ABSTRACT

ZHANG, SIDONG MAX. The Immersed Interface Method for Two Dimensional Poisson / Helmholtz Equations in Complex Number Space. (Under the direction of Dr. Zhilin Li.)

This thesis describes an expanded Immersed Interface Method for solving the two-dimensional Helmholtz /Poisson equations in the complex number space with an interface. Across the interface, the coefficient of the Helmholtz equation may have a finite discontinuity, and the source term of the Helmholtz /Poisson equations can have singular source terms. The solution and its normal derivatives can have discontinuities across the interface. Then we apply the developed method for solving Helmholtz /Poisson equations on irregular domains using the augmented immersed interface method.

This dissertation utilizes a combination of methodologies including the immersed interface method, Fast Fourier Transformation algorithm, augmented strategies, least squared interpolations, and the Generalized Minimal Residual method (GMRES) in complex number space for the Schur complement system. This expanded IIM is structured that the computed solutions are second order convergent towards the exact solution. Moreover, the cost of computation is designed to be efficient when solving the Schur complement system with almost constant number of iterations.

This dissertation also includes numerical experiments that have two different type problems. The results have confirmed the theoretical analysis expectation. The computed solutions showed statistically second order convergence. The proposed method is efficient, robust for wide range of real or complex wave numbers.

The Immersed Interface Method for Two Dimensional Helmholtz/Poisson Equations in Complex Number Space

by
Sidong Max Zhang

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

2013

APPROVED BY:

_____	_____
Dr. Zhilin Li Committee Chair	Dr. Alina Chertock
_____	_____
Dr. XiaoBiao Lin	Dr. Roger McCraw

DEDICATION

This work is dedicated to my mother Zhou, Renzhen for her love and encouragement.

BIOGRAPHY

Mr. Sidong Max Zhang was born in Shanghai, China, 1965. He attended middle-school and high-school at No.1 High School Affiliated to East China Normal University, where he was first inspired by mathematics, science, and computer programming. Upon graduating from Shanghai University of Science and Technology with a BS in Applied Math in 1988, he worked in Shanghai Soap Factory as a statistician and dispatcher for three years.

Mr. Zhang matriculated in Wake Forest University in 1991, and graduated with a Master of Arts degree in Math two years later. Then he survived as a statistical and sometimes IT consultant for about ten years working in various companies, there he earned some valuable interpersonal skill and experience, including the ability of listening and talking to different kinds of people.

Accidentally, Mr. Zhang found his passion for teaching mathematics. He began to pursue his Ph.D. degree at North Carolina State University in 2008, and expecting to graduate in May 2013. After that, he is going to resume his teaching career.

ACKNOWLEDGMENTS

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

First and foremost, I am very grateful for having Dr. Zhilin Li as my thesis advisor and mentor. He provided me with precise research scholarship, invaluable guidance, and vast knowledge for my Ph.D study. Whenever he traveled the world and no matter the time zone, I could always rely on email or Skype to talk to him for academic assistance, support, and encouragement. Dr. Li had profoundly improved my way of doing mathematics. Our friendship became much more than just teacher and pupil.

Besides my advisor, I would like to sincerely thank the rest of my thesis committee: Dr. Xiaobiao Lin, Dr. Alina Chertock, and Dr. Roger McGraw for their encouragement, insight, and knowledge. Dr. Lin also taught my Real and Functional Analysis courses, and the analytical skills that I learned from him will impact me far beyond this dissertation. Dr. McGraw is more than just a teacher to me, because he has given me so much advice in academics and real life that I will benefit forever.

I will always look back on my education with fond memories of the professors and staff at NCSU. Dr. Putch, Dr. Schacter, Dr. Flup, and Dr. Tran are among the best teachers that any student can have. Thanks to Ms. Denise Seabrook for helping me with all of my paper-work.

I am so fortunate to have Peng Song, GuanYu Chen, Gady and Ranya as my academic brothers/sister. Bonded with the knowledge of Immersed Interface Method, we shared the hard struggle and the sweet joy of success together.

I would also like to thank Ms Heather Manhart, Mr. Brian Tew, and Dr. Yu-Mong Hsiao for proof- reading this dissertation. Their suggestions and comments have not only made the sentences flaw more smoothly, but also added some vivid color to the language. Thanks to Dr. Meredith Williams, who has verified the statistical analysis, especially the linear regression models about convergence and their hypothesis tests, which is totally irreplaceable for proving 2^{nd} order convergence.

Last but not least, I would like to thank my family for the unselfish support emotionally and financially. My dear wife Ching, daughter Lily, and son Chris have constantly reminded me to do my "homework". I can always count on Lily for words and spelling, she is better and faster than any software checking package. Chris helped me on the computers, printers, and networks. My in-laws, He YangYuan and Lu YunFang have always been firmly believed in my ability and success. And all this would not be possible without my Mom and Dad, Zhou, RenZhen and Zhang, XiaoShi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

In this dissertation, we studies two-dimensional Helmholtz Equation in complex number space in general irregular domains with a Dirichlet boundary condition:

$$\begin{aligned} &\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+ku=f(x,y), \quad (x,y)\in\Omega\subset R^2 \\ &u(x,y)=0, \quad (x,y)\in\partial\Omega \\ &where \quad u(x,y):\Omega\to C; f(x,y):\Omega\to C; k\in C. \end{aligned} \tag{1.1}$$

These particular equations are used describe many problems related to steady-state oscillations in different media, such as mechanical, acoustical, thermal and electromagnetic phenomena. These problems also have been widely used in military and civil engineering communities.

In reality of the electromagnetic field models, the domains are more than likely to have discontinuous media with general irregular interfaces, and the models may have a complex wave number (k) which represents both electric and magnetic charges. These phenomena present extra challenges for the researchers and engineers, because the traditional methods are usually designed for rectangular domains, and real wave numbers. This dissertation is trying to make progress in answering these challenges.

1.1 The Application Problem and its Difficulty

Developed by Maxwell and Hertz, the theory describing electromagnetic waves can be written in the format of Helmholtz equation [14,27].

For example in [9,43], Figure 1.1, demonstrates the electromagnetic scatting from a two-dimensional open cavity filled with inhomogeneous media. The ground plane and the walls of the open cavity are assumed as perfect electric conductors (PEC), and the interior of the open cavity is filled with non-magnetic materials which may be inhomogeneous. The half space above the ground plan is filled with a homogenous and isotropic medium with its permittivity ε and permeability μ. Also, the electromagnetic scattering by the cavity is governed by the Helmholtz equation along with Sommerfeld's radiation conditions imposed at infinity.

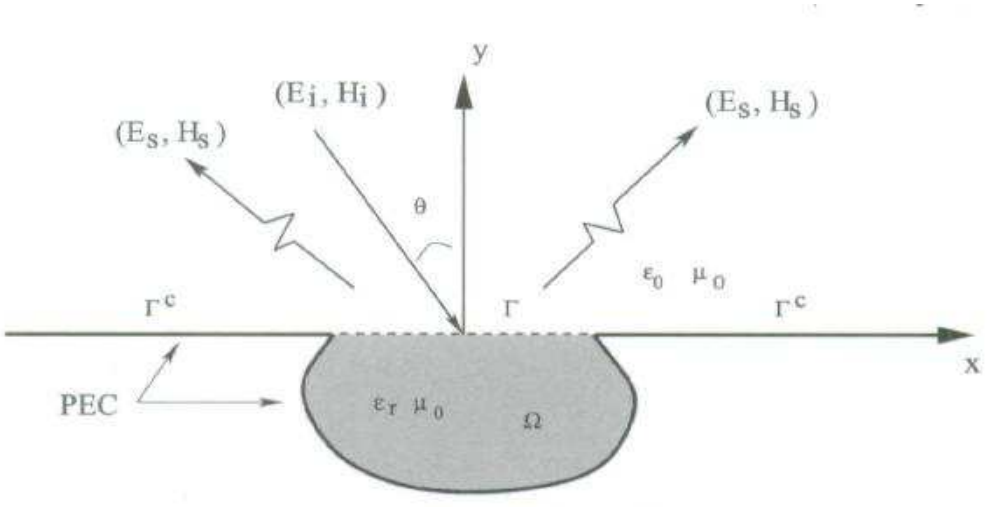


Figure 1.1 Diagram of an Electromagnetic Wave Model.

Due to difficulties in designing the finite difference approximations close to a curved boundary, the irregular domain will be embedded into a larger rectangle domain. So, the original differential equation is extended to the rectangular domain correspondingly by introducing the following jump conditions across the interface, where the interface Γ is the boundary of the original domain.

$$[u]_\Gamma = u^+ - u^- = 0$$
$$[u_n]_\Gamma = C(s).$$

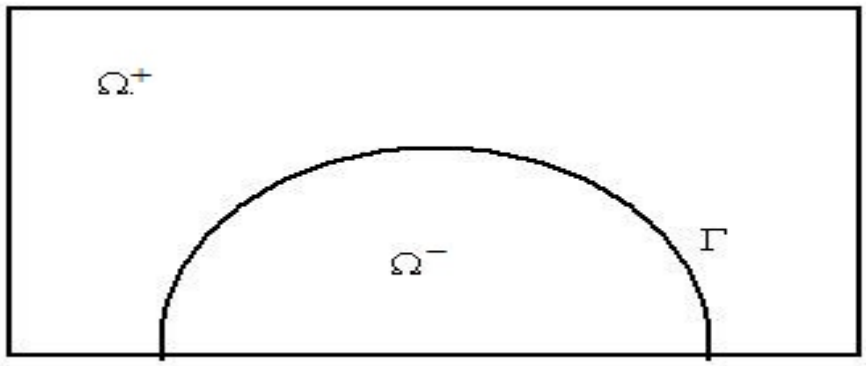


Figure 1.2 Rectangular Domain with An Embed Arbitrary Interface.

Therefore, the harmonic Maxwell equation is reduced to Helmholtz equation format:

$$\Delta u + ku = f(x,y), \qquad (x,y) \in \Omega^-,$$
$$u = 0, \qquad (x,y) \in \Omega^+,$$
$$where \qquad k = -k_0^2 \varepsilon_r \mu_0 .$$

Another advantage of embedding an irregular domain into a rectangular domain is that it is almost no computing cost in generation the grid under Cartesian coordination.

1.2 Helmholtz and Poisson Equations

The Helmholtz equation in rectangular domain has the following form

$$\Delta u + ku = -\Phi \ . \tag{1.2}$$

The particular two-dimensional Helmholtz equation in the Cartesian coordinate system:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + ku = f(x, y), \tag{1.3}$$

where k is the wave number (in some reference, it is expressed as k^2); f is a source. To simplify our notation, we write the right-hand as f(x, y) instead of $-\Phi(x, y)$ in this dissertation. If k=0, then (1.3) became a Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \ . \tag{1.3a}$$

For a homogeneous Helmholtz equation with f≡0, the general analytic solution exists, and can be written as a combination of following particular solutions

$$\begin{aligned}
&u = (A\cos\mu_1 x + B\sin\mu_1 x)(C\cos\mu_2 y + D\sin\mu_2 y), && k = \mu_1^2 + \mu_2^2. \\
&u = (A\cos\mu_1 x + B\sin\mu_1 x)(C\cosh\mu_2 y + D\sinh\mu_2 y), && k = \mu_1^2 - \mu_2^2. \\
&u = (A\cosh\mu_1 x + B\sinh\mu_1 x)(C\cos\mu_2 y + D\sin\mu_2 y), && k = -\mu_1^2 + \mu_2^2. \\
&u = (A\cosh\mu_1 x + B\sinh\mu_1 x)(C\cosh\mu_2 y + D\sinh\mu_2 y), && k = -\mu_1^2 - \mu_2^2.
\end{aligned} \tag{1.4}$$

For inhomogeneous Helmholtz equation in the rectangle domain (say $0 \le x \le a$, $0 \le y \le b$), with Dirichlet boundary condition as prescribed:

$$\begin{aligned}
&u(0, y) = f_1(y), && u(a, y) = f_2(y), \\
&u(x, 0) = f_3(x), && u(x, b) = f_4(x).
\end{aligned} \tag{1.5}$$

The solution for Helmholtz problem exists if $k \neq \pi^2(\frac{n^2}{a^2} + \frac{m^2}{b^2})$; $\quad n = 1,2,... \quad m = 1,2,...$

The solution can be written analytically in an integral equation involving Green's function

$$u(x,y) = \int_0^a \int_0^b f(\xi,\eta) G(x,y,\xi,\eta) d\eta d\xi + \int_0^b f_1(\eta)[\frac{\partial}{\partial \xi} G(x,y,\xi,\eta)]_{\xi=0} d\eta - \int_0^b f_2(\eta)[\frac{\partial}{\partial \xi} G(x,y,\xi,\eta)]_{\xi=a} d\eta + \int_0^b f_3(\xi)[\frac{\partial}{\partial \eta} G(x,y,\xi,\eta)]_{\eta=0} d\xi - \int_0^b f_4(\xi)[\frac{\partial}{\partial \eta} G(x,y,\xi,\eta)]_{\eta=b} d\xi.$$

where the Green's function G(*) has the following forms of representation:

$$G(x,y,\xi,\eta) = \frac{2}{a}\sum_{n=1}^{\infty} \frac{\sin(p_n x)\sin(p_n \xi)}{\beta_n \sinh(\beta_n b)} H_n(y,\eta)$$

or

$$G(x,y,\xi,\eta) = \frac{2}{b}\sum_{m=1}^{\infty} \frac{\sin(q_m y)\sin(q_m \xi)}{\mu_m \sinh(\mu_m a)} Q_m(x,\xi),$$

where

$$p_n = \frac{n\pi}{a}, \qquad \beta_n = \sqrt{p_n^2 - k},$$
$$q_m = \frac{m\pi}{b}, \qquad \mu_m = \sqrt{q_m^2 - k},$$

and

$$H_n(y,\eta) = \begin{cases} \sinh(\beta_n \eta)\sinh[\beta_n (b-y)], & for \quad 0 \le \eta \le y \le b, \\ \sinh(\beta_n y)\sinh[\beta_n (b-\eta)], & for \quad 0 \le y \le \eta \le b, \end{cases}$$
$$Q_m(x,\xi) = \begin{cases} \sinh(\mu_m \xi)\sinh[\mu_m (a-x)], & for \quad 0 \le \xi \le x \le a, \\ \sinh(\mu_m x)\sinh[\mu_m (b-\xi)], & for \quad 0 \le x \le \xi \le a. \end{cases}$$

Though, we can analytically give out the solution for rectangular domains and boundary condition [41], they will become very complicated and not practical for physics and engineering use when the domain is general and irregular. Thus, numerical methods and solutions are still required and in high demand [35].

1.3 Review of Existing Numerical Methods

For a problem defined on an irregular domain, the method most often used is the embedding technique. We will discuss in detail later in chapter 3 and 4 of this dissertation. Then the problem can be treated as a special interface problem. Therefore, the terminology of *interface problems* is introduced to include the problems defined on irregular domains.

There are many algorithms and methods discussed in research papers in the literature that address the interface problem. For example, smoothing method for discontinuous coefficients, harmonic averaging for discontinuous coefficients, immersed boundary method, numerical integral equation method, ghost fluid method, and immersed interface method are the most frequently referenced in the related fields. This dissertation just focus on the two dimensional method that has closest relation to our algorithm, that is the Immersed Boundary method, the numerical integral equation method, and existing Immersed Interface Method.

1.3.1 Immersed Boundary (IB) method

This Immersed Boundary method was originally developed by Peskin [29, 30, 31, 32, 33] to model the blood flow in a human heart, and has been applied to many other problems, particularly in biophysics.

One of the most important ideas in the IB method is the use of a discrete delta function to distribute a singular source to nearby grid points. The commonly used discrete delta functions in one dimension are as following.

Hat delta function

$$\delta_\varepsilon(x) = \begin{cases} (\varepsilon - |x|)/\varepsilon^2, & if\ |x| < \varepsilon, \\ 0, & if\ |x| \geq \varepsilon. \end{cases}$$

Cosine delta function

$$\delta_\varepsilon(x) = \begin{cases} \frac{1}{4\varepsilon}(1+\cos(\frac{\pi x}{2\varepsilon}), & if\ |x| < 2\varepsilon, \\ 0, & if\ |x| \geq 2\varepsilon. \end{cases}$$

And Radial Delta function

$$\delta_\varepsilon(x) = \begin{cases} \frac{1}{8\varepsilon}\left(3 - \frac{2|x|}{\varepsilon} + \sqrt{1 + \frac{4|x|}{\varepsilon} - \frac{4x^2}{\varepsilon^2}}\right), & if \quad |x| < \varepsilon \quad , \\ \frac{1}{8\varepsilon}\left(5 - \frac{2|x|}{\varepsilon} - \sqrt{-7 + \frac{12|x|}{\varepsilon} - \frac{4x^2}{\varepsilon^2}}\right), & if \quad \varepsilon \leq |x| < 2\varepsilon \quad , \\ 0, & if \quad |x| \geq 2\varepsilon \quad . \end{cases}$$

In two dimensions, the discrete delta function often is the product of one dimensional discrete delta functions, such as $\delta_\varepsilon(x,y) = \delta_\varepsilon(x)\delta_\varepsilon(y)$. And the discretized Helmholtz equation at (x_i, y_j) became

$$\sum_{k=1}^{k_s} \gamma_k u_{i+i_k, j+j_k} + k u_{i,j} = \delta_h(x_i - x_i^*)\delta_h(y_i - y_i^*) f_{ij},$$

where k_s is the number of discrete points $\{(x_i^*, y_j^*)\}$ on the interface, $\{\gamma_k\}$'s are the coefficients that involved in the finite difference scheme, h is the mesh spacing. In this way, the singular source is distributed to nearby grids points in a neighborhood of the interface Γ.

The Immersed Boundary method is robust and simple to implement. It has been applied to many problems in mathematical biology and computational fluid mechanics [1, 2, 4, 5, 7, 8, 10, 11, 12, 39, 42, 44]. Various work has been developed to improve the accuracy of the IB method, and it is most time first order convergence results [29, 31,30], with some occasional second order convergence [17, 34]. But, there is not yet any complete analytical convergence proof for the IB method [30]. However, stability analysis of the IB method is given in [37, 38] for a membrane problem.

1.3.2 Integral Equation Method

Greenhaum, Mayo and their collaborators [23, 24, 25, 26] are among the few who first combined integral equation based on the single and double layer theory with finite difference methods to solve a Poisson equation on an irregular domain. The irregular domain is embedded into a larger rectangle, and then the problem is recast as an elliptic interface problem such that the solution is harmonic in the rectangle, excluding the boundary. Taylor expansions at irregular grid points and the integral representation of the particular solution near the irregular boundary are used. The source strength is determined from an integral equation. The jump conditions are derived from the integral equation and are used to derive

the finite difference schemes at all grid points in the rectangular domain so that a fast Poisson solver can be used.

By solving the integral equation and a regular Poisson equation, the algorithm is somehow second order accuracy in L^{∞} norm. The numerical integral equation methods are most effective for homogeneous source terms and certain boundary conditions. Although this method still can be applied for nonhomogeneous source terms and different boundary conditions if with some extra effort. The implementations of these methods, especially when they are coupled with the fast multipole method, are difficult.

1.3.3 Immersed Interface Method

Immersed Interface Method is first developed by LeVeque and Li in 1994 [19]. It is motived by Peskin's IB method, but there are remarkable improvements in IIM. IIM is a sharp interface method which the discontinuities or the jump condition are enforced exactly by prior knowledge or approximately through Augmented strategy [20, 16].

In general cases, standard finite difference methods are used in discretization. At the grid point near or on the interface, a correction term is added according to the jump condition to ensure point-wise convergence. By this approaching, IIM can still take advantage of existing numerical algorithm to solve the differential equation system. In most cases, IIM can achieve second order global accuracy under the infinity norm L^{∞} .

This dissertation is trying further expending Immersed Interface Method into complex-numbered wave number and function, while thus preserving its advantages, like efficient and stable solution with second order convergence.

