

2 2007

THESES

This is to certify that the				
dissertation entitled				

### MATCHED INTERFACE AND BOUNDARY (MIB) METHOD AND ITS APPLICATIONS TO IMPLICIT SOLVENT MODELING OF BIOMOLECULES

presented by

### YONGCHENG ZHOU

degree in

has been accepted towards fulfillment of the requirements for the

Ph.D

Applied Mathematics

Major Professor's Signature ily 25, 2006

Date

MSU is an Affirmative Action/Equal Opportunity Institution

LIBRARY Michigan State University

### PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due. MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

2/05 p:/CIRC/DateDue.indd-p.1

### MATCHED INTERFACE AND BOUNDARY(MIB) METHOD AND ITS APPLICATIONS TO IMPLICIT SOLVENT MODELING OF BIOMOLECULES

By Yongcheng Zhou

### A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

2006

© Copyright by Yongcheng Zhou 2006 All Rights Reserved

#### ABSTRACT

### MATCHED INTERFACE AND BOUNDARY(MIB) METHOD AND ITS APPLICATIONS TO IMPLICIT MODELING OF BIOMOLECULES

By

#### Yongcheng Zhou

This dissertation describes the matched interface and boundary (MIB) method [94, 96] for elliptic interface problems, in which the equations have discontinuous coefficients and possible singular source on the interface. This low regularity leads to the slow convergence or divergence of most traditional numerical methods for smooth problems. The complexity of the interface makes it more difficult to develop efficient numerical methods.

The matched interface and boundary method is closely related to the methods with ghost points while it differs from these methods in the implicit enforcing of the interface conditions. A uniform Cartesian grid is used in the formulation of the MIB to take advantages of the conventional high order central finite difference schemes. Attentions are only paid to the irregular grid points near the interface where the difference schemes involve the fictitious values instead of solely solution values. A local coordinate transformation is used to project the interface conditions defined in the gradient direction on the interface to the coordinate directions, which allows the MIB method to handle irregular interfaces. By iterative application of the interface problems are successfully derived for the first time.

Both immersed interface method (IIM) and MIB methods are used for the accurate solution of the Poisson-Boltzmann equation for electrostatic potentials. Molecular surface generated with MSMS is chosen as the dielectric interface and special techniques are developed to implement this triangulized molecular surface into the finite difference method with Cartesian grid. This is the first application of analytical molecular surface in the finite-difference-based Poisson-Boltzmann solvers, and is also the first method which conserves the continuity of potential flux at the molecular surface. Substantial improvements in the surface potentials and the electrostatic solvation energies are found as a result of this work. It is also found that, by mesh refinements, the solutions of all other traditional Poisson-Boltzmann solvers essentially converge to results of interface methods attained with a coarse mesh. Thus interface methods are relatively more efficient in achieving solutions of the same accuracy. The convergent solvation energies calculated from interface methods at coarse meshes not only permits an accurate energetic analysis of the molecular interactions but also provide an accurate calibration for the Generalized Born method. The accurate potential itself also has significant implications in the analysis of protein/ligand binding and association.

## Acknowledgements

This dissertation is the conclusion of what has been an extremely long journey, one that I could not have completed alone.

First of all, I would like to thank my thesis advisor, Dr. Guowei Wei, for giving me the opportunity to come to MSU, and for his invaluable assistance, guidance and encouragement throughout my Ph.D study and various research projects. He led me into the fascinating world of the scientific research, gave me the opportunity to take on such a stimulating challenge, and has shown me many things that have aided in my scientific growth. Our innumerable times discussing and questioning the problems, and cheering any achievements together will leave a lasting impression. It is always an enjoyable experience working with him.

I am also very grateful to Dr. Michael Feig for his help in my understanding of molecular biology, the applications of my mathematics in computational biology, and for providing me the computational resources. I also much appreciated my advisory committee members, Dr. Keith S. Promislow, Dr. Tien-Yien Li and Dr. Beisheng Yan for their directions, comments and suggestions.

Sincere thanks are also extended to Dr. Shan Zhao for many productive discussions, and Dr. Seiichiro Tanizaki for helping me on CHARMM, and Ms. Sining Yu and Mr. Weihua Geng, for their critical reading of my papers as well as source codes, and Mr. Matthew Ryan Otoole for proofreading this dissertation, and Dr. Zhengfeng Zhou, Director of Graduate Studies and Ms. Barbara Miller, Graduate Secretary in the Department of Mathematics, for their generous help throughout my graduate study at Michigan State University.

