×10^{-4}	1.51	53	58	0.17	0.003
256	2.18×10^{-3}	2.15	1.97×10^{-4}	1.69	60	88	0.17	0.003
512	5.28×10^{-4}	2.05	5.77×10^{-5}	1.77	65	105	0.17	0.003

has a relatively small jump across the interface as what is shown in Figure 5.15, and hence can be easily resolved. This is also reflected in the error plots for the transformed velocity in Figure 5.16-5.17, where the magnitude of the error is relatively larger when $\rho = 1000$ compared with that in the case where $\rho = 0.001$. One can actually observe super quadratic convergence in $\mathbf{u}$ from the Table 5.10. The distributions of eigenvalues are given in Figure 5.18. The method is stable since the smallest eigenvalues λ_S and λ_A are bounded away from zero.

Example 5.4. *In this example, the interface is a complicated star-shaped interface see Figure 5.20a, which is described in the polar coordinates as*

$$r(\theta) = 0.5 + 0.15\sin(6\theta), \qquad 0 \le \theta < 2\pi. \tag{5.105}$$

We assume that the singular force is the surface tension and the force strength $\mathbf{f}(s)$ *is now given by*

$$\mathbf{f}(s) = \gamma \frac{\partial^2}{\partial s^2}\mathbf{X}(s), \tag{5.106}$$

where γ *is the coefficient of the surface tension between the two fluids and* s *is the arc length.*

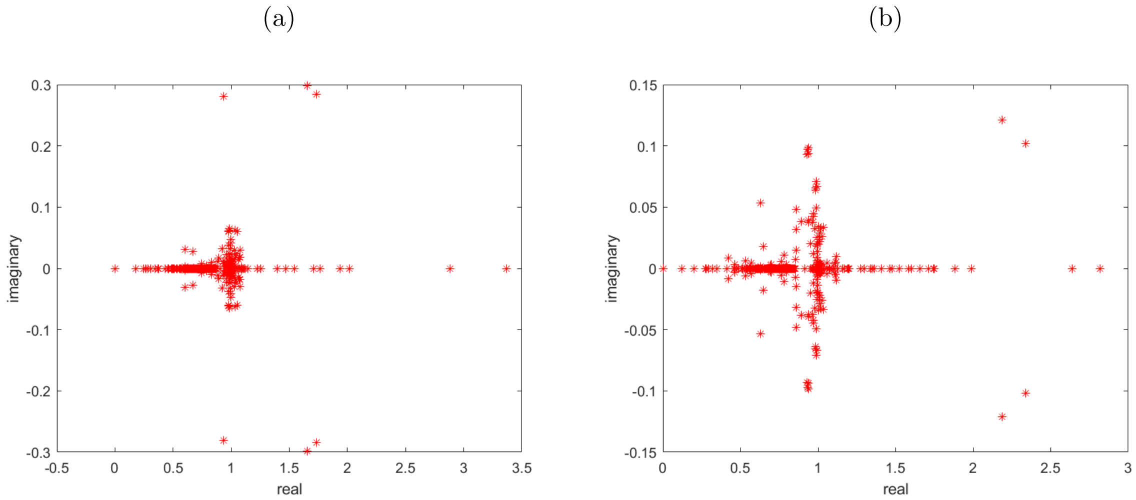

Figure 5.18: The eigenvalue plots of the Schur complement S in Example 5.3 when $N = 64$. (a): $\mu^- = 1$ and $\mu^+ = 0.001$. (b): $\mu^- = 0.001$ and $\mu^+ = 1$.

Table 5.10: A grid refinement analysis for Example 5.3 with $\mu^- = 0.001$ and $\mu^+ = 1$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
32	5.26×10^{-1}		2.19×10^{-2}		32	10	0.14	1.29
64	9.40×10^{-2}	2.48	5.19×10^{-3}	2.08	35	15	0.12	1.28
128	1.34×10^{-2}	2.81	1.39×10^{-3}	1.90	59	26	0.12	1.22
256	2.19×10^{-3}	2.61	5.09×10^{-4}	1.45	62	31	0.12	1.22
512	3.04×10^{-4}	2.85	8.77×10^{-5}	2.54	65	48	0.12	1.22