1.4 Outline of the thesis

Chapter One surveys the general IIM background and literature. The fundamental concepts such as interface problems and jump condition are introduced. The last section describes the structure of this thesis.

In Chapter Two, we studied the solution for Poisson and Helmholtz equations in rectangular domains without interface jump conditions. A new Fast Fourier Transformation method is derived and analyzed for the efficiency and stability.

In Chapter Three, we introduced correction terms at irregular grid points such that the proposed algorithm is 2^{nd} order convergence.

In Chapter Four, we further studied some unknown interface conditions. Augmented Strategies are used to assume one of the unknown interface variables, and estimated by weighed least square interpolation, then we created Schur complement system. Finally, GMRES method was used to solve for the whole system.

Two examples of numerical experiments are presented in Chapter Five. Under different functions, interface condition and computing circumstances, all results are in line with our previous analysis.

The last chapter summarizes the contributions we have achieved and discusses several possible future research topics.

At the end of the thesis is a list of papers, books and presentations which this research has referenced.

CHAPTER 2

ZFFT methods for solving two dimensional Poisson / Helmholtz Equation in Rectangular Domains

The Fast Fourier Transformation (FFT) method for solving two dimensional Poisson's equations was first introduced by Cooley and Tukey [15] in 1965. It was focused on squared domains with uniformed mesh space. The main idea of this method is to take advantage of some beautiful properties of the discrete Fourier transformation, which is able to decompose the tridiagonal matrix into multiplication of eigenvalues and their eigenvectors. Therefore, by substituting the variables back and forth twice, the FFT method only needs O(NlogN) multiplications instead of computing the inverse of the matrix, thus it is much faster.

In 1984, Swarztrauber further developed the FFT method for rectangular domains [40]. He first converted the two-dimensional solution $u_{i,j}$ into 1-dimensional vector u_{mi+j}, and turned the finite difference scheme into a tridiagonal block matrix, and then solve the linear equations system using row reduction method. This method achieved O(NlogN) ($N=m\times n$) efficiency. But it is somehow confusing, and not easy to understand and implement.

To ensure the efficiency of our computing, the FFT method in this dissertation is designed for rectangular domains. For inscribing an arbitrary shape into another geometry shape, a rectangular is more than likely to cover less area than a square. Thus the rectangular constructs less grid points, and therefore less computation.

In addition, we also need double complex precision for our data to keep rounding errors from distorting our computing result. For example, we need the Schur complement residues in very high precision so that the jump condition in the interface can be accurately obtained.

However, there is no existing FFT method that fit for our requirements at the time. Besides, modern day computing environment is completely different from the 60s to 80s, the old Fortran code that developed at that time may be obsolete and may not be complied smoothly nowadays. So, we decided to develop our very own Fast Fourier Transformation methods for rectangular domains with double complex precision. We would like to call the new algorithm ZFFT, it is because this Fast Fourier Transformation method is dealing with complex number. There three slightly variation of the algorithms total, we are going to introducing them one by one.

2.1 ZFFT Method with hx ≠ hy, and m = n

First, we consider the discrete finite difference equation for Poisson equation in a rectangular domain with the standard 5 point finite difference scheme at a rectangular domain can be written as

$$\frac{u_{i+1,j} + u_{i-1,j} - 2u_{ij}}{{h_x}^2} + \frac{u_{i,j+1} + u_{i,j-1} - 2u_{ij}}{{h_y}^2} \approx f_{ij}, \qquad i, j = 1, \cdots, n \qquad , \tag{2.1}$$

where hx and hy are the mesh spaces in their respect x-axis and y-axis direction, and we assume that hx ≠ hy, and m = n. Notice that $f_{i,j}$ is the right hand side value intergraded with boundary condition. To rewrite (2.1) in matrix format

$$\frac{1}{h_x^2}TU + \frac{1}{h_y^2}UT \approx F \quad , \tag{2.2}$$

where

$$T = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -2 \end{pmatrix}_{n,n} \quad and \quad U = \left(u_{i,j}\right)_{n,n}, \quad F = \left(f_{i,j}\right)_{n,n} .$$

Apply discrete Fourier Transformation to each column of T, then $T=V^{-1}DV$. (2.2) became

$$\begin{aligned} & h_y^2 V^{-1} DVU + h_x^2 UV^{-1} DV = h_x^2 h_y^2 F, \\ & where \quad V = \left(S_{i,j}\right)_{n,n}, \quad S_{i,j} = \sin(\frac{2\pi ij}{2(n+1)}), \\ & D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_{12} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}, \quad \lambda_i = -4\sin^2(\frac{i\pi}{2(n+1)}), \quad i, j = 1...n. \end{aligned} \tag{2.3}$$

It is worth to point out that V has another beautiful property that

$$V^{-1} = \frac{2}{n+1}V \quad . \tag{2.4}$$

Multiply $h_x^2 h_y^2$, and V from left and V^{-1} from right to (2.3) at both sides, then

$$h_y^2 DVUV^{-1} + h_x^2 VUV^{-1} D = h_x^2 h_y^2 V FV^{-1} \quad . \tag{2.5}$$

Now, let $\overline{U} = VUV^{-1} = (\overline{u}_{i,j})_{n,n}$, and $\overline{F} = h_x^2 h_y^2 VFV^{-1} = (\bar{f}_{i,j})_{n,n}$, then

$$h_y{}^2 D\overline{U} + h_x{}^2 \overline{U}D = \overline{F} \quad . \tag{2.6}$$

Since D is diagonal matrix, then $\bar{U}$ in (2.5) can be easily solved

$$\bar{u}_{i,j} = \frac{\bar{f}_{i,j}}{h_y^2\lambda_i + h_x^2\lambda_j}, \quad where \ \ i, j = 1\ldots n \ . \tag{2.7}$$

Further, for Helmholtz equation, (2.7) can be

$$\bar{u}_{i,j} = \frac{\bar{f}_{i,j}}{h_y^2\lambda_i + h_x^2\lambda_j + h_x^2 h_y^2 k}, \quad where \ \ i, j = 1\ldots n \, . \tag{2.8}$$

Once we have $\bar{U}$, then reverse discrete Fourier Transformation,

$$U = V^{-1}\bar{U}V \ . \tag{2.9}$$

This Fast Fourier Transformation method is directly derived from the traditional method for square domain. The tridiagonal matrix T is n×n square matrix. However, we estimate the locate truncation error through (2.1), we can find out that it may not guaranteed to be second order convergence for the computed solutions. So we keep on working on the next method.

2.2 ZFFT Method with hx = hy, and m ≠ n

In this method, we still start the finite difference equation for Poisson equation in a rectangular domain with the standard 5 point finite difference scheme. This time, the mesh spacings toward the x-axis and the y-axis directions are equal ($h = h_x = h_y$), but the number of

grid points in that 2 direction are not equal (n ≠ m). It can be written as

$$\frac{u_{i+1,j}+u_{i-1,j}-2u_{ij}}{h^2}+\frac{u_{i,j+1}+u_{i,j-1}-2u_{ij}}{h^2}\approx f_{ij},\qquad i=1,\cdots,m;\quad j=1,\cdots,n. \tag{2.10}$$

Notice that $f_{i,j}$ is the right hand side value intergraded with boundary condition. Rewrite (2.10) in matrix format

$$T_mU+UT_n=h^2F, \tag{2.11}$$

where

$$T_m=\begin{pmatrix}-2&1&0&\cdots&0\\1&-2&1&\cdots&0\\0&1&-2&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&-2\end{pmatrix}_{m,m},\quad T_n=\begin{pmatrix}-2&1&0&\cdots&0\\1&-2&1&\cdots&0\\0&1&-2&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&-2\end{pmatrix}_{n,n},$$
$$U=\left(u_{i,j}\right)_{m,n},\quad F=\left(f_{i,j}\right)_{m,n}. \tag{2.12}$$

By discrete Fourier Transformation, $T_s= V_s^{-1}D_sV_s$, s = m, n, then

$$V_m^{-1}D_mV_mU+UV_n^{-1}D_nV_n=h^2F\ . \tag{2.13}$$

Again, Multiplying h^2, V_m from left and V_n^{-1} from right at both sides of the (2.13), we have

$$D_mV_mUV_n^{-1}+V_mUV_n^{-1}D_n=h^2V_mFV_n^{-1}\ . \tag{2.14}$$

Let

$$\overline{U}=V_mUV_n^{-1}=(\overline{u}_{i,j})_{m,n}\ ,\ \text{and}\ \ \widetilde{F}=h^2V_mFV_n^{-1}=(\widetilde{f}_{i,j})_{m,n}\quad . \tag{2.15}$$

Then substitute (2.15) into (2.14), it become

$$D_m\overline{U}+\overline{U}D_n=\widetilde{F}\ . \tag{2.16}$$

Since D_m and D_n in (2.16) are diagonal, we can simply solve (2.16) for $\bar{U}$,

$$\overline{u}_{i,j} = \frac{\widetilde{f}_{i,j}}{\lambda^{(m)}{}_i + \lambda_j{}^{(n)}}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n \ . \tag{2.17}$$

Similarly, for Helmholtz equation

$$\overline{u}_{i,j} = \frac{\widetilde{f}_{i,j}}{\lambda^{(m)}{}_i + \lambda_j{}^{(n)} + h^2 k}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n \ . \tag{2.18}$$

Reverse the discrete Fourier transformation, we get

$$U = V_m^{-1} \overline{U} V_n \quad . \tag{2.19}$$

This method uses traditional equal mesh spacing, while the numbers of grid points are different in x-axis and y-axis direction. It is most compliable with the traditional numerical analysis, and we can still take advantage of many existing software packages. In this method, we have to construct two different set of discrete Fourier Transformation in (2.13), which cost little extra computing. When it was used iteration like GMRES, we can store and re-use them rather constructing from new each time. More important, the locate truncation error can be estimated as usual, which we will prove later that it is second order convergence for the computed solution against exact solution. So we decided to use this ZFFT II method for the rest of our dissertation.

2.3 ZFFT Method with hx ≠ hy, and m ≠ n

For the completeness of this academic exploration, we want to further study the method with rectangular mesh spacing and unequal number of grid points. Let us consider discrete Poisson equation with, hx ≠ hy and m ≠ n, then

$$\frac{u_{i+1,j}+u_{i-1,j}-2u_{ij}}{h_x^{\,2}}+\frac{u_{i,j+1}+u_{i,j-1}-2u_{ij}}{h_y^{\,2}}=f_{ij}\,,\quad i=1,\cdots,m;\quad j=1,\cdots n. \tag{2.20}$$

Notice that $f_{i,j}$ is the right hand side value intergraded with boundary condition. Rewrite (2.20) as matrix format, then

$$\frac{1}{h_x^2}T_mU+\frac{1}{h_y^2}UT_n\approx F\,, \tag{2.21}$$

where Tm and Tn are defined in (2.12)

Apply discrete Fourier Transformation to each column of T_s , i.e.

$$T_m=V_m^{-1}D_mV_m\ \text{, and}\ T_n=V_n^{-1}D_nV_n\ , \tag{2.22}$$

where $V_n=\left(S_{i,j}^{(n)}\right)_{n,n}$, $S_{i,j}^{(n)}=\sin(\frac{2\pi ij}{2(n+1)})$, $i,\ j=1\ldots n$.

$$D_n=\begin{pmatrix}\lambda_1^{(n)} & 0 & \cdots & 0\\ 0 & \lambda_2^{(n)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n^{(n)}\end{pmatrix},\quad \lambda_i^{(n)}=-4\sin^2(\frac{i\pi}{2(n+1)}).$$

Then, we have

$$h_y^2V_m^{-1}D_mV_mU+h_x^2UV_n^{-1}D_nV_n=h_x^2h_y^2F\,. \tag{2.23}$$

Now, Multiplying $h_x^2 h_y^2$, Vm from left and Vn^{-1} from right for both sides of (2.23), then

$$h_y^2 D_m V_m U V_n^{-1} + h_x^2 V_m U V_n^{-1} D_n = h_x^2 h_y^2 V_m F V_n^{-1}. \tag{2.24}$$

Let

$$\overline{U} = V_m U V_n^{-1} = (\overline{u}_{i,j})_{m,n}, \text{ and } \hat{F} = h_x^2 h_y^2 V_m F V_n^{-1} = (\hat{f}_{i,j})_{m,n}, \tag{2.25}$$

then (2.24) become

$$h_y^2 D_m \overline{U} + h_x^2 \overline{U} D_n = \hat{F}. \tag{2.26}$$

Since D_m and D_n are diagonal, it is easy to solve (2.26) for $\bar{U}$,

$$\overline{u}_{i,j} = \frac{\hat{f}_{i,j}}{h_y^2 \lambda^{(m)}{}_i + h_x^2 \lambda_j^{(n)}}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n. \tag{2.27}$$

Similarly, for Helmholtz equation,

$$\overline{u}_{i,j} = \frac{\hat{f}_{i,j}}{h_y^2 \lambda^{(m)}{}_i + h_x^2 \lambda_j^{(n)} + h_x^2 h_y^2 k}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n. \tag{2.28}$$

Once we have $\bar{U}$, then by the reversed discrete Fourier Transformation (2.19), it is easy to get the solution U.

This third ZFFT method is the combination of ZFFT method I and ZFFT method II. It is a generalized algorithm that solves Poisson or Helmholtz equation. In practice of this research, it is an over kill to simulate the rectangular shape by both measures of rectangular grid shape and unequal mesh size. However, somebody may need this method for future research someday.

2.4 ZFFT Method Summary

We mainly focus on the second complex-numbered Fast Fourier Transformation method (ZFFT II) in rectangular domain for this summary. Other methods in this chapter are almost identical. Here is the step by step approaching of the algorithm

1. Perform discrete Fourier Transformation on F, get

$$\overline{F} = h^2 V_m F V_n^{-1} = \frac{2h^2}{n+1} V_m F V_n \ . \tag{2.29}$$

2. Compute middle solution Ū:

$$\bar{u}_{i,j} = \frac{\bar{f}_{i,j}}{\lambda_i^{(m)} + \lambda_j^{(n)} + h^2 k}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n \ , \tag{2.30}$$

$$\text{where } \lambda_i^{(m)} = -4 \sin^2 \left(\frac{i\pi}{2(m+1)} \right), \quad \lambda_j^{(n)} = -4 \sin^2 \left(\frac{j\pi}{2(n+1)} \right).$$

3. Perform reversed discrete Fourier Transformation on Ū, get

$$U = V_m^{-1} \overline{U} V_n = \frac{2}{m+1} V_m \overline{U} V_n \quad . \tag{2.31}$$

2.5 Efficiency Analysis

From previous section, we can see that the cost of step Two is 3×m×n operation, and step One and Three is seems to be a triple matrix multiplication each. If we do use the

straightforward matrix multiplication method, then the cost is m×m×n+ m×n×n flops, which is $O(m^3)$. However, the triple matrix multiplication actually represents the Discrete Fourier Transformation. Thus, we can use convolutions and recursive algorithm, therefor the computation cost is reduce the (m^2 log_2 m) [6]. Here is the process for V_mFV_n as an example.

Let $F_j=\{f_{1j},f_{2j}, \dots , f_{mj}\}^T$ be the the jth column of the F matrix, and

$$\begin{aligned} &a(\omega) = f_{1j} + f_{2j}\omega + f_{3j}\omega^2 + \cdots + f_{mj}\omega^m, \\ &where \quad \omega = e^{\frac{-2\pi i}{m+1}} = \cos(\frac{2\pi}{m+1}) - i \cdot \sin(\frac{2\pi}{m+1}), \quad i = \sqrt{-1}. \end{aligned} \tag{2.32}$$

ω is also known as a principal (m+1)th root of unity. The DFT of F_j is just the polynomial (2.32) evaluation at the points $\{\omega^0, \omega^1, \omega^2, \dots, w^{m-1}\}$. Conversely, the inversed DFT is the polynomial interpolation producing the coefficients of a polynomial given its values at $\{\omega^0, \omega^1, \omega^2, \dots, w^{m-1}\}$.

Assume $m=2^s$, then we divide the polynomial (2.32) into two equal pieces.

$$\begin{aligned} a(\omega) &= f_{1j} + f_{2j}\omega + f_{3j}\omega^2 + \cdots + f_{mj}\omega^m \\ &= (f_{1j} + f_{3j}\omega^2 + f_{5j}\omega^4 \cdots) + \omega(f_{2j} + f_{4j}\omega^2 + f_{6j}\omega^4 \cdots) \\ &= a_{odd}(\omega^2) + \omega \cdot a_{even}(\omega^2). \end{aligned} \tag{2.33}$$

From above, to evaluate two polynomials a_{odd} and a_{even} of degree m/2-1 at $(\omega^j)^2$. But this is really just m/2 points ω^{2j} for $0 \le j \le m/2-1$ since $\omega^{2j} = \omega^{2(j+\frac{m}{2})}$. Thus evaluating a polynomial of degree m-1=2s-1 at all m (m)th roots of unity is the same as evaluating two polynomials of dgree m/2-1 at all m/2 (m/2)th roots, and then combining the results with m

multiplication additions. This can be done recursively with following algorithm

$$
\begin{aligned}
&\textit{function} \quad DFT(\vec{a}) \\
&\quad n = size(\vec{a}); \\
&\quad \textit{if} \quad n = 1 \quad \textit{then} \quad \textit{return} \quad \vec{a} \\
&\quad \omega = e^{(2\pi/n)} \\
&\quad \vec{\omega} = (\omega^0, \omega^1, \omega^2, \cdots \omega^{n/2-1}) \\
&\quad \vec{a}_{odd} = (a_1, a_3, a_5, \cdots a_{n-1}) \\
&\quad \vec{a}_{even} = (a_2, a_4, a_6, \cdots a_n) \\
&\quad \vec{y}_{odd} = DFT(\vec{a}_{odd}) \\
&\quad \vec{y}_{even} = DFT(\vec{a}_{even}) \\
&\quad \vec{y}_{odd} = \vec{a}_{odd} + \vec{\omega}^T \cdot \vec{y}_{even} \\
&\quad \vec{y}_{even} = \vec{a}_{odd} - \vec{\omega}^T \cdot \vec{y}_{even} \\
&\quad \textit{return} \quad \vec{y} \\
&\textit{end}
\end{aligned}
$$

Therefore, the computing cost of each column of a m×n matrix is $\log_2 m \cdot \frac{3m}{2}$, and by same method, the cost of each row of each column of a m×n matrix is $\log_2 n \cdot \frac{3n}{2}$. So, the total computing cost for a m×n matrix is $O(m \cdot n \cdot \log(m \cdot n))$. The recursive method requires huge memory for stocking of heap, which was not available in the 6os or 80s. Also for small number mesh size, there are no significant different from recursive method compare to direct matrix multiplication.

2.6 Error Analysis

Since we have u(x, y): $R^2 \rightarrow C$, then we can write

u(x, y) = v(x, y) + i·w(x, y) where v, w : $R^2 \rightarrow R$.