I would like to thank my family members back home in China, especially my

parents, Xiaonv Ye and Chenhuo Zhou, and my sisters, Yonghong Zhou and Xuemei Zhou, for their long-time sacrifices, support, confidence and encouragement. Finally and most importantly, I would like to express my utmost appreciation to my wife, Yuhui Sun, for her constant love, understanding and support.

## **Table of Contents**

Chapte	er 1 Introduction	1
1.1	Elliptic interface problem and its numerical methods: a brief review	2
1.2	Illustration of MIB method: 1-D example	7
Chapte	er 2 MIB Method: 2-D Formulation and Numerical Experiments	14
2.1	Second-order 2-D MIB scheme for irregular interface	15
2.2	Higher order 2-D MIB scheme for irregular interfaces	31
2.3	Subtleness in domain extensions	33
2.4	Interpolation formulation	37
	2.4.1 One dimensional formalism	38
	2.4.2 Two dimensional formalism	42
	2.4.3 Comparison of two formulations	45
2.5	Numerical experiments on irregular interfaces	47
Chapte	er 3 MIB Method: 3-D Formulation and Numerical Experiments	64
3.1	Second-Order 3-D MIB Scheme for Irregular Interfaces	64
3.2	Numerical Experiments	78
3.3	Comparison of MIB and IIM	79
Chapte	er 4 Poisson-Boltzmann Equation and Interface Methods	81
4.1	Implicit Solvent Modeling and the Poisson-Boltzmann Equation	82
4.2	Analytical Solutions and Numerical Techniques for the Poisson-Boltzmann	
	Equation: A Brief Review	88
4.3	Dielectric Interface	91
Chapte	er 5 Application of Interface Methods to the Poisson-Boltzmann	
Equ	ation	94
5.1	Implementation of molecular surface in MIB and IIM	95
5.2	Numerical validation of interface method for the Poisson-Boltzmann equation	n101
5.3	Application of interface method for biomolecules	106
Chapte	er 6 Thesis Contribution and Future Research Plan	119
6.1	Thesis Contribution	119
	6.1.1 The Matched Interface and Boundary Method	119
	6.1.2 Interface Methods for the Poisson-Boltzmann equation	120
6.2	Future research plan	121
	6.2.1 Theoretical analysis and fast algebraic solvers for the MIB method	121
	6.2.2 Investigation of biological significance of accurate surface potential	122
AP	PENDIX A Kirkwood Solution to Non-Centered Charged Sphere	125
Refere	nces	128

## List of Tables

1.1	Numerical convergence test for the equation $(\beta u_x)_x = f(x)$ . The computational domain is $[0, 2]$ and the interface is located at $x = a = 0.74174243050441595$ . $\beta = 1, u(x) = \sin(3x)$ if $x \leq a$ and $\beta = 10, u(x) = \cos(3x)$ if $x > a$	12
2.1	Numerical efficiency test of the 2-D Poisson equation in Example 2.5.1. $b = 10, [u] = 0, [\beta u_n] = -0.75.$	48
2.2	Numerical efficiency test of the 2-D Laplace equation in Example 2.5.3. $\beta = 1.$	51
2.3	Numerical convergence test and accuracy test for Example 2.5.4	53
2.4	Robustness test of high order MIB scheme with Example 2.5.4. $\beta^- = 1$ . $n_x = n_y = 40$ .	53
2.5	Numerical convergence test and accuracy test for Example 2.5.5	60
2.6	Numerical convergence test for Example 4	62
2.7	Numerical convergence test for Example 5	62
2.8	Comparison of the second-, fourth- and sixth-order MIB schemes for the Poisson equation (Case 6). The numbers in the parentheses are the CPU times in seconds for the generation of local difference schemes.	63
3.1	Numerical convergence test for Example(3.2.1)	78
3.2	Numerical convergence test for example(3.2.2)	79
4.1	Major Poisson-Boltzmann solvers. Notation: Poisson-Boltzmann equa- tion(PBE), multigrid(MG), algebraic multigrid(AMG) finite difference(FI finite element(FE), Gauss-Seidel(GS), conjugate gradient(CG), succes- sive over relaxation(SOR), stochastic dynamics(SD), quantum mechan- ics(QM), molecular mechanics(MM), molecular dynamics(MD)	D), 93
5.1	Positions and radii of atoms of cyclohexane.	100
5.2	Numerical convergence test for Example $(5.1.1)$	100
5.3	Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the error in the surface potential for a sphere with a centered unit charge. The exact solvation energy is -81.98 $kcal/mol$ .	110