The vector $\partial^2\mathbf{X}/\partial s^2$ is normal to the interface with magnitude equal to the curvature. Assume $\gamma = 0.1$ and the viscosity is

$$\mu(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega^-, \\ 0.5 & \text{if } \mathbf{x} \in \Omega^+, \end{cases} \quad \text{or} \quad \mu(\mathbf{x}) = \begin{cases} 0.5 & \text{if } \mathbf{x} \in \Omega^-, \\ 1 & \text{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.107}$$

The bounded external forcing term is set to be $\mathbf{g} = \mathbf{0}$ and the homogeneous Dirichlet boundary condition is assumed with the domain $\Omega = [-1, 1] \times [-1, 1]$.

This is a general example with a complicated interface. Since the analytic solution is not available, the error in velocity and pressure are measured by using a reference solution which is obtained on a fine 1024 × 1024 grid. The results of the grid refinement analysis are given in Table 5.11. Due to the large curvature of the interface, it does require fine grids to get second order convergence rate. The solution plots are provided in Figure 5.19.

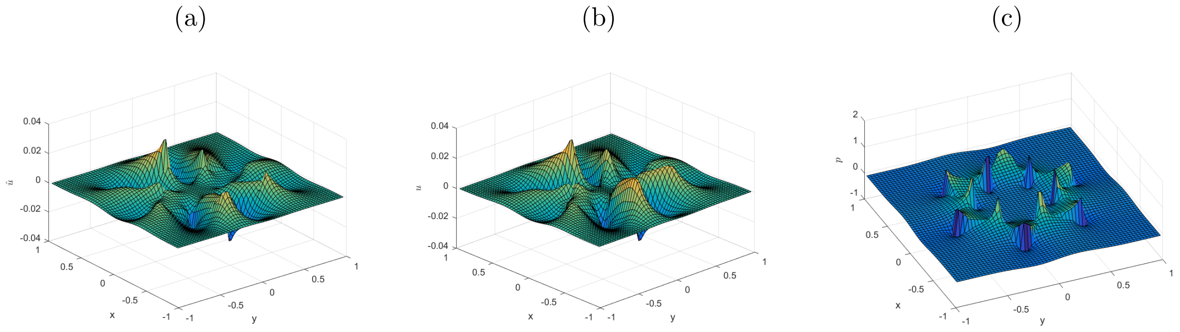

Figure 5.19: The solution plots of the transformed x-component velocity $\hat{u}$, the x-component velocity u and the pressure p for Example 5.4 with jump ratio $\mu^- = 0.5$ and $\mu^+ = 1$ when $N = 64$.

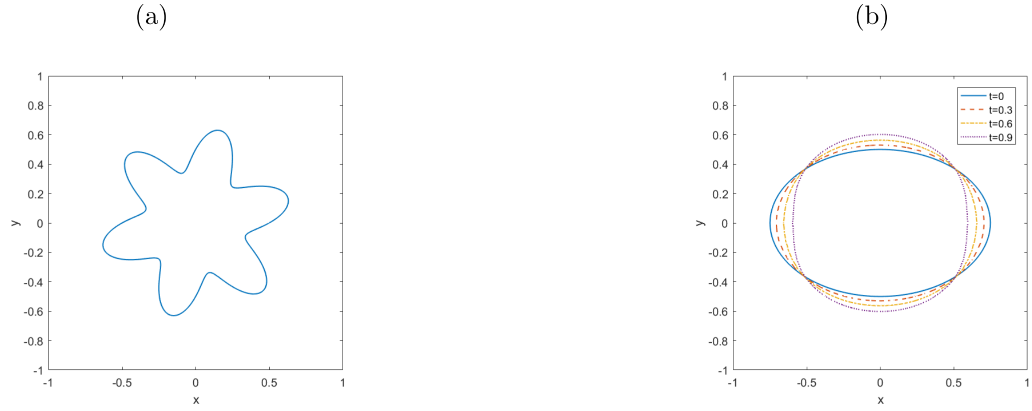

Figure 5.20: (a): A complicated star-shaped interface used in Example 5.4. (b): The configurations of the interface at different times in Example 5.5 when $N = 32$.