First, let us consider the locate truncation error (LTE) for the standard 5-point finite difference scheme for real function v(x,y) at grid point (x_i, y_j).

$$LTE(v_{ij}) = (\frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y})v - \frac{v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j}}{h^2}, \tag{2.34}$$

where i=1,…, m; j=1,…, n, and $v_{ij} = v(x_i, y_j)$. By Taylor expansion;

$$\begin{aligned}
v_{i+1,j} &= v_{i,j} + \frac{\partial}{\partial x}(v_{i,j})h + \frac{1}{2}\frac{\partial^2}{\partial^2 x}(v_{i,j})h^2 + \frac{1}{3!}\frac{\partial^3}{\partial^3 x}(v_{i,j})h^3 + \frac{1}{4!}\frac{\partial^4}{\partial^4 x}(v_{i,j})h^4 + O(h^5), \\
v_{i-1,j} &= v_{i,j} - \frac{\partial}{\partial x}(v_{i,j})h + \frac{1}{2}\frac{\partial^2}{\partial^2 x}(v_{i,j})h^2 - \frac{1}{3!}\frac{\partial^3}{\partial^3 x}(v_{i,j})h^3 + \frac{1}{4!}\frac{\partial^4}{\partial^4 x}(v_{i,j})h^4 + O(h^5), \\
v_{i,j+1} &= v_{i,j} + \frac{\partial}{\partial y}(v_{i,j})h + \frac{1}{2}\frac{\partial^2}{\partial^2 y}(v_{i,j})h^2 + \frac{1}{3!}\frac{\partial^3}{\partial^3 y}(v_{i,j})h^3 + \frac{1}{4!}\frac{\partial^4}{\partial^4 y}(v_{i,j})h^4 + O(h^5), \\
v_{i,j-1} &= v_{i,j} - \frac{\partial}{\partial y}(v_{i,j})h + \frac{1}{2}\frac{\partial^2}{\partial^2 y}(v_{i,j})h^2 - \frac{1}{3!}\frac{\partial^3}{\partial^3 y}(v_{i,j})h^3 + \frac{1}{4!}\frac{\partial^4}{\partial^4 y}(v_{i,j})h^4 + O(h^5).
\end{aligned} \tag{2.35}$$

Substitute (2.35) into (2.34), and simplify

$$LTE(v_{ij}) = \frac{1}{12}h^2(\frac{\partial^4 v}{\partial x^4} + \frac{\partial^4 v}{\partial y^4}) + O(h^3) = O(h^2). \tag{2.36}$$

By the same method, we can have the locate truncation error for w

$$LTE(w_{ij}) = \frac{1}{12}h^2(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4}) + O(h^3) = O(h^2). \tag{2.37}$$

Finally, with the norm for complex number, the locate truncation error for u is

$$
\begin{aligned}
LTE\,(u_{ij}) &= |\, LTE\,(v_{ij}) + i \cdot LTE\,(wi_{j})\,| \\
&= \left\| \frac{1}{12} h^2 \left(\frac{\partial^4 v}{\partial x^4} + \frac{\partial^4 v}{\partial y^4} \right) + O(h^3) + i \left[\frac{1}{12} h^2 \left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} \right) + O(h^3) \right] \right\| \\
&= \sqrt{\left[\frac{1}{12} h^2 \left(\frac{\partial^4 v}{\partial x^4} + \frac{\partial^4 v}{\partial y^4} \right) + O(h^3) \right]^2 + i \left[\frac{1}{12} h^2 \left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} \right) + O(h^3) \right]^2} \\
&= h^2 \sqrt{\left[\frac{1}{12} \left(\frac{\partial^4 v}{\partial x^4} + \frac{\partial^4 v}{\partial y^4} \right) + O(h^1) \right]^2 + i \left[\frac{1}{12} \left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} \right) + O(h^1) \right]^2} = O(h^2).
\end{aligned}
\tag{2.38}
$$

Furthermore, T is diagonally dominant, thus the global error is $O(h^2)$ for Poisson equations. Very similar for Helmholtz equation, if the exact solution exists, i.e.

$$
h^2 k \neq -\lambda_i^{(m)} - \lambda_j^{(n)}, \quad i = 1, \cdots, m; \quad j = 1, \cdots, n. \tag{2.34}
$$

Then the location truncation error is $O(h^2)$ and the global error is also $O(h^2)$.

CHAPTER 3

Immersed Interface Method for Two dimensional Poisson / Helmholtz Equation in Complex Number Space

In this chapter, we try to develop an improved finite difference scheme to handle the discontinuous f and coefficients cross the interface in complex number space. The real number two-dimensional elliptic interface problem was solved using immersed interface method [20]. Our approach is also based on immersed interface method, and very similar to the real number one. The goal of our approach is to obtain a finite difference scheme that works with discontinuous f and second order convergence guaranteed.

3.1 Interface Embedding

Let Ω be a convex domain in two dimensions within there is an irregular interface Γ. Let Ω^+ and Ω^- be the two regions of the interface, see following Figure 3.1.

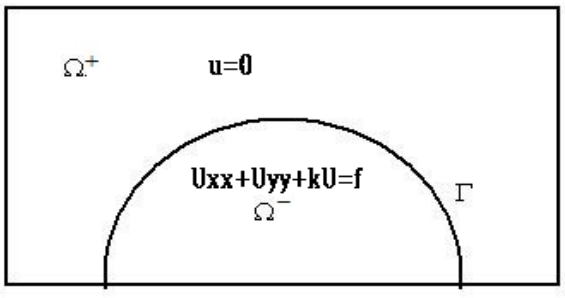


Figure 3.1 Function Domain and Embed Interface.

We are considering the following Helmholtz / Poisson problem

$$Uxx + Uyy + kU = f, \qquad in \quad \Omega \ . \tag{3.1}$$

With some boundary condition on $\partial\Omega$ and jump conditions on the interface Γ.

$$[u]_\Gamma = u^+ - u^- = w. \tag{3.2}$$

$$[u_n]_\Gamma = \frac{\partial u^+}{\partial n} - \frac{\partial u^-}{\partial n} = g. \tag{3.3}$$

In this study, we assume that the interface Γ is arbitrarily smooth, Ω is piecewise smooth, k and f are piecewise continuous in Ω^+ and Ω^- respectably, and along the interface, w has continuous second derivatives and q has continuous first derivatives. Then the solution u has piecewise second order derivatives components in Ω, that is $u \in C^2$ in Ω^+ or Ω^-, but not in Ω .

Since the f and/or k may be discontinuous across the interface, the solution and its derivatives may also be non-smooth or even discontinuous across the interface. Therefore the traditional standard finite difference schemes will not work properly for this class of problems.

Below is outline of our approach, step by step:

- Select a point (x_i^*, y_j^*) on interface Γ near grid point (x_i, y_j).
- Apply a local coordinate transformation in the directions normal and tangential to Γ at (x_i^*, y_j^*).
- Derive the interface relations relating + or – values at (x_i^*, y_j^*) in local coordinates.

- Choose some additional points to form a modified stencil.
- Setup and solve a system of linear equations for the coefficients γ_k's.
- Compute the correction term C_{ij}.
- Add C_{ij} into standard finite difference scheme for PDE, then solve.

3.2 Local Coordinate Transformation

Unless otherwise stated, we are using uniformed mesh grid size, $h_x=h_y$ and the traditional standard five-point finite difference stencil in our study.

Definition: a grid point is called **regular** if all the grid points in the centered 5-point stencil are on the same side of the interface; otherwise, a grid point is called **irregular** if not all the grid points in the centered 5-point stencil are on the same side of the interface.

For example, in Figure 3.1, point 3, 4, 7 are regular grid points, while point 1,2,5,6,8,9 are irregular grid points.

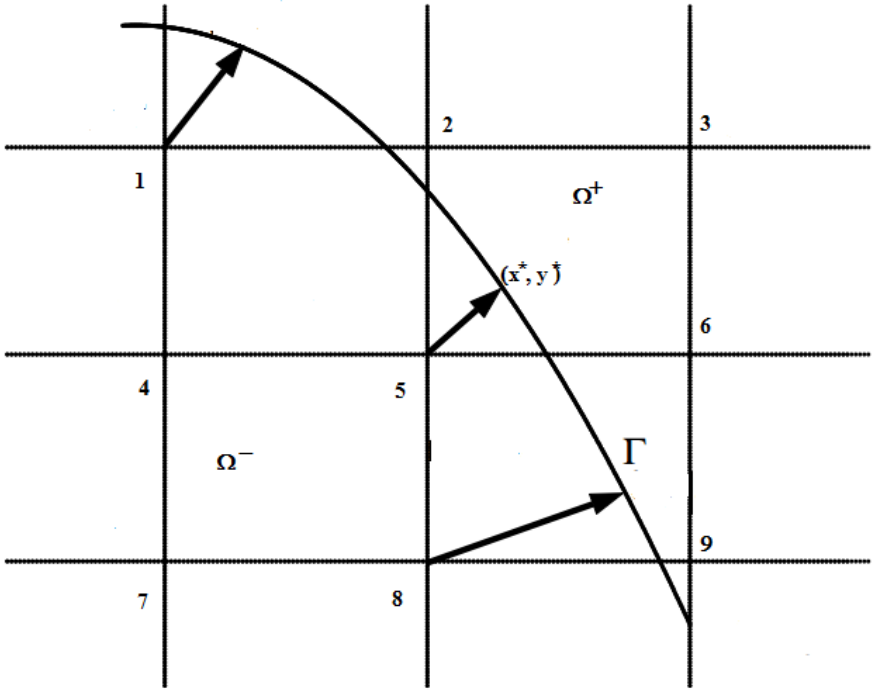


Figure 3.2 Interface, Regular and Irregular Points.

We consider a fixed point (x*,y*) on the interface, and define a local ξ-η coordinate system by

$$\begin{cases} \xi = (x - x^*)\cos\theta + (y - y^*)\sin\theta, \\ \eta = -(x - x^*)\sin\theta + (y - y^*)\cos\theta. \end{cases} \tag{3.4}$$

where θ is the angle between the x-axis and the normal direction, pointing to the direction of a specified side, say the "+" side. At the point (x*, y*), the interface Γ can be written as

$$\xi = \chi(\eta) \qquad with \qquad \chi(0) = 0, \quad \chi'(0) = 0\,. \tag{3.5}$$

The curvature of the interface at (x*,y*) is χ''.

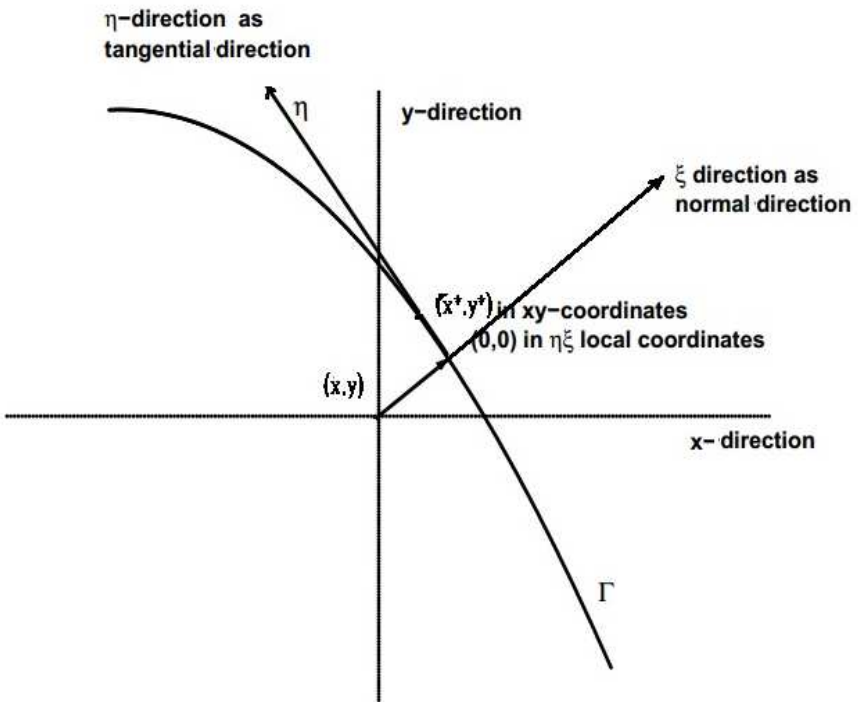


Figure 3.3 An Irregular Grid Point, its Orthogonal Projection, and the Local Coordination.

Note that under the local coordinate transformation (3.4), the partial differential equation (3.1) remains unchanged, that is:

$$[u_{\xi\xi} + u_{\eta\eta} + ku] = [f]. \tag{3.1a}$$

3.3 Interface Relations

Let (x*, y*) be a point on the interface Γ. Assume that u(x,y) has second order derivative in the neighborhood of (x*, y*) corresponding to the local coordinates at (0, 0). Then from jump condition (3.2), we can immediately have

$$u^{+} = u^{-} + w. \tag{3.6}$$

Since (3.5), we can use the notation [u] = w(η) and $[u_n]$ = g(η) in local coordinate system. Differentiating (3.2) with respect to ξ along the interface, we get

$$u_\xi^+ = u_\xi^- + g\,. \tag{3.7}$$

Differentiating (3.2) with respect to η along the interface, we get

$$[u_\xi]\chi' + [u_\eta] = w'(\eta)\,. \tag{3.8}$$

Setting η=0, we get

$$u_\eta^+ = u_\eta^- + w'\,. \tag{3.9}$$

Differentiating (3.8), we obtain

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

Setting η=0, we get

$$u_{\eta\eta}^+ = u_{\eta\eta}^- + (u_\xi^- - u_\xi^+)\chi'' + w''\,. \tag{3.10}$$

In local coordinates, (3.3) can also be written as

$$u_\xi^+ - u_\eta^+\chi' = u_\xi^- - u_\eta^-\chi' + g\sqrt{1+(\chi')^2}\,. \tag{3.11}$$

Differentiating (3.11) with respect to η along with the interface, we have

$$\begin{aligned} u_{\xi\xi}^+\chi' + u_{\xi\eta}^+ - \frac{d}{d\eta}(u_\eta^+)\chi' - u_\eta^+\chi'' = {} & u_{\xi\xi}^-\chi' + u_{\xi\eta}^- - \frac{d}{d\eta}(u_\eta^-)\chi' - u_\eta^-\chi'' \\ & + g'(\eta)\left(\sqrt{1+(\chi')^2} + \frac{g(\eta)\chi'\chi''}{\sqrt{1+(\chi')^2}}\right). \end{aligned} \tag{3.12}$$

Setting η=0, we get

$$u_{\xi\eta}^+ = u_\xi^- - u_\xi^+ + (u_\eta^+ - u_\eta^-)\chi'' + u_{\xi\eta}^- + g'\,. \tag{3.13}$$

From (3.1a), we have

$$u^{+}_{\xi\xi} + u^{+}_{\eta\eta} - u^{-}_{\xi\xi} - u^{-}_{\eta\eta} + ku^{+} - ku^{-} = f^{+} - f^{-}.$$

Solving for $u_{\xi\xi}{}^{+}$, then

$$u^{+}_{\xi\xi} = u^{-}_{\xi\xi} + u^{-}_{\eta\eta} - u^{+}_{\eta\eta} - kw + [f]. \tag{3.14}$$

These interface relation are used in deriving the finite difference method in later section discussing correction term.

3.4 The Finite Difference Scheme of the IIM

At regular grid points (x_i, y_j), we can use the standard central five-point stencil finite difference schemes

$$\begin{aligned} u_{xx}(x_i, y_j) &\approx \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h_x}, \\ u_{yy}(x_i, y_j) &\approx \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{h_y}. \end{aligned} \tag{3.15}$$

If the solution is 2nd order continuous or higher, then it relatively easy to show that the locate truncation error in these points are O(h2) in previous chapter.

We now focus on irregular grid points, which the solutions are discontinuous across interface. Taking an irregular grid point (x_i, y_j), we try to develop the modified finite difference scheme like following

$$\sum_{k=1}^{k_s} \gamma_k u_{i+i_k, j+j_k} + ku_{ij} = f_{ij} + C_{ij}, \tag{3.16}$$

where the summation is take over k_s neighborhood points center at (x_i, y_j), and γs are the coefficients of the finite difference scheme. Our goal is to find proper coefficients γs and Cs, such that the finite difference scheme is still second order accurate

Generally speaking, the wave number k is usually constant. And we would like coefficient γs still keep as same as the ones in standard 5-point central finite difference scheme. That is, the coefficients are $\gamma_k = 1/h^2$ for the four neighbors of (x_i, y_j), and $-4/h^2$ for the master grid point (x_i, y_j).

3.5 Correction Terms at Irregular Grid Points

The Taylor expansion of $u(x_{i+ik}, y_{j+jk})$ about (x_i, y_j) under the local coordinates is

$$u(x_{i+i_k}, y_{j+j_k}) = u(\xi,\eta) = u^{\pm} + \xi_k u_{\xi}^{\pm} + \eta_k u_{\eta}^{\pm} + \frac{1}{2}\xi_k^2 u_{\xi\xi}^{\pm} + \xi_k\eta_k u_{\xi\eta}^{\pm} + \frac{1}{2}\eta_k^2 u_{\eta\eta}^{\pm} + O(h^3), \tag{3.17}$$

where the "+" or "–" sign is chosen depending on whether (ξk, ηk) lies on the "+" or "–" side of interface Γ. After the expansions of all terms, $u(x_{i+ik}, y_{j+jk})$, used in the finite difference equation (3.16), the local truncation error Tij can be espressed as a linear combination of the values $u^{\pm}$, $u_{\xi}^{\pm}$, $u_{\eta}^{\pm}$, $u_{\xi\xi}^{\pm}$, $u_{\xi\eta}^{\pm}$, $u_{\eta\eta}^{\pm}$ as following

$$
\begin{aligned}
T_{ij} &= \sum_{k=1}^{k_s} \gamma_k u(x_{i+i_k}, y_{j+j_k}) - ku(x_i, y_j) - f(x_i, y_j) - C_{ij} \\
&= a_1 u^- + a_2 u^+ + a_3 u_\xi^- + a_4 u_\xi^+ + a_5 u_\eta^- + a_6 u_\eta^+ \\
&\quad + a_7 u_{\xi\xi}^- + a_8 u_{\xi\xi}^+ + a_9 u_{\eta\eta}^- + a_{10} u_{\eta\eta}^+ + a_{11} u_{\xi\eta}^- + a_{12} u_{\xi\eta}^+ \\
&\quad - ku^- - f^- - C_{ij} + \max\{|\gamma_k|\} O(h^3).
\end{aligned}
\tag{3.18}
$$

The quantities $f^{\pm}$ are the limiting values of the function f at (x_i^*, y_j^*) from the "+" or "–" side of the interface. The coefficients $\{a_j\}$ depend only on the position of the stencil relative to the interface. They are independent of the PDE, u, k, f and the jump conditions w and g. If we define the index set K^+ and K^- by

$$
K^{\pm} = \{k : (\xi_k, \eta_k) \; is \quad on \quad the \quad "\pm" side \quad of \quad \Gamma\}.
$$

Then the {aj}'s are given by

$$
\begin{aligned}
a_1 &= \sum_{k \in K^-} \gamma_k, & a_2 &= \sum_{k \in K^+} \gamma_k, \\
a_3 &= \sum_{k \in K^-} \xi_k \gamma_k, & a_4 &= \sum_{k \in K^+} \xi_k \gamma_k, \\
a_5 &= \sum_{k \in K^-} \eta_k \gamma_k, & a_6 &= \sum_{k \in K^+} \eta_k \gamma_k, \\
a_7 &= \frac{1}{2} \sum_{k \in K^-} \xi_k^2 \gamma_k, & a_8 &= \frac{1}{2} \sum_{k \in K^+} \xi_k^2 \gamma_k, \\
a_9 &= \frac{1}{2} \sum_{k \in K^-} \eta_k^2 \gamma_k, & a_{10} &= \sum_{k \in K^+} \eta_k^2 \gamma_k, \\
a_{11} &= \sum_{k \in K^-} \xi_k \eta_k \gamma_k, & a_{12} &= \sum_{k \in K^+} \xi_k \eta_k \gamma_k.
\end{aligned}
\tag{3.19}
$$

Using the interface relations from (3.6) to (3.14), we can eliminate the quantities from one side, say "+" side, using the quantities from the other side, say "–" side, and combining likely terms and rearrange

$$
\begin{aligned}
T_{ij} &= (a_1 + a_2)u^- + (a_3 + a_4)u_\xi^- + (a_5 + a_6)u_\eta^- \\
&\quad + (a_7 + a_8 - 1)u_{\xi\xi}^- + (a_9 + a_{10} - 1)u_{\eta\eta}^- + (a_{11} + a_{12})u_{\xi\eta}^- \\
&\quad - ku^- - f^- + (\hat{T}_{ij} - C_{ij}) + \max\{|\gamma_k|\}O(h^3),
\end{aligned} \tag{3.20}
$$

where

$$
\begin{aligned}
\hat{T}_{ij} &= a_2 w + a_{12} g' + (a_6 + a_{12}\chi'')w' + a_{10} w'' \\
&\quad + (a_4 + a_8\chi'' - a_{10}\chi'')g + a_8([f] + kw - w'').
\end{aligned} \tag{3.21}
$$

Luckily, if we choose the coefficients γs as for standard 5-point central finite difference scheme, then following equations are satisfied

$$
\begin{aligned}
&a_1 + a_2 = 0, \\
&a_3 + a_4 = 0, \\
&a_5 + a_6 = 0, \\
&a_7 + a_8 = 1, \\
&a_9 + a_{10} = 1, \\
&a_{11} + a_{12} = 0.
\end{aligned} \tag{3.22}
$$

In addition, let

$$
\begin{aligned}
C_{ij} = \hat{T}_{ij} &= a_2 w + a_{12} g' + (a_6 + a_{12}\chi'')w' + a_{10} w'' \\
&\quad + (a_4 + a_8\chi'' - a_{10}\chi'')g + a_8([f] + kw - w''),
\end{aligned} \tag{3.23}
$$

here, C_{ij} depends on the curvature (w'') of the interface, which means it is difficult to get an analytic expression for the correction terms.