5.4	Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and error in the surface potential for a sphere with a single noncentered charge.	111
5.5	Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and error in the surface potential for a sphere with a single noncentered charge	112
5.6	Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the error in the surface potential for a sphere with two noncentered charges	113
5.7	Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the error in the surface potential for a sphere with two noncentered charges	114
5.8	Electrostatic solvation energy in $kcal/mol$ of two positively charged spheres at four different mesh sizes $h = 0.5, 0.2, 0.1$ and 0.05. The distance between the centers of two spheres $D$ vary from 2Å to 5Å.	115
5.9	Summary of test proteins and the computed electrostatic solvation energies with MIB and PBEQ at different mesh sizes.	116
5.10	Summary of test proteins and the computed electrostatic solvation energies with MIB and PBEQ at different mesh sizes.	117

## List of Figures

- 1.1 Illustration of MIB method for 1-D problem. The dash line indicates the position of interface x = a, where u(x) has a discontinuity. Real solutions u(x) at grid points are marked in black and the fictitious values f(x) are marked in green. Fictitious values  $f_{i+1}$  and  $f_{i+2}$  are the smooth extension of the solution in left subdomain;  $f_{i-1}$  and  $f_i$  are the smooth extension of the solution in right subdomain. These fictitious values are solved by matching two interface conditions simultaneously.
- 2.1 Three typical situations for an interface crossing the mesh lines. In the left chart, the interface passes the x- mesh line at point A and y- mesh line at point B. The interface conditions are approximated at point A and point B to generate the difference scheme for  $(\beta u_x)_x$ and  $(\beta u_y)_y$ , respectively. In the middle chart, the interface conditions are approximated at point A to obtain the difference scheme only for  $(\beta u_x)_x$  and the regular central difference is used for  $(\beta u_y)_y$ . In the right chart, the difference scheme for  $(\beta u_y)_y$  is set up from the approximation of the interface conditions at point B while term  $(\beta u_x)_x$  is treated with the regular central difference.
- 2.2Irregular point (i, j) and the interface. The interface crosses the xmesh line at  $(x_o, y_o)$ . The fictitious values are  $f_{i,j}$  and  $f_{i+1,j}$  (green). The vertical dash line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (o, j+2), (o, j+1) and (o, j) right at  $(x_o, y_o)$ . The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at  $(x_o, y_o)$ .
- $\mathbf{2.3}$ Irregular point (i, j) and the interface. The interface crosses the ymesh line at  $(x_o, y_o)$ . The dash horizontal line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (i - i)2, o), (i-1, o) and (i, o) right at  $(x_o, y_o)$ . The jumps  $[u], [\beta u_n]$  and  $[u_\tau]$
- 2.4 Irregular point (i, j) and the interface. The interface passes the grid point (i, j). No auxiliary lines and auxiliary points are needed. The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at the point  $(i, j), \ldots, \ldots$ 26
- Irregular point (i, j) and the interface. The interface crosses the x-2.5mesh line at  $(x_o, y_o)$ . The fictitious values are  $f_{i,j}$  and  $f_{i+1,j}$  (green). The vertical dashed line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (o, j-1), (o, j+1) and (o, j) right at  $(x_o, y_o)$ . The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at  $(x_o, y_o)$ .

30

8

16

17

22

2.6	A typical stencil used in constructing a fourth-order scheme for $u_{xx}$ and $u_x$ . There are two pairs of fictitious points in this case: $f_{i,j}, f_{i+1,j}$ and $f_{i-1,j}, f_{i+2,j}, \ldots, \ldots, \ldots, \ldots$ .	31
2.7	A typical stencil used in constructing a sixth order scheme for $u_{xx}$ and $u_x$ . Now there are three pairs of fictitious points in this case: $f_{i,j}, f_{i+1,j}$ and $f_{i-1,j}, f_{i+2,j}$ as well as $f_{i-2,j}, f_{i+3,j}, \ldots, \ldots$	32
2.8	Fictitious value $f_{i,j+2}$ shall be used for the discretization of $u_y, u_{yy}$ at grid point $(i, j + 1)$ . However, $f_{i,j+2}$ can not be solved along the y-direction but it can be solved along x-direction, i.e., solved together with fictitious values $f_{i-2,j+2}, f_{i-1,j+2}$ and $f_{i+1,j+2}, \ldots, \ldots$	34
2.9	The distribution of fictitious points (stars) for a $40 \times 40$ mesh with irregular interface. Each fictitious domain roughly has two layers of fictitious points, supporting a fourth-order central difference scheme. The green stars represent the smooth continuation of the interior sub- domain and the red stars denote the smooth continuation of the outer subdomain.	36
2.10	(Same as Figure 2.2) Irregular point $(i, j)$ and the interface. The interface crosses the $x$ - mesh line at $(x_0, y_0)$ . The vertical dashed line is the auxiliary line on which three auxiliary points (in empty circle) are defined: $(o, j + 2), (o, j + 1)$ and $(o, j)$ right at $(x_0, y_0)$ . The jumps $[u], [\beta u_n]$ and $[u_{\tau}]$ are evaluated at $(x_0, y_0)$ .	43
2.11	Computed solution (upper) and the error (lower) for the 2-D Poisson equation in Example 2.5.1. $\beta = 10, [u] = 0, [\beta u_n] = -0.75.$	49
2.12	The computed solution (upper) and the error (lower) for Example $2.5.2$ .	50
2.13	Computed solution (upper) and the error (lower) for the 2-D Laplace equation in Example 2.5.3. $\beta \equiv 1. \dots \dots \dots \dots \dots \dots$	52
2.14	The computed solution (upper) and the error (lower) for Example 2.5.4 with second-order method and $n_x = n_y = 40. \ldots \ldots \ldots \ldots$	54
2.15	The computed solution (upper) and the error (lower) for Example2.5.5.	57
2.16	Upper: the computed solution of Example(2.5.6); Lower: computed solution of Example(2.5.7). Both are computed with mesh $n_x = n_y = 100.$	61
2.17	The solution of the high order MIB method for the 2-D Poisson equation in Example 2.5.8. $-u(x, y)$ is plotted.	62