Table 5.11: A grid refinement analysis for Example 5.4 with $\mu^- = 1$, $\mu^+ = 0.1$ or $\mu^- = 0.5$, $\mu^+ = 1$.

N	$E(\mathbf{u})$	r	$E(p)$	r	N_U	N_G	λ_S	λ_A
$\mu^- = 1$, $\mu^+ = 0.5$								
64	6.50×10^{-3}		3.33×10^{-1}		25	14	0.23	3.09
128	1.40×10^{-3}	2.22	1.13×10^{-1}	1.56	31	20	0.22	3.14
256	8.27×10^{-4}	0.76	7.08×10^{-2}	0.67	39	27	0.22	3.14
512	2.06×10^{-4}	2.01	1.91×10^{-2}	1.89	44	39	0.22	3.14
$\mu^- = 0.5$, $\mu^+ = 1$								
64	1.07×10^{-2}		3.49×10^{-1}		25	10	0.28	6.49
128	5.80×10^{-3}	0.88	2.59×10^{-1}	0.43	30	16	0.25	6.60
256	7.35×10^{-4}	2.98	5.98×10^{-2}	2.11	39	23	0.25	6.60
512	1.91×10^{-4}	1.94	1.64×10^{-2}	1.87	43	34	0.25	6.60

Example 5.5. *(Ellipse-shaped bubble under Stokes flow). The purpose of this example is to show that the present method can handle moving interface problem. The interface is an elliptic shape, which is represented by the zero level set function*

$$\phi(x, y) = \sqrt{\frac{x^2}{0.75^2} + \frac{y^2}{0.5^2}} - 1. \tag{5.108}$$

Similar to the above example, we assume the singular force is provided by the surface tension and the force strength $\mathbf{f}(s, t)$ *is now time-dependent and given by*

$$\mathbf{f}(s, t) = \gamma \frac{\partial^2}{\partial s^2} \mathbf{X}(s, t). \tag{5.109}$$

Assume $\gamma = 1$ *and viscosity is*

$$\mu(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \Omega^-, \\ 0.1 & \textit{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{5.110}$$

The bounded external forcing term is set to be $\mathbf{g} = \mathbf{0}$ *and the homogeneous Dirichlet boundary condition is imposed with the domain* $\Omega = [-1, 1] \times [-1, 1]$.

This is a moving interface example. The evolution of the level set function is governed by the Hamilton-Jacobi equation

$$\phi_t + \Delta\phi \cdot \mathbf{u} = 0. \tag{5.111}$$

We solve this equation by the ENO (essential nonoscillating) scheme described in [62]. Since we assume zero gravity, the surface tension will bring the bubble back to a circular shape. In our simulation, the interface becomes a circle around $t = 0.9$. The configurations of the interface at different times are plotted in Figure 5.20b.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we developed two different types of immersed interface methods, augmented IIM and direct IIM, to solve interface problems. The major contributions are the following:

- We proposed a new augmented immersed interface method for general elliptic interface problems with variable coefficients that have finite jumps across a general interface, and non-homogeneous jump conditions. Both the computed solution and its gradient from each side of the interface are second order globally. The convergence of the method has been proved both in one and two dimensions under appropriate regularity assumptions and a piecewise constant $\beta(\mathbf{x})$. For a variable coefficient $\beta(\mathbf{x})$, the conclusions are still true if h is small enough, that is, in the asymptotic sense.

- We proposed a new direct immersed interface method for general elliptic interface problems with a variable coefficient that have finite jumps across a general interface with non-homogeneous jump conditions. Both the computed solution and its gradient from each side of the interface are second order globally. The notable feature of this method is that no augmented variable is needed. Thus, the method is easier and computationally efficient. Our numerical experiments indicated that the condition number of the system of finite difference equations is independent of the coefficient and the jump conditions. The second order convergence of the method is theoretically proved in one-dimensional case by using Green's function. In two-dimensional cases, the second order convergence of algorithm using the optimization is proved by enforcing the discrete elliptic maximum principle. For the original algorithm without using the optimization, its second order convergence is demonstrated in numerical tests.