Thus, we have a method that the local truncation error at (x_i^*, y_j^*) in (3.18) is second order convergence guaranteed in theory. Later in our numerical experiments results, the global truncation errors are second order convergence under L^∞ norm.

Since the standard 5-point central finite difference scheme is used, and only the right-hand sides of the finite difference equations need to be modifies by adding a correction term. So the Fast Fourier Transformation solver as we described in previous chapter can be applied to solve the system of finite difference equations. This makes the Immersed Interface Method very efficient because the computational cost on the irregular points is relatively small. Further in next chapter, under the augmented strategy, this IIM method still can be used effectively even when one of the jump conditions [u] or $[u_n]$ is unknown.

CHAPTER 4

Augmented Strategies

In many interface problems, the jump conditions for the solution U and the derivative of the solution U_n in the interface are coupled together, and one of them are usually unknown.

We start with assume the unknown jump condition as some augmented variable g of codimension, and its discrete form G. The approximate solution U and the augmented variable G together form a large linear system representing the original problem, thus it was relatively easy to be understood but sometime too big to solved. Then, by eliminate U from the matrix vector equations, we try to solve for the augmented variable G using the Schur complement system, which is generally much smaller than that for U. GMRES iterative method is used first solving the original problem, with assumed initial augmented variable; then finding the residual of the constraint using the computed approximate solution given the augmented variable.

Augmented method do not required a Green's function, and no need to set up the system of equations; and it can be applied to general PDEs with or without source term. All boundary conditions shall be working just fine. Only high precision data type required when implementation Schur complement. The only way to derive an accurate algorithm in that kind of problem is perhaps the augmented approach.

The original idea of the augmented strategy for the interface problem is introduced in [21] to solve elliptic interface problems, and then further developed in [22] applied to generalized Helmholtz equation on irregular domains.

4.1 The Augmented Variable

In this dissertation, we only studied Dirichlet boundary condition. Other boundary condition such as Neumann and Robin, however, can be derived using same methodology.

Since Dirichlet boundary condition, we have already know the solution u at boundary, $u=w(x,y)$, $(x,y)\in\Gamma$. so it is natural to select the normal derivative $[u_n]$ as the augmented variable g. Further by discretization, we can write $[u]_\Gamma$ as $W=\{W_1, W_2, \dots W_{nb}\}$ and $[u_n]_\Gamma$ as $G=\{G_1, G_2, \dots G_{nb}\}$.

4.2 Discrete System of Equations in Matrix-Vector Form

From previous chapter, we knew that the correction term C_{ij} depends on $\{G_k\}$ and $\{W_k\}$ continuously. Then the Helmholtz equation (1.2) can be written as following discrete from

$$AU + B(W,G) = F\,, \tag{4.5}$$

where U and F are the vectors formed by $\{U_{ij}\}$ and $\{F_{ij}\}$. From (3.23), we knew that B(W,G) is a linear function of W and G, and can be written as

$$B(W,G) = BG - B_1W\ , \tag{4.6}$$

where B and B_1 are two matrices with entries. Thus (4.5) becomes

$$AU + BG = F + B_1W = F_1\,. \tag{4.7}$$

On the other hand, if the solution U from the system (4.7) is known, we can interpolate $\{U_{ij}\}$ linearly to get $\{U_n^{\pm}(X_k)\}$, which is an approximation to the normal derivative from each side of the interface at $\{X_k\}$, $1 \le k \le n_b$. The interpolation scheme is very important to the accuracy of our augmented algorithm, we will discuss it in more detail next section. Since the interpolation is linear, we can represent it as following

$$CU + DG = W \,, \tag{4.8}$$

where C and D are the linear interpolation scheme to approximate W.

Combining (4.7) and (4.8) together, we have

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ G \end{bmatrix} = \begin{bmatrix} F_1 \\ W \end{bmatrix}. \tag{4.9}$$

Remark: A, B, C and D represent the operation of their respect scheme, and they may not be written explicitly into matrix format.

4.3 Least Squared Interpolation

In this study, we involved the complex number least squared interpolation scheme from a Cartesian grid to form interface. It is almost identical to the work in real number [18]. The performing of this least square interpolation scheme is crucial to the accuracy and the iterations of the GMRES, and thus irreplaceable to whole augmented method.

The interpolation scheme for approximating U^-_n can be written as

$$\bar{U}_n(x^*, y^*) = \sum_{k=1}^{k_s} \gamma_k U_{i^*+i_k,\, j^*+j_k} - C \quad . \tag{4.10}$$

where k_s is the number of grid points involved in the interpolation scheme, (x_{i^*}, y_{j^*}) is the closest grid point to the projected interface point (x*, y*), C is the correction term and γ_k. is the coefficients for the interpolation. Note that γ_k and C are depend on (x*, y*). It is clear that we have to determine the coefficients $\{\gamma_k\}$ and C to complete the interpolation.

The coefficients $\{\gamma_k\}$ are determined by minimizing the interpolation error of (4.10) when $U_{i^*+ik,\ j^*+jk}$ is substituted with the exact solution $u(x_{i^*+ik},\ y_{j^*+jk})$. Using the local coordinates system

$$\begin{cases} \xi = (x - x^*)\cos\theta + (y - y^*)\sin\theta, \\ \eta = -(x - x^*)\sin\theta + (y - y^*)\cos\theta. \end{cases} \tag{4.11}$$

Centered at the point (x*, y*), and denoted the local coordinates of $(x_{i^*+ik},\ y_{j^*+jk})$ as $(\xi_k,\ \eta_k)$, we have the following from the Taylor expansion at (x*, y*) or (0,0) in the local coordinates:

$$\begin{aligned} & u(x_{i^*+i_k}, y_{j^*+j_k}) = u(\xi_k, \eta_k) \\ & = u^{\pm} + \xi_k u_{\xi}^{\pm} + \eta_k u_{\eta}^{\pm} + \frac{1}{2}\xi_k{}^2 u_{\xi\xi}^{\pm} + \xi_k \eta_k u_{\xi\eta}^{\pm} + \frac{1}{2}\eta_k{}^2 u_{\eta\eta}^{\pm} + O(h^3). \end{aligned} \tag{4.12}$$

where the "+" or "-" sign is chosen depending on whether $(\xi_k,\ \eta_k)$ lies on the "+" or "-" side of the interface Γ, and $u^{\pm}$, $u_{\xi}^{\pm}$, …, $u_{\eta\eta}^{\pm}$ are evaluated at local coordinates (0,0), or (x*,y*) in the original coordinates system. Be careful with these two coordinate system, they are confusing yet necessary for different computing function.

We carry out this expansion for all the grid points involved in the interpolation scheme and adding them together, that is plug all (4.12) (k=1, …, k_s) into (4.10). Afternoon

combining likely terms and re-arrange them, we have

$$\begin{aligned} U_n^-(x^*, y^*) \approx\ & a_1 u^- + a_2 u^+ + a_3 u_\xi^- + a_4 u_\xi^+ + a_5 u_\eta^- + a_6 u_\eta^+ \\ & + a_7 u_{\xi\xi}^- + a_8 u_{\xi\xi}^+ + a_9 u_{\eta\eta}^- + a_{10} u_{\eta\eta}^+ + a_{11} u_{\xi\eta}^- + a_{12} u_{\xi\eta}^+ - C, \end{aligned} \tag{4.13}$$

where the $\{a_i\}$ are defined as following

$$\begin{aligned} a_1 &= \sum_{k\in K^-} \gamma_k, & a_2 &= \sum_{k\in K^+} \gamma_k, \\ a_3 &= \sum_{k\in K^-} \xi_k \gamma_k, & a_4 &= \sum_{k\in K^+} \xi_k \gamma_k, \\ a_5 &= \sum_{k\in K^-} \eta_k \gamma_k, & a_6 &= \sum_{k\in K^+} \eta_k \gamma_k, \\ a_7 &= \frac{1}{2}\sum_{k\in K^-} \xi_k^2 \gamma_k, & a_8 &= \frac{1}{2}\sum_{k\in K^+} \xi_k^2 \gamma_k, \\ a_9 &= \frac{1}{2}\sum_{k\in K^-} \eta_k^2 \gamma_k, & a_{10} &= \sum_{k\in K^+} \eta_k^2 \gamma_k, \\ a_{11} &= \sum_{k\in K^-} \xi_k \eta_k \gamma_k, & a_{12} &= \sum_{k\in K^+} \xi_k \eta_k \gamma_k. \end{aligned} \tag{4.14}$$

Since $u^+ = u^- + w$ and $u_n^+ = u_n^- + g$. and the interface relations. We can express all the quantities from the "+" side in (4.13) in terms of those from the "-" side and the known quantities. Thus , when $U_{i^*+ik,\ j^*+jk}$ is substituted for the exact solution $u(x_{i^*+ik},\ y_{j^*+jk})$, (4.10) can be written as

$$\begin{aligned} U_n^-(x^*, y^*) &= \sum_{k=1}^{k_s} \gamma_k u(x_{i^*+i_k}, y_{j^*+j_k}) - C \\ &= a_1 u^- + a_2 u^+ + a_3 u_\xi^- + a_4 u_\xi^+ + a_5 u_\eta^- + a_6 u_\eta^+ + a_7 u_{\xi\xi}^- + a_8 u_{\xi\xi}^+ \\ &\quad + a_9 u_{\eta\eta}^- + a_{10} u_{\eta\eta}^+ + a_{11} u_{\xi\eta}^- + a_{12} u_{\xi\eta}^+ - C \\ &= (a_1 + a_2) u^- + (a_3 + a_4) u_\xi^- + (a_5 + a_6) u_\eta^- + (a_7 + a_8) u_{\xi\xi}^- \\ &\quad + (a_9 + a_{10}) u_{\eta\eta}^- + (a_{11} + a_{12}) u_{\xi\eta}^- + a_2[u] + a_4[u_\xi] + a_6[u_\eta] \\ &\quad + + a_8[u_{\xi\xi}] + a_{10}[u_{\xi\eta}] + a_{12}[u_{\eta\eta}] - C. \end{aligned}$$

To minimize the interpolation error, we should set the following linear system of equations for the coefficients $\{\gamma_k\}$ by matching the terms of u^-, u_ξ^-, …, $u_{\xi\eta}^-$:

$$
\begin{aligned}
&a_1 + a_2 = 0, \qquad && a_3 + a_4 = 1, \\
&a_3 + a_4 = 0, \qquad && a_7 + a_8 = 0, \\
&a_9 + a_{10} = 0, \qquad && a_{11} + a_{12} = 0.
\end{aligned} \tag{4.15}
$$

If the linear system (4.15) has a solution, then we can obtain a second-order interpolation scheme for the normal derivative u_n^- by choose an appropriate correction term C. From (4.12) and (4.15), we can see that the system of equations for the $\{\gamma_k\}$ is independent of the jump conditions which means we can calculate $\{\gamma_k\}$ outside of GMRES iteration.

In this study, we choose between 6 to 16 closest grid points to (x*,y*) as the interpolation stencil. If less than 6 different grid points ($k_s > 6$) in a neighborhood of (x*,y*) are used in the interpolation, we will have an under-determined system of linear equation system. If more than 16 points are chosen, then the computing cost will be very high without significant accuracy improvement.

We chose SVD method to solve (4.15). The SVD algorithm is very stable and can be found in many software packages, such as Linpack and Lapack. The SVD solution has the smallest 2-norm among all feasible solutions

$$\sum_{k=1}^{k_s} (\gamma_k^*)^2 = \min_{\gamma_k} \{ \sum_{k=1}^{k_s} (\gamma_k)^2 \} .$$

For such a solution, the magnitude of γ_k^* is well under control, which in important to the stability of the entire algorithm. Once the $\{\gamma_k\}$'s are computed, then the $\{a_k\}$'s can be obtained quickly. Now the correction term C is determined from following:

$$
\begin{aligned}
C = {} & a_2 w + a_4 g + a_6 w' + a_8 (g\chi'' - w'' + [f]) \\
& + a_{10} (w'' - g\chi'') + a_{12} (w'\chi'' + g').
\end{aligned} \tag{4.16}
$$

Further, through the relation $u_n^+ = u_n^- + g$, we can obtain

$$U_n^+(x^*, y^*) \cong \sum_{k=1}^{k_s} \gamma_k U_{i^*+i_k, j^*+j_k} - C + g \ . \tag{4.17}$$

In this study, the interface is represented by the zero set of a level set function, the least squares interpolation is used to approximate the surface derivatives using their values at the orthogonal projections of irregular grid points. The accurate with local support is second-order. It is robust in smooth error distribution. The trade-off is that we have to solve a an underdetermined $6\times k_s$ linear system of equations ($k_s \geq 6$), which is manageable.

4.4 Schur complement system

It is possible to directly solve the linear equation system in (4.9). However, by doing so, we have to deal with a matrix of $O((m \times n \times n_b)^2)$. This is way too much computing cost without an existing fast solver. The more efficient way is to using Schur complement and GMRES method, which we can also take advantage of new FFT passion solver we discussed in previous chapter.

By solving for U from (4.7), then substitute it back into (4.8), we can eliminating U from (4.9), and now have another much smaller $O((n_b)^2)$ linear system

$$(D - CA^{-1}B)G = W - CA^{-1}F_1 \ , \tag{4.18}$$

on the left hand side of the equation, $D\text{-}CA^{-1}B$ is the Schur complement of A. If the least square interpolation C is second order approximation to the continuous jump condition on the interface, then the Schur complement can be prove as invertible.

To solve (4.15), we first evaluate the right-hand side of the Schur complement. Set initial value G = **0**, then solve (4.5) or (4.7) to get U(**0**) which is $A^{-1}F_1$ from (4.7). Now the residual of the Schur complement for G=0 is

$$W - CA^{-1}F_1 = W - CU(0).$$

Second we evaluate the matrix-vector multiplication need by GMRES iteration. Solve the coupled system (4.7) to get U(G), interpolate U(G) as CU(G), then compute the residual as $R(G) = D - CU(G)$.

4.5 Solving by GMRES

Since the coefficient matrix of the Schur complement is no guarantee to be symmetric positive definite, the GMRES iterative method is preferred, because GMRES method requires only matrix-vector multiplication.

The Generalized Minimum RESidual (GMRES) method for real number was first proposed by Saad and Schultz [36] in 1986, later extended to complex number space by Fraysse and Giraud and Gratton [13] in 2005. It is among the most widely used Krylov solvers for the iterative solution of general large, sparse and non-symmetric (or non Hermitian) linear system. However, GMRES method in this study is still customized to fit our specific computational requirement.

The number of GMRES iteration depends on the condition number of the Schur complement, which seems to be proportional to 1/h. Therefore, the number of iteration will grow linearly as we decrease the mesh size.

To improve our computing performs, we use the weighted least squares interpolation (in previous section) to approximate u_n^+ or u_n^-, then use following formula respectively

$$u_n^- = -g + [u_n] \quad or \quad u_n^+ = -g + [u_n]$$

to force the solution to satisfy the flux jump condition, and then updated by the GMRES method. This is a precondition for the Schur complement system.

With this modification, the number of iteration for solving the Schur complement system seems be a constant and independent of the mesh size h. The numerical experiments have verified that conclusion. This precondition is an acceleration process and with no modification of the algorithm, and little computational cost.

CHAPTER 5

Numerical Experiments Results

We have performed a number of numerical experiments using Immersed Interfaced Method with Augmented Strategy. The results confirmed our analysis that the our method is second order convergence for Helmholtz / Poisson problem in complex number space with arbitrary interface.

All our computations were done using IBM Thinkpad R51 or HP Pavilion a706n Linux and later on Dell Latitude E6520 laptop computer. These are ordinary business or family computers which usually work with 2.1GHz CPU and 2GB memory, they are not very powerful machine. But the computing can be achieved in seconds or a few minutes, depends on the mesh size. For mesh size m=128 or less, the results are immediately printed out, and recorded CPU time is merely zero. For typical mesh size m=256, the computing can be finished in usually in 2 to 3 minutes. Once I had tried m=2048, and the computation took about 2 hours.

All project program are written/rewritten and complied using open source Fortran GNU 95 (http://www.g95.org). G95 is viewed as mildly fast. G95 has another advantage is its compatibility with other version, like legacy Fortran77, or propertied Intel's Visual Fortran. So we can use some existing packages like Linpack, FishPack without any modification. But the best advantage is its ability of cross-platforms, including Windows,

Unix/Linux and Mac. We can write the code once, and run it everywhere. G95 does not come with an Intergraded Developing Environment (IDE), and it relay on old-fashioned debug technical skill. However, Zeus (http://www.zeusedit.com) provides a free version editor that is very helpful.

The source codes are mostly upgraded from its real number IIM source code. Almost all the subroutines and functions have changed from double precision to double complex. The geometry functions and /or subroutines are modified to compatible with the sub-immersed boundary condition from original fully immersed interface condition. Complex number norm is created to replace the real number norm in the GMRES subroutine. The Fast Fourier Transformation solver for Poisson / Helmholtz problem is complete new method which is detailed described in Chapter Two.

The Error in the result table is defined as below using L^{∞} morn.

$$E_n = \left\| Error_n \right\|_{\infty} = \max_{\substack{i=0,\dots m \\ j=0,\dots n \\ (x_i, y_j)\in\Omega^-}} | e_{ij}^{(n)} | , where \quad e_{ij}^{(n)} = u_{ij}^{(n)} - u(x_i, y_j) , \tag{5.1}$$

where u_{ij} is the computed solution at (x_i, y_j), while $u(x_i, y_j)$ the exact solution, n is the mesh size. We also display the rate of two successive errors.

$$Rate = \log_2 \left(\frac{\| E_n \|_{\infty}}{\| E_{2n} \|_{\infty}} \right). \tag{5.2}$$

For a first order convergence method, the rate approaches to 1; and for a second order convergence method, the rate approaches to 2. The overall convergence order is displayed at

the bottom of the table, using least square linear regression for logarithm of error vs. logarithm of mesh size. The formula for the slope (noted as a) is following

$$a = \frac{n_s \sum (\log(E_n)\log(n)) - \sum \log(E_n) \sum \log(n)}{n_s \sum (\log(n))^2 - \left(\sum \log(n) \right)^2}, \tag{5.3}$$

where n_s is the total number of mesh size. The order of convergence is actually the negative slope of the linear regression model. It can be treated an overall rate for the entire test block. The complete linear regression model is also call called best fit line which can be written

$$\log_2(Error) = a \log_2(n) + b, \tag{5.4}$$

where a stated in (5.3) and b is the intercept of the linear model, and the formula is

$$b = \frac{\sum \log(E_n) \sum (\log(n))^2 - \sum \log(n) \sum \log(E_n)\log(n)}{n_s \sum (\log(n))^2 - \left(\sum \log(n) \right)^2}. \tag{5.5}$$

A hypothesis test (say, t-test) is performed for the convergence order / slope of the linear model of each test block at significant level of α. Here α is the probability of Type I error, when the Null Hypothesis is rejected but it is true in fact. This is to ensure that the method developed in this dissertation is truly second order convergence.

$$H_0 : a = -2(claim); \quad H_1 : a \neq -2 \qquad (two-tailed).$$

We choose α=0.05 (for strong evidence) or 0.01 (for weak evidence). The degree of freedom is n – 2 = 5 – 2 = 3. So, $t_{\alpha/2}$=3.182 when α=0.05; $t_{\alpha/2}$=5.841 when α=0.01. The confident interval is constructed as

$$\hat{a} - t_{\alpha/2} S_{est} \sqrt{\frac{1}{S_{xx}}} \le a \le \hat{a} + t_{\alpha/2} S_{est} \sqrt{\frac{1}{S_{xx}}}\,, \tag{5.6}$$

where $\hat{a}$ is the computed convergence order from observation, and a is "true" convergence order from our method, and S_{est} is the standard error of the estimate which is the standard deviation of the observed $\log_2(E_n)$ values about the predicted $\log_2(E_n')$ values

$$S_{est} = \sqrt{\frac{\sum (\log_2(E_n) - \log_2(E_n'))^2}{5-2}}\,. \tag{5.7}$$

And S_{xx} is the sum of squared difference for independent variables.

$$S_{xx} = \sum_{x_i} (x_i - \bar{x})^2 = \sum_{i=5}^{9} (i-7)^2 = 10\,, \tag{5.8}$$

where x_i is the value of logarithm of the mesh size, $\log_2(n)$, and n=32, 64, 128, 256 and 512.