- 3.1 Local topology around irregular point (i, j, k). The interface crosses the *x*-mesh line at the point  $S = (x_o, y_o, z_o)$  between (i, j, k) and (i+1, j, k), which are two irregular points, on which two fictitious values,  $f_{i,j,k}$ and  $f_{i+1,j,k}$  are defined (marked with green dots). Two auxiliary lines (dashed line) are sketched passing through S, one on the x - y plane and the other on the x - z plane. Two auxiliary points (in empty circle) (o, j, k+1) and (o, j, k+2) are placed on the auxiliary line on the x - zplane to facilitate the discretization of  $u_z^-$ ; Also, two auxiliary points (o, j, k+1) and (o, j, k+2) are placed on the auxiliary line on the x - yplane to facilitate the discretization of  $u_y^-$ .
- 3.2 Local topology with interface intersecting at a grid point (i, j, k). Two fictitious values,  $f_{i,j,k}$  and  $f_{i+1,j,k}$  are defined (marked with green dots). 77
- 5.1 The illustration of mesh lines and triangle as well as their intersect points. If the intersect point happens to be a vertex, or passes through a side of the triangle, or the sum of three angles  $\theta_1, \theta_2$  and  $\theta_3$  is  $2\pi$ , then the intersect point is inside the triangle....
- 2-D diagram of the accessibility determined by chasing intersec points 5.2 of mesh lines and molecular surface. Accessible grid points are marked by solid red circles and the inaccessible grid points are marked by empty black circles. For the upper mesh line  $(y = y_4)$  which has 6 intersection points with the molecular surface(the right most one is counted twice as the mesh line is tangent to the molecular surface at this point), the grid points (1,4) through (3,4) are accessible because they are located before the first intersection point; grid points (4,4) through (5,4) are inaccessible because they are located between the first and the second intersection point; grid points (6,4) through (8,4) are accessible again because they are located between the second and the third intersection point; grid points (9,4) and (10,4) are inaccessible again because they are located between the third and the fourth intersection point; grid points (11,4) through (15,4) are accessible because they are located between the fourth and the fifth intersection point. Between the fifth and the sixth intersection points there are no grid points. After the sixth intersection point which is the last one on this mesh line, all the grid points are accessible.

97

99

73

- 5.3 The grid points and interface point needed for the formulation of the IIM scheme [13] at the irregular point(i, j, k). The IIM method manages to find the Taylor expansions of the solution accurate to  $O(h^3)$ at both sides of the interface, for which a total of 16 expansion coefficients are to be determined. Original IIM requires 10 grid points and 6 interface conditions at an interface point(p' or p) to solve for the 16 polynomial coefficients. These 10 points(red) are chosen to be (i, j, k), other 6 grid points directly connecting to (i, j, k) and 3 more grid points inside the cube, preferably at the other side of the interface. IIM with optimization employs all the 27 grid points (red and green) inside the cube to produce constrained quadratic optimization problems from which 16 polynomial coefficients are solved. See [13] for the detailed formulation of 3-D IIM scheme. It is not necessary to choose the interface point to be the nearest one as p'. For simplicity we choose p', the intersect point of the interface with the mesh line, as 101
- 5.5 Left y-axis: the deviation of electrostatic solvation energy calculated with PBEQ at h = 0.5 (circle), h = 0.25(diamond) and h = 0.2(square) from that calculated with MIB at h = 0.5. Right y-axis: the solid line plots the total absolute charge of the proteins. x-axis from left to right: 1ajj, 1vii, 2erl, 1bbl, 1cbn, 2pde, 1sh1, 1fca, 1ptq, 1bor, 1uxc, 1vjw, 1fxd, 1hpt, 1mbg, 1bpi, 1r69, 451c, 1neq, 1a2s, 1svr, 1frd, 1a63 and 1a7m.
  5.6 The map of electrostatic potential on the surface of protein 1a63. Upper: potential computed with MIB; Lower: Difference of potentials
- computed with PBEQ and with MIB.