- We proposed two new ADI methods using augmented variables to separate different scales. The two ADI methods have the same structure and advantages of the classical ADI method. Each directional sweep involves solving a sequence of tridiagonal system of equations. If $[\beta u]$ is used as the augmented variable, then the finite difference equations in terms of the coefficients are almost the same in terms of βu as the classical ADI method with added correction terms. The time step constraint is $\Delta t \leq Ch/\beta_{max}$ which is not considered as a severe restriction. If $\frac{d[\beta u]}{d\tau}$ is chosen as the augmented variable, then the finite difference coefficients will be modified with closed form of expressions and the method is asymptotically unconditionally stable.

- We proposed a new direct immersed interface method for solving two-phase incompressible Stokes equations with an interface and a piecewise constant viscosity. The computed numerical solutions are second order accurate for both velocity and pressure. The eigenvalue analysis in our numerical experiments indicate that our method is stable and the solutions computed from both Uzawa and GMRES iterative method converge to exact solutions.

6.2 Future work

- Our proposed ADI methods are originally designed for solving two-dimensional models. Using $[\beta u]$ as the augmented variable, this approach can also be extended to solve three-dimensional problems and we expect to get similar results.

- Our proposed ADI methods perform well for the problems with piecewise constant coefficients and continuous solution. But for more general problems, it will be better to use the Crank-Nicolson method. We plan to design a direct IIM using the Crank-Nicolson scheme to solve general parabolic interface problems.

- We also plan to extend our direct IIM based on MAC scheme to solve incompressible two-phase Navier-Stokes equations.

REFERENCES

[1] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5:207–213, 1970.

[2] J. T. Beale and A. T. Layton. On the accuracy of finite difference methods for elliptic problems with interfaces. *Commun. Appl. Math. Comput. Sci.*, 1:91–119, 2006.

[3] Jacob Bedrossian, James H Von Brecht, Siwei Zhu, Eftychios Sifakis, and Joseph M Teran. A second order virtual node method for elliptic problems with interfaces and irregular domains. *Journal of Computational Physics*, 229(18):6405–6426, 2010.

[4] François Bouchon and Gunther H Peichl. The immersed interface technique for parabolic problems with mixed boundary conditions. *SIAM Journal on Numerical Analysis*, 48(6):2247–2266, 2010.

[5] J. Bramble and J. King. A finite element method for interface problems in domains with smooth boundaries and interfaces. *Advances in Comput. Math.*, 6:109–138, 1996.

[6] James H Bramble, Joseph E Pasciak, and Apostol T Vassilev. Analysis of the inexact uzawa algorithm for saddle point problems. *SIAM Journal on Numerical Analysis*, 34(3):1072–1092, 1997.

[7] B. Camp, T. Lin, Y. Lin, and W. Sun. Quadratic immersed finite element spaces and their approximation capabilities. *Adv. Comput. Math.*, 24:81–112, 2006.

[8] John R. Cannon and Mario Primicerio. A two phase Stefan problem with flux boundary conditions. *Ann. Mat. Pura Appl. (4)*, 88:193–205, 1971.

[9] S. Chen, B. Merriman, P.Smereka, and S. Osher. A fast level set based algorithm for Stefan problems. 135:8–29, 1997.

[10] Xiaohong Chen, Xiufang Feng, and Zhilin Li. A direct method for accurate solution and gradient computations for elliptic interface problems. *Numerical Algorithms*, pages 1–32, 2018.

[11] Xiaohong Chen, Zhilin Li, and Juan Ruiz Álvarez. A direct iim approach for two-phase stokes equations with discontinuous viscosity on staggered grids. *Computers & Fluids*, 2018.

[12] Z. Chen and J. Zou. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numer. Math.*, 79:175–202, 1998.

[13] I-Liang Chern and Yu-Chen Shu. A coupling interface method for elliptic interface problems. *Journal of Computational Physics*, 225(2):2138–2174, 2007.

[14] S. Chou, D. Kwak, and K. Wee. Optimal convergence analysis of an immersed interface finite element method. *Advances in Computational Mathematics*, 33:149–168, 2010.

[15] X. Feng, Z. Li, and L. Wang. Analysis and numerical methods for some crack problems. *Int. J. Num. Anal. & Model., Series B*, 2:155–166, 2011.

[16] R. M. Furzeland. A comparative study of numerical methods for moving boundary problems. *J. Inst. Maths Applics*, 26:411–429, 1980.