If a = –2.00 fall between our confidence interval with $\alpha = 0.05$, then there is significant evidence not to reject the claim that this method is the second order convergence for this test block. If a = –2.00 fall between our confidence interval with $\alpha = 0.01$, then there is evidence to not to reject the claim that this method is second order convergence for this test block. If a = –2.00 fall outside the confidence interval with $\alpha = 0.01$, then there is significant evidence to reject the claim that this method is the second order convergence for this test block.

In addition, P-value is also computed through inversed t-distribution function with d.f. = 3 from t_{test} value

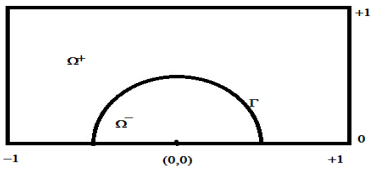
$$t_{test} = \frac{\hat{a} - (-2)}{S_{est} / \sqrt{S_{xx}}} . \tag{5.9}$$

We also conducted Pearson Product Moment Correlation Coefficient (PPMC) for the relationship in each block. This Correlation Coefficient is commonly known as r

$$r = \frac{n_s \sum \log(E_n) \log(n) - \sum \log(E_n) \sum \log(n)}{\sqrt{\left[n_s \sum (\log(n))^2 - \left(\sum \log(n) \right)^2 \right] \left[n_s \sum (\log(E_n))^2 - \left(\sum \log(E_n) \right)^2 \right]}} . \tag{5.6}$$

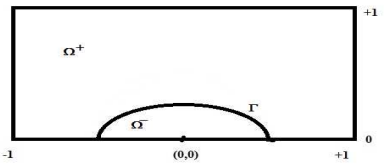
If the value of r is close to +1, then there is a strong positive linear relationship between the variables; If the value of r is close to −1, then there is a strong negative linear relationship between the variables; If the value of r is close to 0, then there is a no or weak linear relationship between the variables [3].

We use three different geometry boundaries as irregular interfaces in our numerical experiments. They are half circle, half oval and arbitrary half flower shape as following



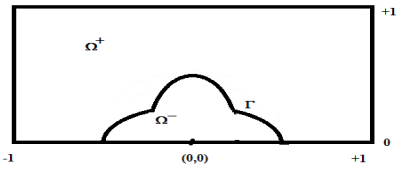
(a) $\Gamma = \{(x, y) : x^2 + y^2 = r^2\}$,

A Half Circle Interface,



(b) $, \Gamma = \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} = 1 \right\}$

A Half Oval Interface,



(c) $\Gamma = \left\{ (x, y) : \varphi_1 = \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2}; \quad \varphi_2 = \frac{x^2}{b_0^2} + \frac{y^2}{a_0^2}; \quad \min(\varphi_1, \varphi_2) = 0 \right\}$,

An Arbitrary Half Flower Petal Interface.

Figure 5.1 Domains and Interfaces.

5.1 Numerical Example 1

In this example, we consider a problem with a discontinuous source term f, and the differential equation is

$$u_{xx} + u_{yy} + ku = f, \qquad (x, y) \in \Omega, \tag{5.4}$$

with exact solution as

$$u = \begin{cases} x^2 + y^2 - 4, & (x, y) \in \Omega^-, \\ 0, & (x, y) \in \Omega^+. \end{cases} \tag{5.5}$$

and if k=0 then the source terms is

$$f = \begin{cases} 4, & (x, y) \in \Omega^-, \\ 0, & (x, y) \in \Omega^+. \end{cases} \tag{5.6}$$

The interface and the wave number k are given in each individual case. the Dirichlet boundary condition and jump condition $[u]_\Gamma$=w will be determined by the exact solution accordingly. Beware that the other jump condition $[u_n]_\Gamma$ = g is unknown

Case 1.1 Half Circular Interface

Let the domain Ω= {(x, y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and

Interface: $\Gamma := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} = 1, a_0 = 0.5, b_0 = 0.5 \right\}$,

Subdomain that is inside interface:

$$\Omega^- := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} \le 1, \quad a_0 = 0.5, \quad b_0 = 0.5, \quad -1 \le x \le 1, \quad 0 \le y \le 1 \right\}.$$

Subdomain that is outside interface: $\Omega^+ = \Omega / \Omega^-$.

Table 5.1a Error Analysis for Grid Refinement of Example 5.1.

	K=100i		k=100		k=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	4.6002536E-02		3.3543979E-01		7.2610372E-02	
64	1.0881148E-02	2.08	4.6449764E-02	2.85	1.6160096E-02	2.17
128	3.4165863E-03	1.67	1.5463574E-02	1.58	4.9224887E-03	1.71
256	8.4377786E-04	2.02	3.3379200E-03	2.21	1.2196654E-03	2.01
512	2.1049513E-04	2.00	8.2083288E-04	2.03	3.0374241E-04	2.01
Convergence Order	1.92		2.11		1.95	

Table 5.1b More Error Analysis for Grid Refinement of Example 5.1.

	k=0		k=10+10i		K=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	4.2429899E-02		7.0610581E-02		4.7572807E-02	
64	1.0470141E-02	2.02	1.7668004E-02	2.00	1.1739928E-02	2.02
128	2.6140307E-03	2.00	4.4181842E-03	2.00	2.9297816E-03	2.00
256	7.3682237E-04	1.83	1.1027800E-03	2.00	7.8627861E-04	1.90
512	1.8355658E-04	2.00	2.8224852E-04	1.97	1.9748188E-04	1.99
Convergence Order	1.95		1.99		1.97	

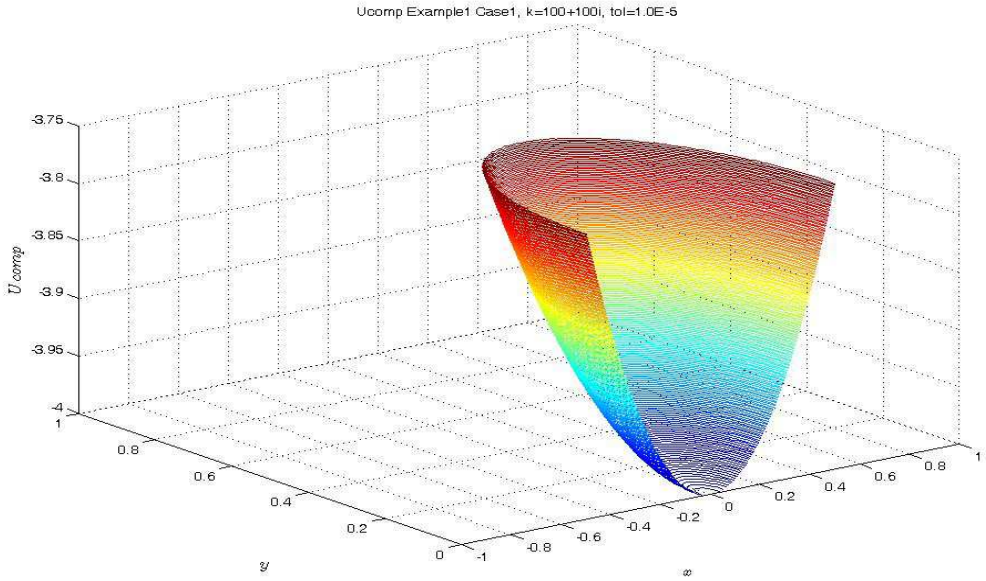


Figure 5.2 Computed Solution for Problem One
with A Half Circle Interface, k=100+100i, m=256, n=128.

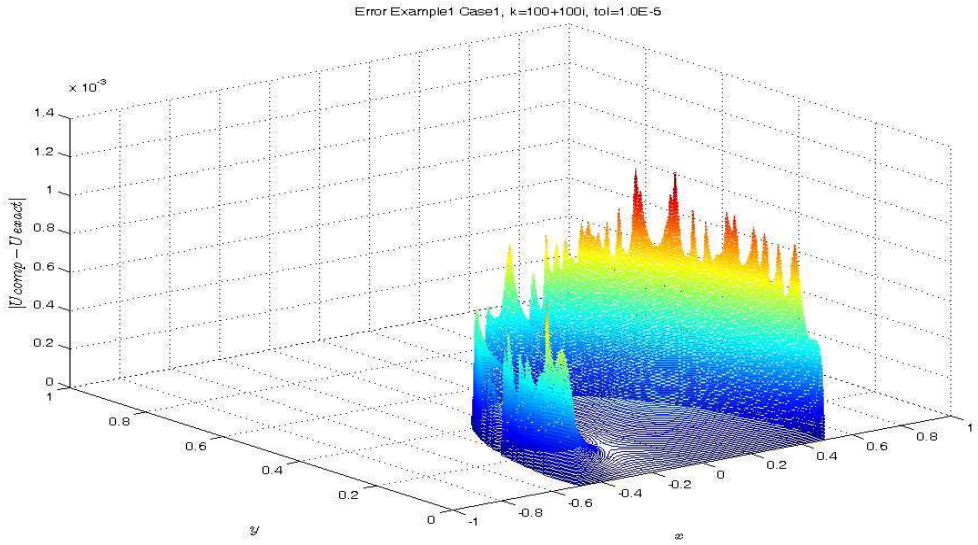


Figure 5.3 Error of Computed Solution for Problem One
with A Half Circle Interface, k=100+100i, m=256, n=128.

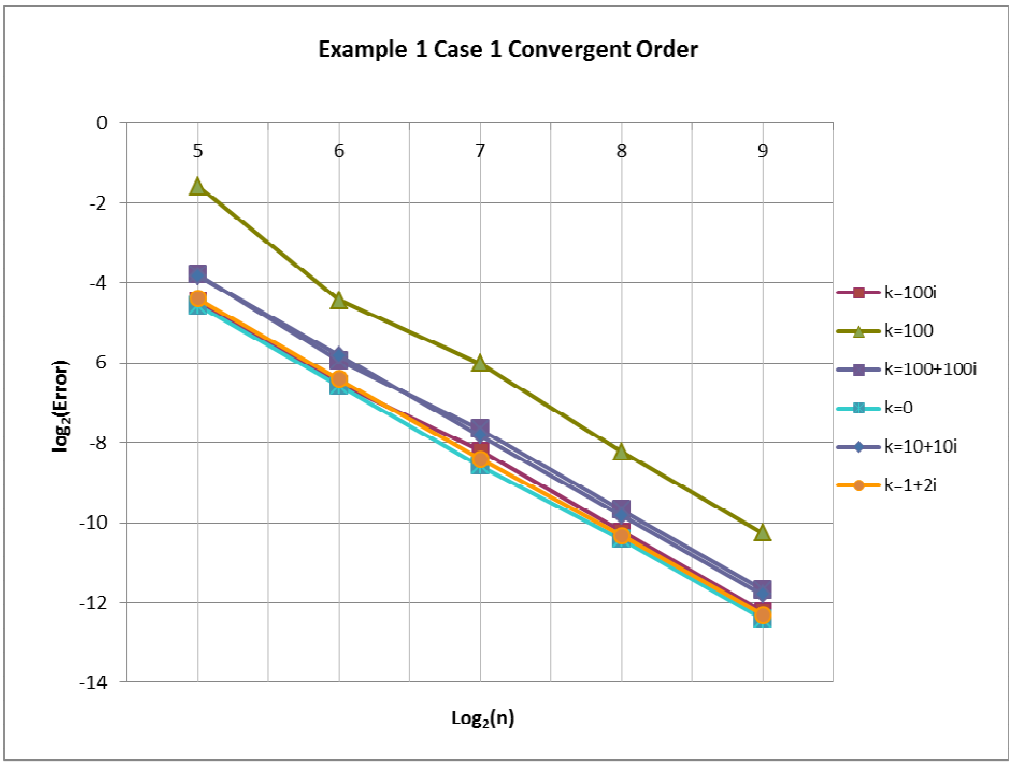


Figure 5.4 Convergence Order Comparision
for Problem One with a half circle interface.

Table 5.2 Convergence Analysis for Example 5.1.1.

	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
Coefficient Correlation	-0.9995	-0.9967	-0.9995	-0.9999	-1.0000	-0.9999
Slope	-1.9232	-2.1148	-1.9530	-1.9534	-1.9935	-1.9725
Intercept	5.1462	8.7044	5.9180	5.1672	6.1379	5.4394
S_{est}	0.1095	0.3144	0.1093	0.0591	0.0121	0.0373
Min 95% C.I.	-2.0334	-2.4312	-2.0630	-2.0129	-2.0057	-2.0100
Max 95% C.I.	-1.8130	-1.7984	-1.8431	-1.8940	-1.9813	-1.9349
P-value	0.1134	0.3318	0.2672	0.0882	0.1909	0.1019
Accept $H_0 : a = -2$	Yes	Yes	Yes	Yes	Yes	Yes

Case 1.2 Half Oval Interface

Let domain Ω= {(x,y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and

interface $\Gamma := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} = 1, a_0 = 0.5, b_0 = 0.25 \right\}$,

Subdomain that is inside the interface:

$$\Omega^- := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} \le 1, a_0 = 0.5, b_0 = 0.25, -1 \le x \le 1, 0 \le y \le 1 \right\}.$$

Subdomain that is outside the interface $\Omega^+ = \Omega / \Omega^-$.

Table 5.3a Error Analysis For Grid Refinement of Example 5.1.2.

	k=100i		k=100		k=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	3.8657325E-02		4.4417541E-02		6.0811784E-02	
64	8.7199213E-03	2.15	1.1272433E-02	1.98	1.3027898E-02	2.22
128	2.6332450E-03	1.73	3.4108313E-03	1.72	3.9079273E-03	1.74
256	8.2955764E-04	1.66	9.1640248E-04	1.90	1.1840489E-03	1.72
512	2.2077671E-04	1.91	2.4399532E-04	1.91	3.1353999E-04	1.92
Convergence Order	1.83		1.86		1.87	

Table 5.3b More Error Analysis For Grid Refinement of Example 5.1.2.

	k=0		K=10+10i		k=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	4.2351202E-02		6.4132756E-02		4.9644458E-02	
64	1.0529019E-02	2.01	1.6197542E-02	1.99	1.2348346E-02	2.01
128	2.6216007E-03	2.01	4.0506224E-03	2.00	3.0729950E-03	2.01
256	6.5443496E-04	2.00	1.0100359E-03	2.00	7.6676177E-04	2.00
512	1.6352861E-04	2.00	2.5205691E-04	2.00	1.9147906E-04	2.00
Convergence Order	2.00		2.00		2.00	

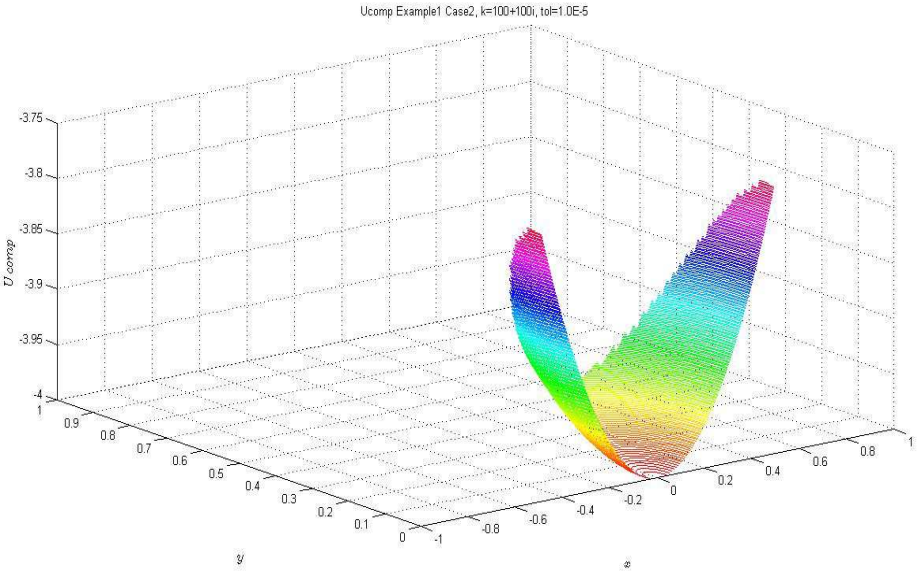


Figure 5.5 Computed Solution for Problem One
with A Half Oval Interface, k=100+100i, m=256, n=128.

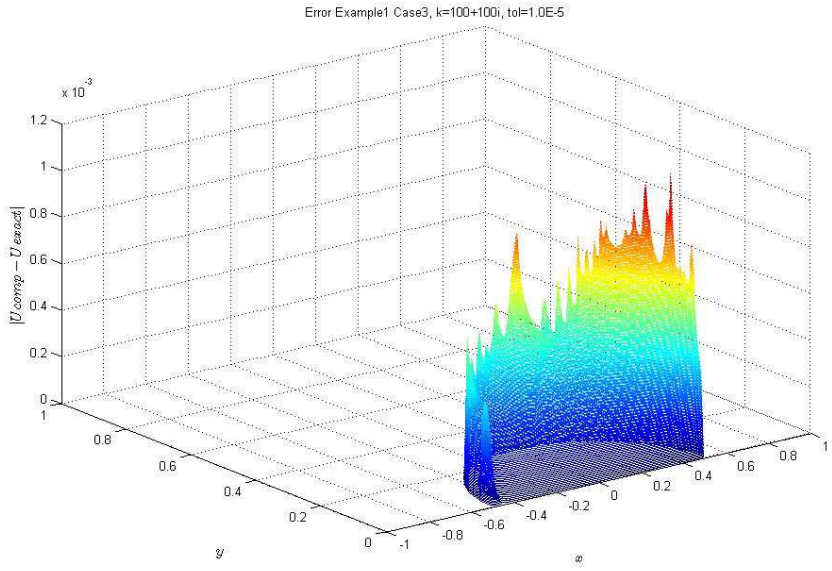


Figure 5.6 Error of Computed Solution for Problem One
with A Half Oval Interface, k=100+100i, m=256, n=128.

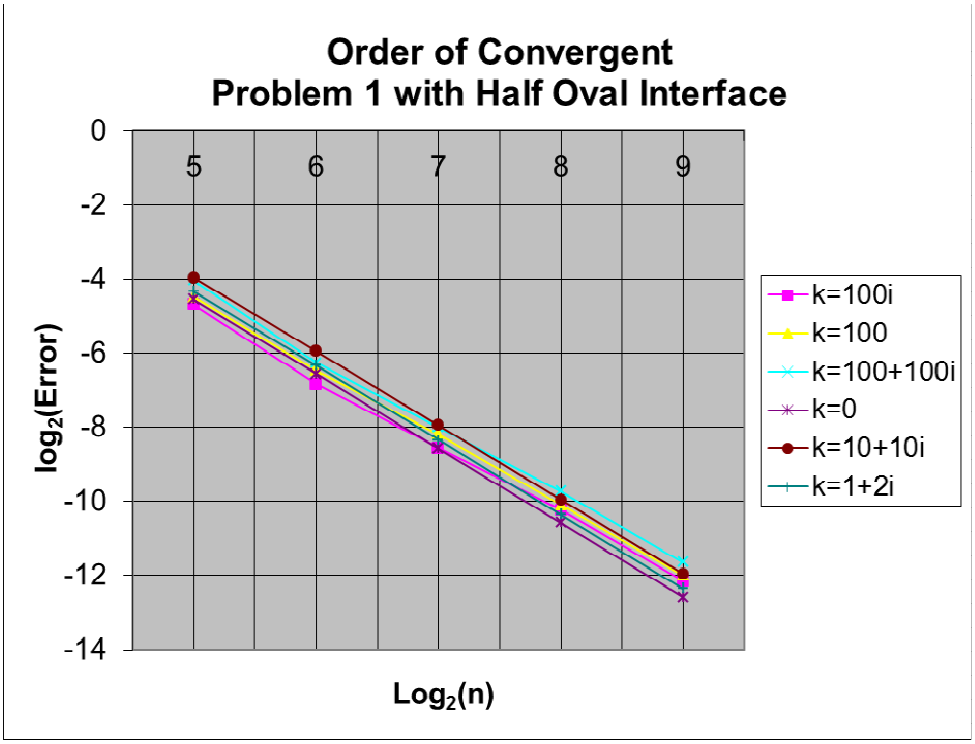


Figure 5.7 Convergence Order Comparison for Problem One with A Half Oval Interface.

Table 5.4 Convergence Analysis for Example 5.1.2.

	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
Coefficient Correlation	-0.9981	-0.9997	-0.9979	-1.0000	-1.0000	-1.0000
Slope	-1.8354	-1.8522	-1.8785	-2.0054	-1.9965	-2.0057
Intercept	6.1810	6.5784	7.0828	7.4695	8.0215	7.7008
Sest	0.1796	0.0733	0.1919	0.0021	0.0067	0.0017
Min 95% C.I.	-2.0162	*-1.9259**	-2.0715	*-2.0075**	-2.0033	*-2.0074**
Max 95% C.I.	-1.6547	*-1.7785**	-1.6854	*-2.0033**	-1.9898	*-2.0040**
P-value	0.0626	0.0078	0.1389	0.0039	0.2009	0.0017
Accept $H_0 : a = -2$	Yes	No	Yes	No	Yes	N0

*: a=–2 is fall outside of the confident interval with α=0.01.

Case 1.3 Arbitrary Half Flower Petal Interface

Let domain Ω= {(x,y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and the interface

$$\Gamma := \left\{ \begin{array}{c} (x,y) : \varphi_1(x,y) = \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} - 1, \quad \varphi_2(x,y) = \frac{x^2}{b_0^2} + \frac{y^2}{a_0^2} - 1, \quad a_0 = 0.5, \quad b_0 = 0.25, \\ \varphi(x,y) = \min\{\varphi_1, \varphi_2\} = 0, \quad -1 \le x \le 1, \quad 0 \le y \le 1 \end{array} \right\},$$

So, the subdomain that is inside the interface

$$\Omega^- := \left\{ \begin{array}{c} (x,y) : \varphi_1 = \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} - 1, \quad \varphi_2 = \frac{x^2}{b_0^2} + \frac{y^2}{a_0^2} - 1, \quad a_0 = 0.5, \quad b_0 = 0.25, \\ . \qquad \varphi(x,y) = \min\{\varphi_1, \varphi_2\} \le 0, \quad -1 \le x \le 1, \quad 0 \le y \le 1. \end{array} \right\}.$$

The subdomain that is outside the interface $\Omega^+ = \Omega / \Omega^-$.

Table 5.5a Error Analysis for Grid Refinement of Example 5.1.3.

	K=100i		K=100		K=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	5.1913879E-02		4.1377305E-01		8.4508015E-02	
64	1.0052570E-02	2.37	8.2334070E-02	2.33	1.4939966E-02	2.50
128	3.0053951E-03	1.74	2.3477317E-02	1.81	4.3482050E-03	1.78
256	8.3011452E-04	1.86	7.2167304E-03	1.70	1.1887442E-03	1.87
512	2.2101530E-04	1.91	1.5688298E-03	2.20	3.1446435E-04	1.92
Convergence Order	1.93		1.96		1.98	

Table 5.5b More Error Analysis for Grid Refinement of Example 5.1.3.

	K=0		K=10+10i		K=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	5.3374629E-02		7.0759979E-02		5.9816160E-02	
64	1.2864436E-02	2.05	2.2450431E-02	1.66	1.4717447E-02	2.02
128	4.8032416E-03	1.42	7.2590087E-03	1.63	5.3513614E-03	1.46
256	6.5180997E-04	2.88	1.1405516E-03	2.67	7.3578566E-04	2.86
512	4.5682590E-04	0.52	6.5899523E-04	0.79	5.0810818E-04	0.54
Convergence Order	1.80		1.78		1.81	

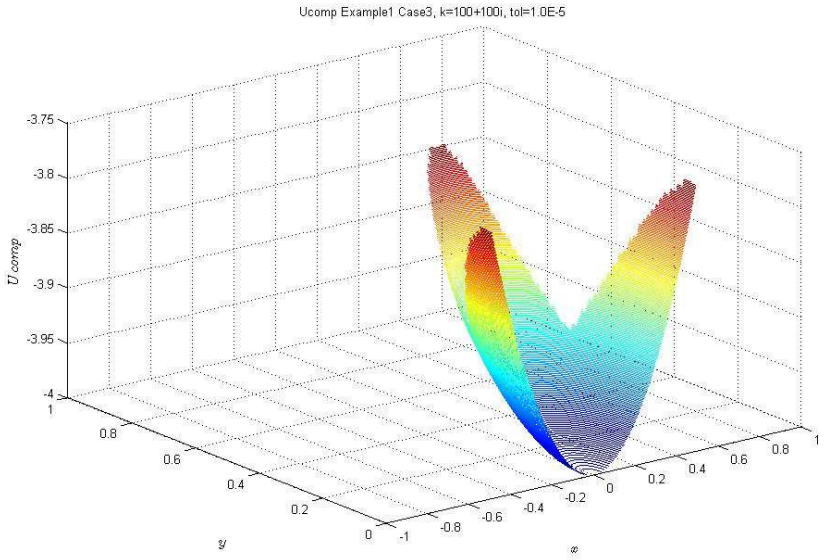


Figure 5.8. Computed Solution for Problem One
with An Arbitrary Flower Petal Interface, k=100+100i, m=256, n=128.

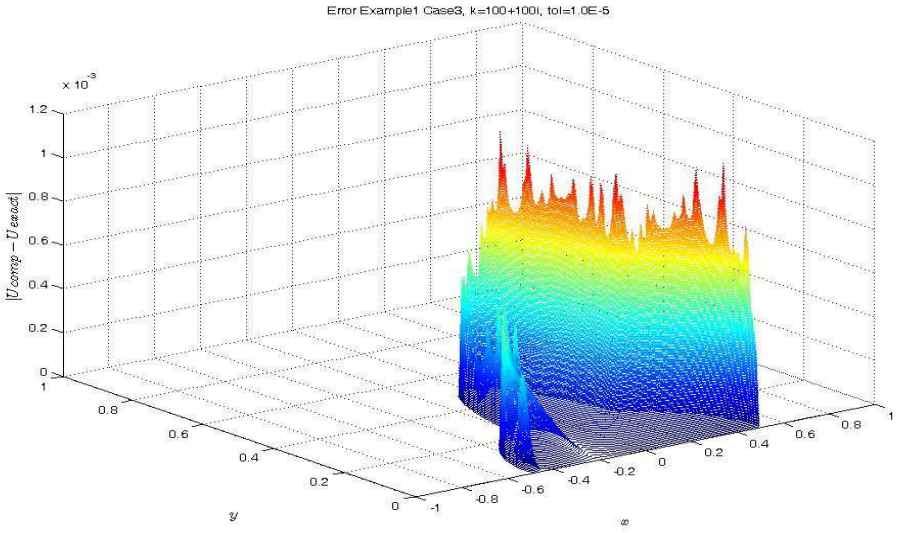


Figure 5.9. Error of Computed Solution for Problem One
with An Arbitrary Flower Petal Interface, k=100+100i, m=256, n=128.

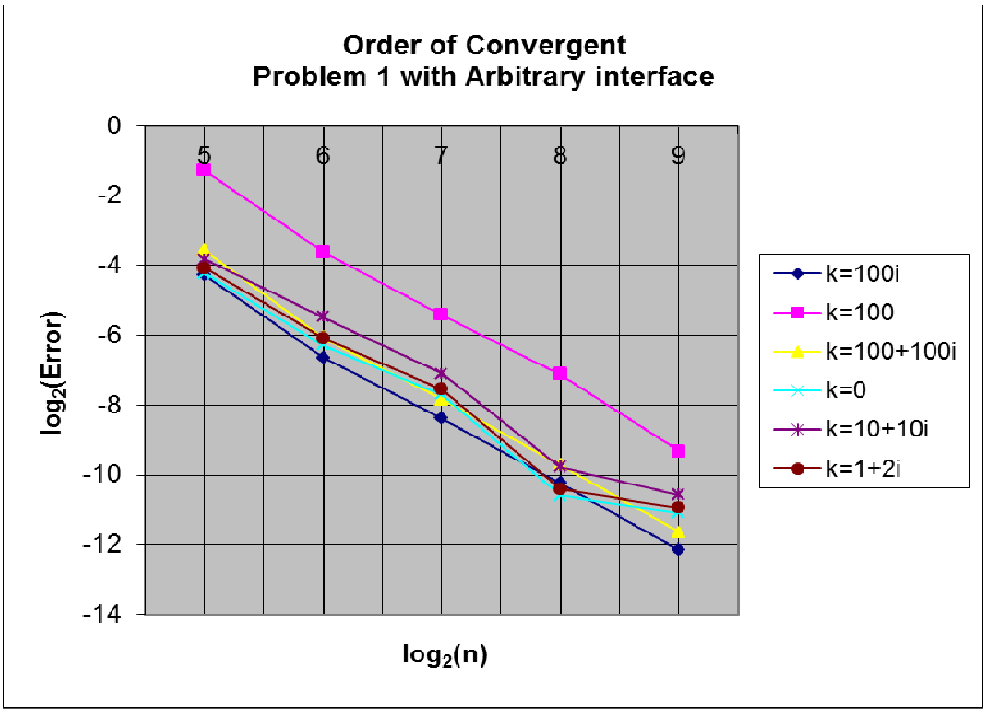


Figure 5.10 Convergence Order Comparison for Problem One with An Arbitrary Flower Petal Interface.

Table 5.6 Convergence Analysis for Example 5.1.3.

	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
Coefficient Correlation	-0.9976	-0.9972	-0.9968	-0.9908	-0.9918	-0.9913
Slope	-1.9642	-1.9334	-2.0235	-2.0488	-1.9494	-2.0495
Intercept	7.3523	10.1501	8.3788	8.1677	8.0757	8.3452
Sest	0.2157	0.2311	0.2566	0.4418	0.3963	0.4299
Min 95% C.I.	-2.1813	-2.1660	-2.2817	-2.4934	-2.3482	-2.4820
Max 95% C.I.	-1.7471	-1.7008	-1.7654	-1.6042	-1.5507	-1.6169
P-value	0.6360	0.4296	0.7906	0.7500	0.7136	0.7400
Accept $H_0 : a = -2$	Yes	Yes	Yes	Yes	Yes	Yes

Table 5.7 Computing Cost Analysis: Number of Iteration for Example 5.1.1.

Mesh	k=100i	k=100	k=100+100i	k=0	k=10+10i	K=1+2i
32	15	10	11	8	10	9
64	13	10	10	8	10	9
128	13	10	11	8	10	9
256	13	10	11	8	10	9
512	13	10	11	8	9	9

Table 5.8 Computing Cost Analysis: Number of Iteration for Example 5.1.2.

Mesh	k=100i	k=100	k=100+100i	k=0	K=10+10i	k=1+2i
32	10	10	11	8	10	8
64	10	12	11	9	10	9
128	10	12	10	10	11	10
256	9	12	10	9	11	9
512	9	12	10	9	11	9

Table 5.9 Computing Cost Analysis: Number of Iteration for Example 5.1.3.

Mesh	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
32	11	12	13	11	12	11
64	10	12	11	11	12	11
128	16	17	17	17	16	16
256	11	14	12	12	13	12
512	18	17	19	17	20	19

5.2 Numerical Example 2

In this example, we consider a problem with a discontinuous source term f, and the differential equation is

$$u_{xx} + u_{yy} + ku = f, \qquad (x, y) \in \Omega, \tag{5.4}$$

with exact solution as

$$u = \begin{cases} e^x \cos y, & (x, y) \in \Omega^-, \\ 0, & (x, y) \in \Omega^+. \end{cases} \tag{5.8}$$

and if k=0 (Poisson equation), then the source terms is

$$f = 0, \quad (x, y) \in \Omega. \tag{5.9}$$

The interface and the wave number k are given in each individual case, the Dirichlet boundary condition and jump condition $[u]_\Gamma$=w will be determined accordingly. Beware that the other jump condition $[un]_\Gamma$=g is unknown

Case 2.1 Half Circular Interface

Let the domain Ω= {(x, y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and

Interface: $\Gamma := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} = 1, a_0 = 0.5, b_0 = 0.5 \right\}$.

Subdomain that is inside interface:

$$\Omega^- := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} \le 1, \quad a_0 = 0.5, \quad b_0 = 0.5, \quad -1 \le x \le 1, \quad 0 \le y \le 1 \right\}.$$

Subdomain that is outside interface: $\Omega^+ = \Omega / \Omega^-$.

Table 5.10a Error Analysis for Grid Refinement of Example 5.2.1.

	K=100i		k=100		k=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	1.9172309E-02		1.3101805E-01		2.8496815E-02	
64	4.9807631E-03	1.94	2.1360192E-02	2.62	7.1232807E-03	2.00
128	1.3884691E-03	1.84	6.3771377E-03	1.74	1.9176186E-03	1.89
256	3.4969866E-04	1.99	1.4476327E-03	2.14	4.7597509E-04	2.01
512	8.8140329E-05	1.99	3.6014252E-04	2.01	1.1939369E-04	2.00
Convergence Order	1.94		2.09		1.97	

Table 5.10b More Error Analysis for Grid Refinement of Example 5.2.1.

	k=0		K=10+10i		k=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	4.2454093E-02		4.2454093E-02		4.7394051E-02	
64	1.0469232E-02	2.02	1.0469232E-02	2.02	1.1685877E-02	2.02
128	2.6134649E-03	2.00	2.6134649E-03	2.00	2.9153959E-03	2.00
256	7.3696011E-04	1.83	7.3696011E-04	1.83	7.7192953E-04	1.92
512	1.8364497E-04	2.00	1.8364497E-04	2.00	1.9422000E-04	1.99
Convergence Order	1.95		1.95		1.98	

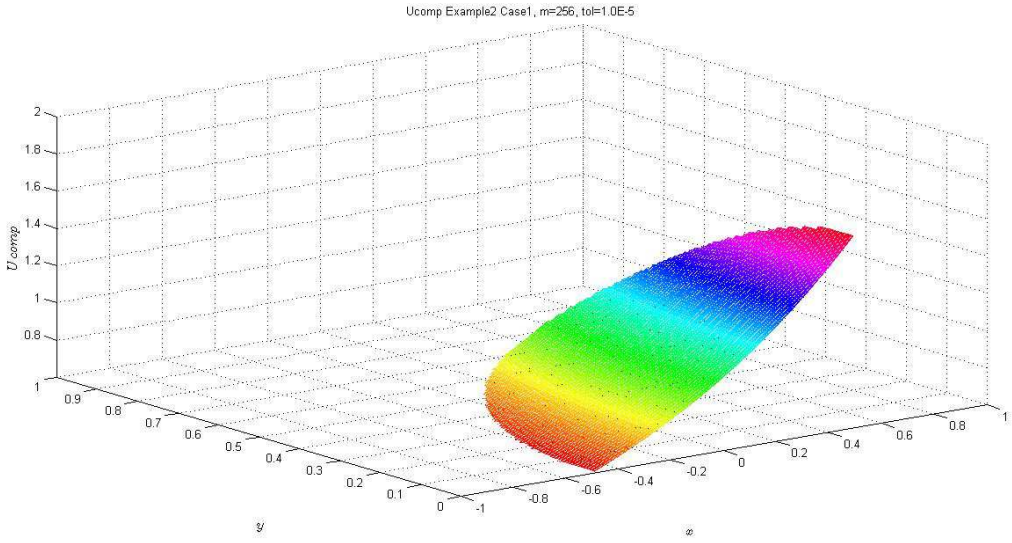


Figure 5.11 Computed Solution for Problem Two
with A Half Circle Interface, k=100+100i, m=256, n=128.

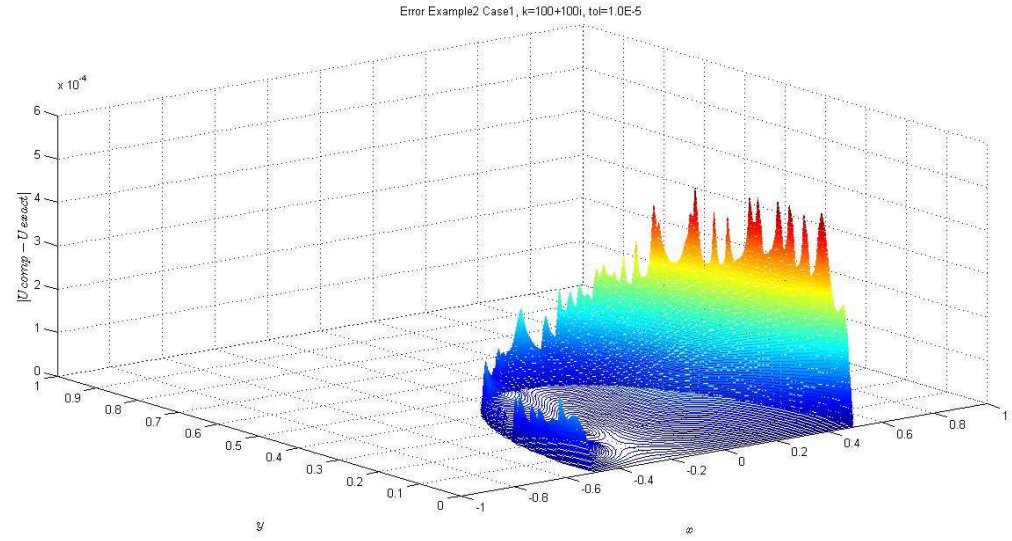


Figure 5.12. Error of Computed Solution Problem Two
with A Half Circle Interface, k=100+100i, m=256, n=128.