[17] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order.* Classics in Mathematics. Springer-Verlag, Berlin, 2001. Reprint of the 1998 edition.

[18] Yan Gong, Bo Li, and Zhilin Li. Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions. *SIAM Journal on Numerical Analysis*, 46(1):472–495, 2008.

[19] W. Hackbusch. *Elliptic Differential Equations: Theory and Numerical Treatment.* Springer-Verlag, 1992.

[20] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comput. Methods Appl. Mech. Engrg.*, 191:5537–5552, 2002.

[21] X. He, T. Lin, and Y. Lin. Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *Int. J. Numer. Anal. Model.*, 2010.

[22] Jeffrey Lee Hellrung, Luming Wang, Eftychios Sifakis, and Joseph M Teran. A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions. *Journal of Computational Physics*, 231(4):2015–2048, 2012.

[23] S. Hou, Z. Li, L. Wang, and W. Wang. A numerical method for solving elasticity equations with sharp-edged interfaces. *Commun. Comput. Phys.*, 12:595–612, 2012.

[24] S. Hou and X. Liu. A numerical method for solving variable coefficient elliptic equation with interfaces. 202:411–445, 2005.

[25] T. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. 134:236–252, 1997.

[26] T. Hou, X. Wu, and Y. Zhang. Removing the cell resonance error in the multiscale finite element method via a Petrov-Galerkin formulation. *Comm. Math. Sci.*, 2:185–205, 2004.

[27] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Removing the stiffness from interfacial flows with surface tension. 114:312–338, 1994.

[28] Juri D Kandilarov and Lubin G Vulkov. The immersed interface method for two-dimensional heat-diffusion equations with singular own sources. *Applied numerical mathematics*, 57(5-7):486–497, 2007.

[29] T Kelley, C. Iterative methods for linear and nonlinear equations. *Raleigh NC: North Carolina State University*, 1995.

[30] Do Y Kwak, Kye T Wee, and Kwang S Chang. An analysis of a broken p_1-nonconforming finite element method for interface problems. *SIAM Journal on Numerical Analysis*, 48(6):2117–2134, 2010.

[31] Anita T Layton. An efficient numerical method for the two-fluid stokes equations with a moving immersed boundary. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2147–2155, 2008.

[32] William J. Layton, Friedhelm Schieweck, and Ivan Yotov. Coupling fluid flow with porous media flow. *SIAM J. Numer. Anal*, 40:2003, 2003.

[33] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations, Steady State and Time Dependent Problems.* SIAM, 2007.

[34] 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.

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

[36] Z. Li. The immersed interface method using a finite element formulation. *Applied Numer. Math.*, 27:253–267, 1998.

[37] Z. Li. On convergence of the immersed boundary method for elliptic interface problems. *Math. Comp.*, 84(293):1169–1188, 2015.

[38] Z. Li and K. Ito. *The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains.* SIAM Frontier Series in Applied mathematics, FR33, 2006.

[39] Z. Li, K. Ito, and M-C. Lai. An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. *Computers and Fluids*, 36:622–635, 2007.

[40] Z. Li, T. Lin, and X. Wu. New Cartesian grid methods for interface problem using finite element formulation. *Numer. Math.*, 96:61–98, 2003.

[41] Z. Li and A. Mayo. ADI methods for heat equations with discontinuities along an arbitrary interface. In *Proc. Symp. Appl. Math.* W. Gautschi, editor, volume 48, pages 311–315. AMS, 1993.

[42] Z. Li and B. Soni. Fast and accurate numerical approaches for Stefan problems and crystal growth. *Numerical Heat Transfer, B: Fundamentals*, 35:461–484, 1999.

[43] Z. Li, L. Xi, Q. Cai, H. Zhao, and R. Luo. A semi-implicit augmented IIM for Navier-Stokes equations with open and traction boundary conditions. 297:182–193, 2015.

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

[45] Zhilin Li and Kazufumi Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM Journal on Scientific Computing*, 23(1):339–361, 2001.

[46] Zhilin Li, Haifeng Ji, and Xiaohong Chen. Accurate solution and gradient computation for elliptic interface problems with variable coefficients. *SIAM Journal on Numerical Analysis*, 55(2):570–597, 2017.