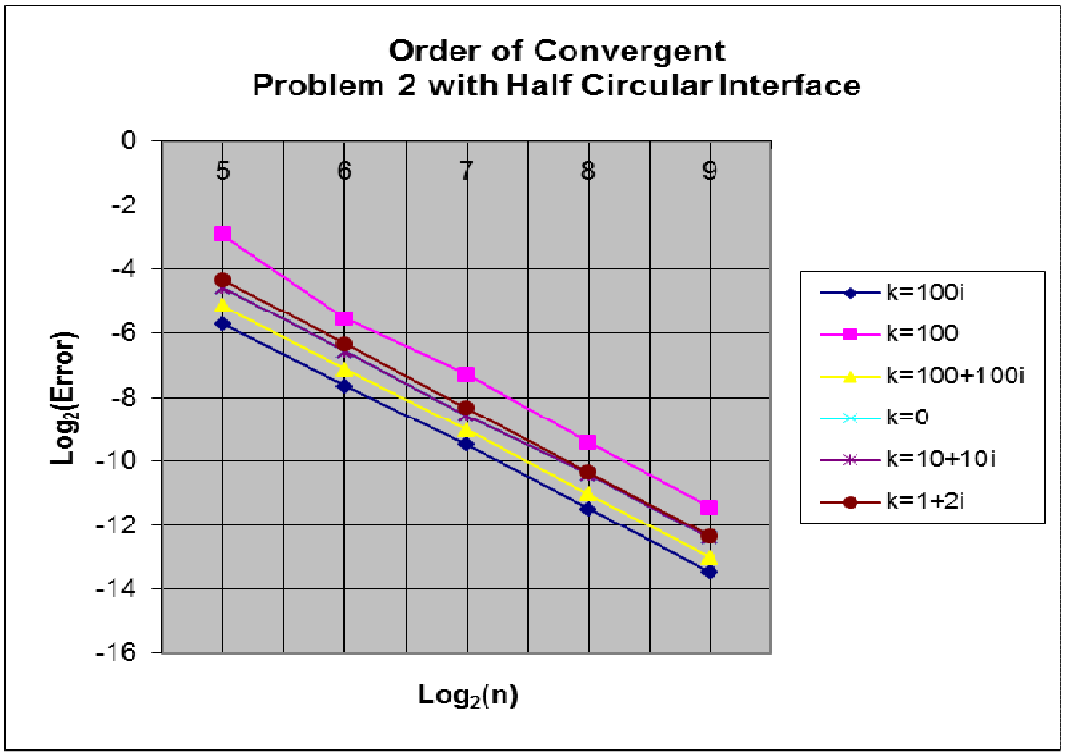


Figure 5.13 Convergence Order Comparison for Problem Two with An Half Circle Interface.

Table 5.11 Convergence Analysis for Example 5.2.1.

	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
Coefficient Correlation	-0.9999	-0.9983	-1.0000	-0.9999	-0.9999	-1.0000
Slope	-1.9362	-2.0897	-1.9701	-1.9534	-1.9534	-1.9782
Intercept	3.9939	7.2989	4.7187	5.1674	5.1674	5.4654
Sest	0.0488	0.2232	0.0340	0.0595	0.0595	0.0319
Min 95% C.I.	*-1.9853**	-2.3143	-2.0044	-2.0133	-2.0133	-2.0103
Max 95% C.I.	*-1.8871**	-1.8651	-1.9359	-1.8935	-1.8935	-1.9461
P-value	0.0257	0.2933	0.0694	0.0896	0.0896	0.1192
Accept $H_0 : a = -2$	Maybe	Yes	Yes	Yes	Yes	Yes

*: a=–2 falls between 99% confident Interval [-2.0264,-1.8460].

Case 2.2 Half Oval Interface

Let domain Ω= {(x,y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and interface

$$\Gamma := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} = 1, a_0 = 0.5, b_0 = 0.25 \right\}.$$

Subdomain that is inside the interface:

$$\Omega^- := \left\{ (x, y) : \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} \le 1, a_0 = 0.5, b_0 = 0.25, -1 \le x \le 1, 0 \le y \le 1 \right\}.$$

Subdomain that is outside the interface $\Omega^+ = \Omega / \Omega^-$.

Table 5.12a Error Analysis for Grid Refinement of Example 5.2.2.

	k=100i		k=100		k=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	1.3909343E-02		1.3494268E-02		2.1417184E-02	
64	3.5219551E-03	1.98	3.4273603E-03	1.98	5.1010118E-03	2.07
128	1.1136624E-03	1.66	1.0692560E-03	1.68	1.5781984E-03	1.69
256	3.9142595E-04	1.51	2.7907968E-04	1.94	5.4499299E-04	1.54
512	9.8978365E-05	1.98	7.4918455E-05	1.90	1.3846264E-04	1.98
Convergence Order	1.74		1.86		1.78	

Table 5.12b More Error Analysis for Grid Refinement of Example 5.2.2.

	K=0		k=10+10i		K=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	4.2345275E-02		6.3370572E-02		4.9557268E-02	
64	1.0527419E-02	2.01	1.6014427E-02	1.99	1.2323084E-02	2.01
128	2.6216757E-03	2.01	4.0061370E-03	2.00	3.0667723E-03	2.01
256	6.5430825E-04	2.00	9.9966997E-04	2.00	7.6546386E-04	2.00
512	1.6348735E-04	2.00	2.4947955E-04	2.00	1.9119254E-04	2.00
Convergence Order	2.00		2.00		2.00	

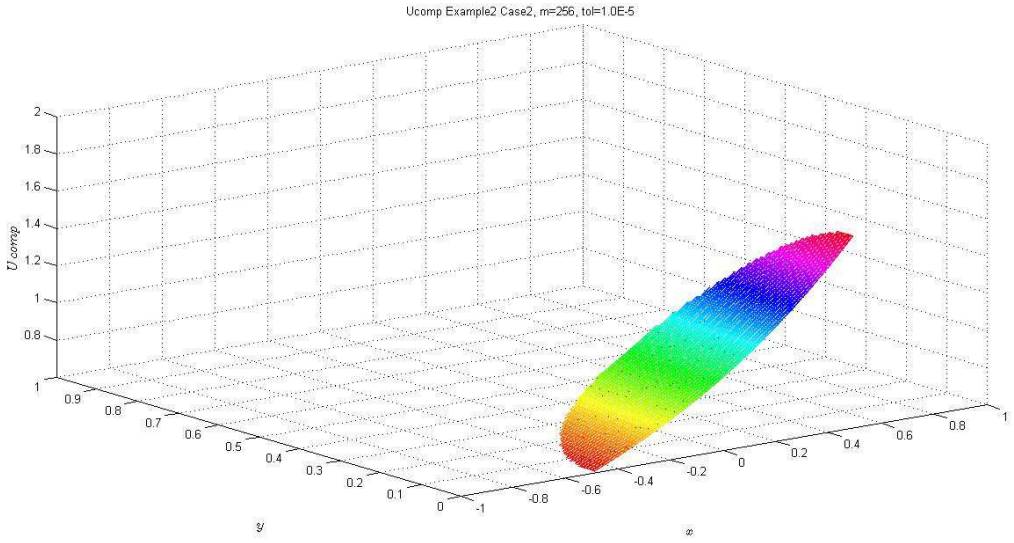


Figure 5.14. Computed Solution for Problem Two
with A Half Oval Interface, k=100+100i, m=256,n=128.

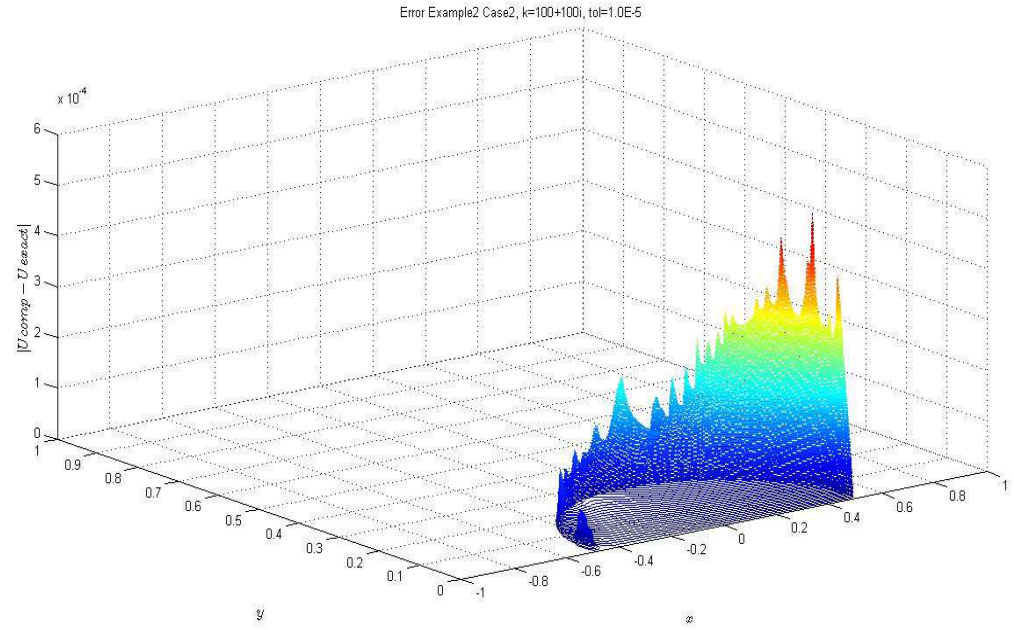


Figure 5.15. Error of Computed Solution for Problem Two
with A Half Oval Interface, k=100+100i, m=256, n=128.

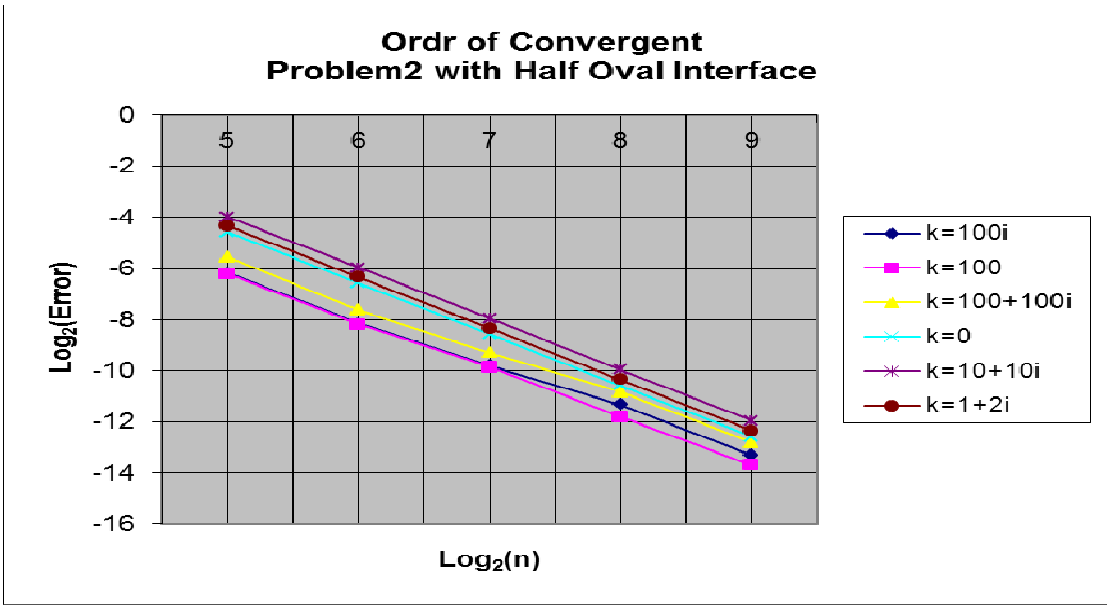


Figure 5.16 Convergence Order Comparison for Problem Two with A Half Oval Interface.

Table 5.13 Convergence Analysis for Example 5.2.2.

	k=100i	k=100	k=100+100i	k=0	K=10+10i	k=1+2i
Coefficient Correlation	-0.9989	-0.9997	-0.9988	-1.0000	-1.0000	-1.0000
Slope	-1.7439	-1.8604	-1.7773	-2.0042	-1.9979	-2.0045
Intercept	2.4575	3.0666	3.2155	5.4567	6.0169	5.6852
Sest	0.1504	0.0766	0.1611	0.0027	0.0067	0.0027
Min 95% C.I.	*-1.8953**	*-1.9375***	*-1.9394****	-2.0069#	-2.0047	-2.0072##
Max 95% C.I.	*-1.5925**	*-1.7833***	*-1.6152****	-2.0014#	-1.9912	-2.0018##
P-value	0.0126	0.0104	0.0221	0.0169	0.4016	0.0132
Accept $H_0 : a = -2$	Maybe	Maybe	Maybe	Yes	Yes	Maybe

*: a=–2 fall between 99% Confident Interval [-2.0218,-1.4660].
**: a=–2 fall between 99% Confident Interval [-2.0019, -1.7189].
***: a=–2 fall between 99% Confident Interval [-2.0748, -1.4797].
#: a=–2 fall between 99% Confident Interval [-2.0092, -1.9991].
##: a=–2 fall between 99% Confident Interval [-2.0094, -1.9995].

Case 2.3 Arbitrary Half Flower Petal Interface

Let domain Ω= {(x,y): $-1 \le x \le 1$ and $0 \le y \le 1$ }, and the interface

$$\Gamma := \left\{ \begin{array}{c} (x,y) : \varphi_1(x,y) = \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} - 1, \quad \varphi_2(x,y) = \frac{x^2}{b_0^2} + \frac{y^2}{a_0^2} - 1, \quad a_0 = 0.5, \quad b_0 = 0.25, \\ \varphi(x,y) = \min\{\varphi_1, \varphi_2\} = 0, \quad -1 \le x \le 1, \quad 0 \le y \le 1. \end{array} \right\},$$

So, the subdomain that is inside the interface

$$\Omega^- := \left\{ \begin{array}{c} (x,y) : \varphi_1 = \frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} - 1, \quad \varphi_2 = \frac{x^2}{b_0^2} + \frac{y^2}{a_0^2} - 1, \quad a_0 = 0.5, \quad b_0 = 0.25, \\ \varphi(x,y) = \min\{\varphi_1, \varphi_2\} \le 0, \quad -1 \le x \le 1, \quad 0 \le y \le 1. \end{array} \right\}.$$

The subdomain that is outside the interface $\Omega^+ = \Omega / \Omega^-$.

Table 5.14a Error Analysis for Grid Refinement of Example 5.2.3.

	K=100i		k=100		K=100+100i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	1.7045674E-02		1.1645818E-01		2.6767626E-02	
64	3.5129496E-03	2.28	4.1921111E-02	1.48	4.9592515E-03	2.43
128	1.1078943E-03	1.66	1.5779112E-02	1.41	1.5496942E-03	1.68
256	3.9008112E-04	1.51	3.1673795E-03	2.32	5.4311581E-04	1.51
512	9.8402477E-05	1.99	8.5726590E-04	1.88	1.3750733E-04	1.98
Convergence Order	1.80		1.79		1.84	

Table 5.14b More Error Analysis for Grid Refinement of Example 5.2.3.

	k=0		K=10+10i		k=1+2i	
Mesh	Error	Rate	Error	Rate	Error	Rate
32	5.2950678E-02		7.0444768E-02		5.9067049E-02	
64	1.2893373E-02	2.04	2.1849600E-02	1.69	1.4873880E-02	1.99
128	4.8007356E-03	1.43	7.1862374E-03	1.60	5.2856641E-03	1.49
256	6.5230164E-04	2.88	1.0328548E-03	2.80	7.3415854E-04	2.85
512	4.5758565E-04	0.52	6.6852508E-04	0.62	5.0220057E-04	0.55
Convergence Order	1.80		1.78		1.81	

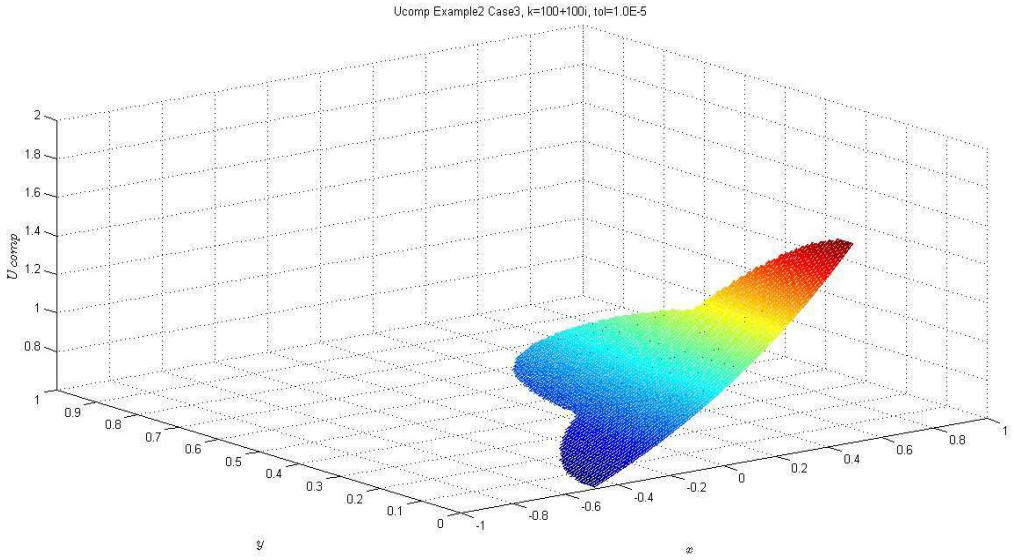


Figure 5.17. Computed Solution for Problem Two
with An Arbitrary Flower Petal Interface k=100+100i, m=256, n=128.

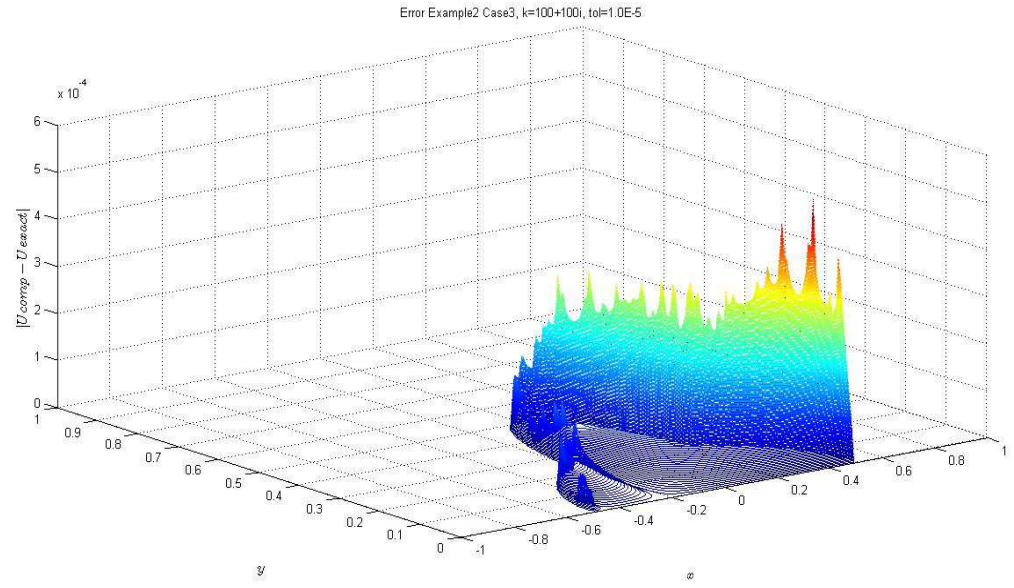
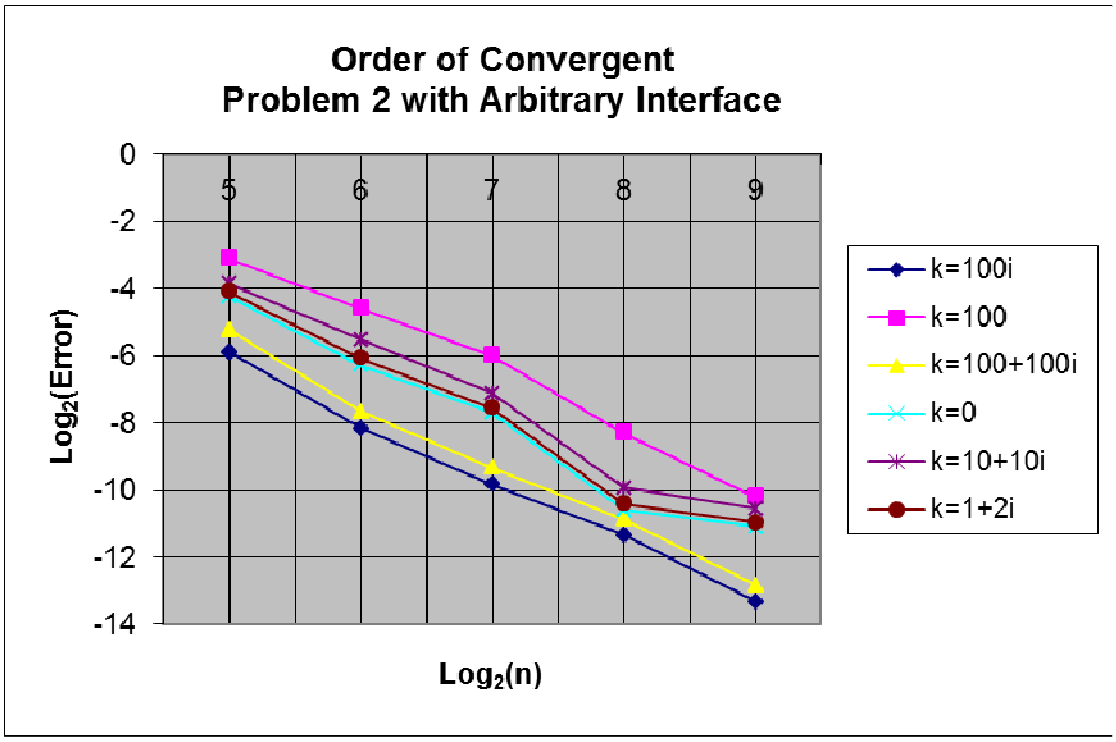


Figure 5.18. Error of Computed Solution for Problem Two
with An Arbitrary Flower Petal Interface k=100+100i, m=256, n=128.



5.19 Convergence Order Comparison for Problem Two
with An Arbitrary Flower Petal Interface.

Table 5.15 Convergence Analysis for Example 5.2.3.

	k=100i	k=100	k=100+100i	k=0	k=10+10i	k=1+2i
Coefficient Correlation	-0.9976	-0.9953	-0.9966	-0.9851	-0.9876	-0.9862
Slope	-1.8044	-1.7898	-1.8400	-1.8014	-1.7842	-1.8096
Intercept	2.9346	6.0977	3.7028	4.6308	5.1031	4.8500
S_{est}	0.2307	0.3180	0.2774	0.5734	0.5180	0.5554
Min 95% C.I.	-2.0365	-2.1098	-2.1192	-2.3783	-2.3054	-2.3685
Max 95% C.I.	-1.5722	-1.4698	-1.5609	-1.2244	-1.2629	-1.2508
P-value	0.0750	0.1278	0.1658	0.3534	0.2792	0.3577
Accept $H_0 : a = -2$	Yes	Yes	Yes	Yes	Yes	Yes

Table 5.16 Computing Cost Analysis: Number of Iteration for Example 5.2.1.

Mesh	k=100i	k=100	k=100+100i	k=0	K=10+10i	k=1+2i
32	9	14	10	8	8	9
64	9	12	9	7	7	8
128	9	12	10	8	8	9
256	9	12	10	9	9	8
512	9	12	10	9	9	9

Table 5.17 Computing Cost Analysis: Number of Iteration for Example 5.2.2.

Mesh	k=100i	k=100	k=100+100i	k=0	k=10+10i	K=1+2i
32	9	10	11	8	10	9
64	9	12	10	7	9	8
128	8	12	10	9	10	9
256	8	11	10	9	10	8
512	8	12	10	9	10	9

Table 5.18 Computing Cost Analysis: Number of Iteration for Example 5.2.3.

Mesh	k=100i	k=100	k=100+100i	k=0	K=10+10i	k=1+2i
32	10	11	12	11	12	12
64	9	11	10	11	11	11
128	14	16	15	15	16	15
256	10	13	11	13	13	12
512	17	17	15	18	19	18

5.3 Discussion

The entire numerical experiment are consisted of two group of problem, three set of interface, six different wave numbers and five increasing mesh size from 32 to 512. So we have total 180 original measurements for the error between exact solution and computed

solution; 180 number of iteration for the solver performs. We consider different mesh size but same problem, same interface and same wave number as one convergence test block in this dissertation.

Correlation study is conducted for every block of convergence on the correlation of the logarithm of computational error and logarithm of mesh size. More specifically the Pearson Product Moment Correlation Coefficient (PPMC) is computed, for the logarithm of the mesh size n as the independent variable, and the logarithm of the complex number norm of error as the dependent variable. The correlation coefficients are overwhelmingly close to −1.0. (See row of Coefficient Correlation at the statistical analysis tables). This shows there is strong negative linear relationship between the variables. The larger the mesh size, the smaller the error of the computed solution toward to exact solution. This is in line with what we have found later in linear regression analysis.

Linear regression is also conducted to find the line of best fit for every block of convergence [28]. There are 36 convergence order numbers. Their average value is 1.913 with standard deviation of 0.097. (See the row of Slope at statistical analysis tables). This shows that our computed solutions convergence to exact solution at the order very close to 2.

T-test is performed for each of convergence order (i.e. slope of linear convergence model, we write it as a notation for the rest of this discussion) [28]. See Rows of Sest (Standard Error of Estimate), Min.95% C.I. (Minimum Value of 95% Confident Interval),

Max.95% C.I. (Maximum Value of 95% Confident Interval) and P-value in the statistical analysis tables. Below is the summary:

Table 5.19 Summary of the Statistics Analysis for Convergence.

Catalog	T-test Conclusion	Frequency
Conv Order > –2	Enough Evident (α=0.01) to reject $H_{0:}$ a = –2	1
Conv.Order = -2^{+}	Weak Evident (α=0.01) not to reject $H_{0:}$ a = –2	4
Conv.Order = –2	Strong Evident (α=0.05) to accept $H_{0:}$ a = –2	27
Conv.Order = -2^{-}	Weak Evident (α=0.01) not to reject $H_{0:}$ a = –2	2
Conv.Order < –2	Enough Evident (α=0.01) to reject $H_{0:}$ a = –2	2

Please notice that there are few cases (total 3 as 8.3%) that we have to reject the claim of second order convergence one way or the other. This is because that there are small number of observations and thus narrow standard deviation for the data. The data is overwhelmingly normal distributed, and strongly support the claim of second order convergence.

Therefore, we are very confident that this method is second order convergence.

The convergence order for computational error is not affected by different functions and the geometry shapes of different interfaces. Our experimental functions and interfaces are all piecewise smooth. So, there exist remote isolated points which require special attention. For example, when conducting least square interpolation for the left most and right most bottom irregular points near the interface, we have to introduce the exact boundary

condition instead of relying only on the values from neighborhood points. The second order convergence for the least square interpolation is very crucial for the augment strategy and success of the entire project.

Our algorithm and its executions are stable, robust and efficient. We have 2 groups of test functions, three interfaces with level set, six wave numbers including from zero to 100+100i, and mesh size ranged from 32 to 512. The algorithm is always capable to return the approximate solution without hiccup. And the iteration of FFT solver remains relatively constant, independent to the problem, interface, wave number and mesh size, only varies with the GMRES error toleration.

Further, some random extreme parameters are experimented as individual cases. For example, when experiments with large wave number like, k=10000, or 10000i, the convergence order is affected compare to other wave number under the same mesh size, which are in line with other researchers' finding. Another example, when using large mesh size, like m=1024, 2048, the computational solution are still achieved second order convergence, through the computing time is couple hours long. As long as the Schur complement residual is relatively large compare with the GMRES error toleration, our method will return the solution with approximate second order of convergence. In future, execution updates, such as operating in a superior computer with fast CPU and huge memory, will be sure to improve perform of this algorithm.

CHAPTER 6

Conclusions

6.1 Research Conclusions

This dissertation has extended the Immersed Interface Method to two dimensional Helmholtz / Poisson equation problem in complex number space, the partial differential equation also have discontinuities and singularities in the coefficients and the solutions. In the process, we have achieved following.

- Developing new two-dimensional Fast Fourier Transformation method in complex number space to solve Helmholtz / Poisson equations on rectangular domains. This method is second order convergence with respect to the mesh size, it is stable and efficient. (Chapter 2).
- Analyzing and extending the Immersed Interface Method in two-dimensional Helmholtz / Poisson equation problem in complex number space with discontinuous and singularities. (Chapter 3).
- Investigating and modifying the Augment Strategy for function in complex number space which involves least square interpolation, Schur complement and GMRES (Chapter 4).

- Implementing all above methods/algorithms/strategies in numerical experiments, the numerical e10xperiments are successful and the results are consistent with our analytical prediction. (Chapter 5).

6.2 Future Research Work

In the future we would like to conduct further numerical investigation about other boundary conditions, i.e. Neumann and Robin conditions. Although the main theatrical analysis is presented in Chapter Two, the change of boundary condition may be required many subroutines and functions due to the nature of our interface, which demands more time and energy. Additional test examples in complex number space are also needed. We have so far explored wide range of wave number from 0 to 100+100i, but just two examples.

There are other plan of work can be considered, maybe in a long term effect. There are high convergence order Immersed Interface Method existing for two dimensional real-numbered Poisson and /or Helmholtz problem with discontinuous coefficients and solution. It would be an interesting challenge to extend it to complex number space. Another possible direction worth exploring is the three dimensional Poisson / Helmholtz problems in complex number space. They may have lots of real world application in electromagnet.

REFERENCES

[1]. K. Arthurs, L. Moore, C. Peskin, E. Pitman, and H. Layton. Modeling arteriolar flow and mass transport using the immersed boundary method. J. Comput. Phys., 147:402-440, 1998.

[2]. R.P. Beyer. A computational model of the cochlea using the immersed boundary method. J. Comput. Phys. 98:145-162, 1992.

[3]. A. Bluman. Elementary Statistics – A Step By Step Approach 3rd Ed. McGrawHill 2006.

[4]. D. Bottino. Modeling viscoelastic networks and cell deformation in the context of the immersed boundary method. J. Comput. Phys. 147:86-113, 1998.

[5]. R. Cortez, N. Cowen, R. Dillon, and L. Fauci. Simulation of swimming organisms: Coupling internal mechanics with external fluid dynamics. Comput. Sci. Engrg., 6:38-45, 2004.

[6]. James W. Demmel, Applied Numerical Linear Algebra, SIAM , 1997, pp.270-274.

[7]. R. H. Dillon, L. J. Fauci, and A. L. Fogelson, Modeling biofilm processes using the immersed boundary method. J. Comput. Phys., 129:57-73, 1996.

[8]. L. J. Fauci. Interaction of oscillating filaments: Acomputational study. J. Comput. Phys., 86:294-313, 1990.

[9]. X.Feng, Z.Li and Z.Qiao, High Order Compact Finite Difference Schemes For the Helmholtz Equation with Discontinuous Coefficients, J. Comp. Math.,Vol.29 No.3,2011.

[10]. A. L. Fogelson. A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting. J. Comput. Phys., 56:111-134, 1984.

[11]. A. L. Fogelson. Continuum models of platelet aggregation: Formulationand mechanical properties. SIAM J. Appl. Math., 52: 1089—1110, 1992.

[12]. A. L. Fogelson, and C. S. Peskin. A fast numerical method for solving the three dimensional Stokes equations in the presence of suspended particles. J. Comput. Phys., 79: 50—69, 1988.

[13]. V. Fraysse, L. Giraud, S. Gratton and J. Langou. A set of GMRES Routines for Real and Complex Arithmetics on High Performance Computers. ACM Transactions on Math. Software, Vol. 31, No. 2, June 2005, Page 228-238.

[14]. Z. Haznadar; Z. Stih. "Electromagnetic Fields Waves and Numerical Methods" IOS Press 2000.

[15]. Cooley James W.; Tukey, John W. (1965) "An algorithm for the machine calculation of complex Fourier series" Math. Comput. 19:297—301.

[16]. B. C. Khoo, Z. Li, P. Lin, Interface Problems and Methods in Biological and Physical Flows, World Scientific, 2009.

[17]. M. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J. Comput. Phys., 160:705—719, 2000.

[18]. Randall J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equation, SIAM, 2007.

[19]. R. L. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal., 31:1019—1044, 1994.

[20]. Zhilin Li, K. Ito, The Immersed Interface Method Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM, 2006.

[21]. Z. Li, T. Lin, Y. Lin, and R. C. Rogers, An immersed finite element space and its approximation capability, Numerical Methods for Partial Differential Equations, 20(3) 338-367, 2004. DOI: 10.1002/num. 10092.

[22]. Z. Li, H. Zhao, and H. Gao. A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid. J. Computing Phys., 152:281-304, 1999.

[23]. A. Mayo. The fast solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Numer. Anal. 21:285-299, 1984.

[24]. A. Mayo. A decomposition finite difference method for the fourth order accurate solution of Poisson's equation on general regions. Int. J. High Speed Comput., 3:89-105, 1991.

[25]. A. Mayo and A. Greenbaum. Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions. SIAM J.Sci. Statist. Comput., 13:101-118, 1992.

[26]. A. Mayo and A. Greenbaum. Rapid parallel evaluation of integrals in potential theory on general three-dimensional regions. J. Comput. Phys., 145:731742, 1998.

[27]. Claus Muller “Foundations of the Mathematical Theory of Electromagnetic Waves” Springer-Verlag, 1969.

[28]. R.Ott and M.Longnecker. Statistical Methods and Data Analysis. 5th Ed. Duxbury Thomson Learning 2001.

[29]. C. S. Peskin. Numerical analysis of blood flow in the heart. J. Comput. Phys., 25:220-252, 1977.

[30]. C. S. Peskin. The immersed boundary method. Acta Numer., 11:479-517, 2002.

[31]. C. S. Peskin and D. M. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. J. Comput. Phys., 37:113-132, 1980.

[32]. C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid. J. Comput. Phys., 81:372-405, 1989.

[33]. C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart. II. Contractile fibers. J. Comput. Phys., 82:289-297, 1989.

[34]. C. S. Pesking and B. F. Printz. Improved volume conservation in the computation of flows with immersed elastic boundaries. J. Comput. Phys., 105:33—46, 1993.

[35]. A.D. Polyanin, Handbook of Linear Partial Differential Equations for Engineers and Scientists, Chapman & Hall/CRC 2002.

[36]. Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J. Sci. Stat. Comput., 7:856-869, 1986.

[37]. J. Stockie and B. T. R. Wetton. Stability analysis for the immersed fiber problem. SIAM J. Appl. Math., 55:1577-1591, 1995.

[38]. J. Stockie and B. T. R. Wetton. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. J. Comput. Phys., 147:41-64, 1999.

[39]. D. Sulsky and J. U. Brackbill. A numerical method for suspension flow. J. Comput. Phys., 96: 339—368, 1991.

[40]. Paul N. Swarztrauber, Fast Poisson Solvers. Second International Conference on Computational Modeling of Free and Moving Boundary Problems 93, 1984.

[41]. J. A.N. Tikhonov and A.A. Tikhonov, Equations of Mathematical Physics, Dover Publ., New York, 1990.

[42]. N. Wang and A. L. Fogelson. Computational methods for continuum models of platelet aggregation. J. Comput. Phys., 151:649—675, 1999.

[43]. M. Zhao, Z. Qiao, T. Tang, A Fast High Order Method for Electromagnetic Scattering by Lage Open Cavities, J. Comp. Math.,Vol.29 No.3,2011.

[44]. L. Zhu and C. Peskin. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. J. Comput. Phys., 179:452—468, 2002.

APPENDICES

Appendix A

PROLOGUE OF THE PACKAGE FFT_POISSON_COMPLEX3

A FORTRAN CODE FOR SOLVING DOUBLE PRECISON COMPLEX NUMBER

HELMHOLTZ PROBLEM

```
   subroutine fft_poisson_complex3(m,n,f,h,k,u)

C**********************************************************************
C
C   Fast Fourier Transformation solver in retangular domain
C
C**********************************************************************
C
C This function solves the 2-D Helmholtz problem
C
C        U_xx + U_yy + k U = f(x,y)
C
C using the Fast Fourier Transformation in complex number.
C The domain is defined on a retangular region with equal sized
C mesh spacing, but the size m and n may be different.
C
C----------------------------------------------------------------------
C
C inputs:
C   m = # of row of the grid matrix
C   n = # of col of the grid matrix
C   f = matrix of modified rhs values evaluated at interior meshpoints
C   h = mesh spacing for x and y, hx=hy=h
C   k = constant in above formula
C
C----------------------------------------------------------------------
C
C outputs:
C   u = solution to PDE at interior meshpoints
C
C----------------------------------------------------------------------
```

```
C
C working spaces:
C  vm: m X m matrix for discrete Fourier transformation
C  vn: n X n matrix for discrete Fourier transformation
C  lambdam: 1 X m eigenvalue for discrete Fourier transformation
C  lambdan: 1 X n eigenvalue for discrete Fourier transformation
C  fbar: m X n matrix, discrete Fourier transformation f
C  ubar: m X n matrix, discrete Fourier transformation u
C  z: m X n matrix, middle value
C
C-----------------------------------------------------------------------
C
C Subrountinges called:
C  Matrix3Multi: Multiplication of 3 matrix  A(mXm) X B(mXn) X C(nXn)
C  ConstantMultiMatrix: constant multiplation to matrix
C
C-----------------------------------------------------------------------
C
C Precision: double complex (f-g,k,r-z)
C            double precision (a-e,h,l,o-q)
C            integer (m, n)
C
C-----------------------------------------------------------------------
C
C Written by Sidong Max Zhang, February, 2012
C
C***********************************************************************
C
C                END OF DOCUMENTATION FOR fft_poisson_complex3
C
C***********************************************************************
```

Appendix B

PROLOGUE OF THE PACKAGE GMRES

A FORTRAN CODE FOR SOLVING COMPLEX NUMBER SCHUR COMPLEMENT

RESIDUAL EQUATION USING GMRES METHOD

```
      subroutine gmres(n1max,mm,m,n,n1,n2,imax,a,b,c,d,h,phi,x,y,
     1           cinfo,zinfo,index,index2,zelmbda,uj,unj,u,f,
     2           zx0,zbf,tol,svdc1,svdc2,iter,error)

C**********************************************************************
C
C   GMRES for Schur complement residual equation
C
C**********************************************************************
C
C This subroutine solves
C
C        (D-CA^(-1)B)G=CA^(-1)F_1
C
C using the Generalized Minimum Residual method for complex number.
C to solve Schur complement in the Immersed Interface Method package
C
C----------------------------------------------------------------------
C
C inputs:
C   n1max =maximum number of irregular point allowed
C   mm = The number of previous vector used, (mm=80), for example
C   m = # of row of the grid matrix
C   n = # of col of the grid matrix
C   n1 = exact number of irregular point along the interface
C   n2 = exact inside irregular point along the interface
C   imax = maximum number of iteration allowed
C   a, b,c,d = left, right, bottom and top boundary
C   h = mesh spacing for x and y, hx=hy=h
C   phi = zero level set function for interface
C   x,y = grid points, x(0:m), y(0:n)
```

```
C   cinfo = irregular point information
C   zinfo = irregular point informtion
C   index =index for the irregular point
C   index2 =index for the inside irregular pint
C   zelmbda = wave number in the original equation
C   uj = [u] u jump condition on the interface
C   unj = [u_n] u_n jump condition on the interface
C   u = initial computed solution
C   f = matrix of modified rhs values evaluated at interior meshpoints
C   zx0 = initial residual of the Schur complement
C   zbf = residual of the Schur complement
C   tol =maximum norm 2 error allowed
C   svdc1 = coefficient of the least square interpolation 1
C   svdc2 = coefficient of the least square interpolation 2
C   error = norm 2 error
C
C-----------------------------------------------------------------------
C
C outputs:
C   u = solution to PDE at interior meshpoints
C   iter = number of iteration performed
C
C-----------------------------------------------------------------------
C
C working spaces:
C  zhg:
C  v:
C  vk:
C  zs:
C  zx11:
C  zhj:
C  ztemp:
C
C-----------------------------------------------------------------------
C
C Subrountinges called:
C   matvet1 computing matrix A and vector x, then find A^(-1)x
C   resid return vector zr=zb-zx
C   DZNRM2 returns the euclidean norm of a vector
```

```
C   ZSCAL scales a vector by a constant
C   ZDOTC forms the dot product of a vector
C
C-----------------------------------------------------------------------
C
C Precision: double complex (f-g,u-w,z)
C            double precision (a-e,h,o-t,x-y)
C            integer (i-n)
C
C-----------------------------------------------------------------------
C
C Written by Sidong Max Zhang, October, 2012
C
C***********************************************************************
C
C                END OF DOCUMENTATION FOR GMRES
C
C***********************************************************************
```