[47] Zhilin Li and Yun-Qiu Shen. A numerical method for solving heat equations involving interfaces. In *Electronic Journal of Differential Equations, Conf*, volume 3, 1999.

[48] T. Lin and X. Zhang. Linear and bilinear immersed finite elements for planar elasticity interface problems. *J. Comput. Appl. Math.*, 236:4681–4699, 2012.

[49] Tao Lin, Yanping Lin, and Xu Zhang. Partially penalized immersed finite element methods for elliptic interface problems. *SIAM J. Numer. Anal.*, 53(2):1121–1144, 2015.

[50] J. Liu. Open and traction boundary conditions for the incompressible Navier-Stokes equations. 228:7250–7267, 2009.

[51] Jiankang Liu and Zhoushun Zheng. Iim-based adi finite difference scheme for nonlinear convection–diffusion equations with interfaces. *Applied Mathematical Modelling*, 37(3):1196–1207, 2013.

[52] Vincent Martin, Jérôme Jaffré, and Jean E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691 (electronic), 2005.

[53] Ralf Massjung. An unfitted discontinuous galerkin method applied to elliptic interface problems. *SIAM Journal on Numerical Analysis*, 50(6):3134–3162, 2012.

[54] S. L. Mitchell and M. Vynnycky. Finite-difference methods with increased accuracy and correct initialization for one-dimensional Stefan problems. *Appl. Math. Comput.*, 215(4):1609–1621, 2009.

[55] Fernando Morales and R. E. Showalter. The narrow fracture approximation by channeled flow. *J. Math. Anal. Appl.*, 365(1):320–331, 2010.

[56] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations.* Cambridge press, 1995.

[57] C. S. Peskin and D. M. McQueen. A general method for the computer simulation of biological systems interacting with fluids. *Symposia of the Society for Experimental Biology*, 49:265, 1995.

[58] Michael Pruitt. Large time step maximum norm regularity of L-stable difference methods for parabolic equations. *Numer. Math.*, 128(3):551–587, 2014.

[59] Michael Pruitt. Maximum norm regularity of periodic elliptic difference operators. *ESAIM*, 2015.

[60] Y. Saad. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. 7:856–869, 1986.

[61] N. Sukumar, D. L. Chopp, and B. Moran. Extended finite element for three-dimensional fatigue crack propagation. *Engineering Fracture Mechanics*, 70:29–48, 2003.

[62] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[63] Zhijun Tan, KM Lim, and BC Khoo. An implementation of mac grid-based iim-stokes solver for incompressible two-phase flows. *Communications in Computational Physics*, 10(5):1333–1362, 2011.

[64] Lars B. Wahlbin. *Superconvergence in Galerkin finite element methods*, volume 1605 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1995.

[65] J. Wang and Y. Xiu. A weak Galerkin finite element method for second-order elliptic problems. *J. Comput. Appl. Math.*, 241:103–115, 2013.

[66] A. Wiegmann. Analytic solutions of a multi-interface transmission problem and crack approximation. *Inverse Problems*, 16:401–411, 2000.

[67] A. Wiegmann, Z. Li, and R. LeVeque. Crack jump conditions for elliptic problems. *Applied Math. Letters*, 12:81–88, 1999.

[68] H. Wu and Y. Xia. An unfitted *hp*-interface penalty finite element method for elliptic interface problems. *arXiv:1007.2893*, 2010.

[69] J. Xu. Error estimates of the finite element method for the 2nd order elliptic equations with discontinuous coefficients. *J. Xiangtan University*, No. 1:1–5, 1982.

[70] X. Yang, B. Li, and Z. Li. The immersed interface method for elasticity problems with interface. *Dynamics of Continuous, Discrete and Impulsive Systems.*, 10:783–808, 2003.

[71] Zhimin Zhang and Ahmed Naga. A new finite element gradient recovery method: superconvergence property. *SIAM J. Sci. Comput.*, 26(4):1192–1213 (electronic), 2005.

[72] Shan Zhao. A matched alternating direction implicit (adi) method for solving the heat equation with interfaces. *Journal of Scientific Computing*, 63(1):118–137, 2015.

[73] YC Zhou, Shan Zhao, Michael Feig, and Guo-Wei Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213(1):1–30, 2006.