## Chapter 1

### Introduction

Many problems in fluid dynamics, material science and molecular biology can be modeled by elliptic equations with singular sources or discontinuous diffusion coefficients at fixed or moving internal interfaces, such as viscous incompressible flow with an elastic weightless immersed boundary (Stokes problem of blood flow [63]), stable or unstable crystal growth (Stefan problem [22, 25]) or implicit solvent modeling in biomolecular simulation (Poisson-Boltzmann equation [28]). The singularities in the source and/or the diffusion coefficient will cause the reduction of the global regularity of the solution, and consequently, lead to poor performances of most standard numerical methods designed for smooth solutions, usually seen through degradation of the convergence rate or even leadint to divergence of the numerical scheme.

This dissertation is about the design, analysis and applications of an accurate and robust numerical algorithm, the matched interface and boundary(MIB) method, for elliptic interface problems on Cartesian grids. The MIB method was first proposed for the numerical solution of the Maxwell's equation for electromagnetic fields in media with straight material interfaces [94]. It will be demonstrated that the MIB method can be generalized to 2-D or 3-D elliptic problems with arbitrary interfaces given an appropriate local coordinate transformation, a sophisticated organization of interface conditions and a deliberated selection of grid points for the solution of fictitious values. Substantial improvement on accuracy and resolution, as well as savings of the computational cost are achieved with this new method. Among the variety of potential applications of interface methods, we are particularly interested in the application of MIB to the numerical solution of the Poisson-Boltzmann equation derived from an implicit solvent model of biomolecular salty, aqueous environment. This application possesses physical elegance, mathematical complexity and biological importance.

## 1.1 Elliptic interface problem and its numerical methods: a brief review

An elliptic interface problem is given by the equation

$$\nabla \cdot (\beta \nabla u) - \kappa(\mathbf{x}) \mathbf{u}(\mathbf{x}) = \rho(\mathbf{x}), \quad \mathbf{x} \in \mathbf{\Omega} = \mathbf{\Omega}^+ \cup \mathbf{\Omega}^-$$
(1.1)

with an appropriate boundary condition on  $\partial\Omega$ . For simplicity,  $\Omega$  is assumed to be a regular domain, such as a rectangle in two dimensions (2D) or a cuboid in three dimensions (3D).  $\Omega^-$  and  $\Omega^+$  are two complementary subdomains of  $\Omega$  which are separated by an internal interface  $\Gamma$ . On this interface either the diffusion coefficient  $\beta$  is discontinuous or the source term  $\rho(x)$  is singular, or both have singularities. Supplemented by the underlying physics of the equation, we have two jump conditions at the interface:

$$[u] = u^{+}(\mathbf{X}(s)) - u^{-}(\mathbf{X}(s)) = \phi(s), \quad [\beta u_{n}] = \beta^{+}u_{n}^{+}(\mathbf{X}(s)) - \beta^{-}u_{n}^{-}(\mathbf{X}(s)) = \psi(s).$$
(1.2)

where  $\mathbf{X}(s)$  is a point on the interface  $\Gamma$ , s is the arc-length parametrization of the interface  $\Gamma$ , and n is the outer normal direction of the interface. The superscript  $\pm$ denotes the limiting value of a function from one side or the other of the interface. In either subdomain  $\Omega^-$  or  $\Omega^+$  the function  $u(\mathbf{x})$  is smooth, but global regularity of the solution is usually low, due to the finite jumps in the solution and/or its gradient on the interface. Most of the standard numerical methods (excluding discontinuous Galerkin or similar methods) assume that the solution is smooth inside its discretization stencil, which will not hold for the case at hand if the discretization stencil includes grid points located in different subdomains. Special treatments or modifications are therefore necessary to extend the application of many well developed numerical methods such as finite difference method and finite element method to these elliptic interface problems.

One of the earliest investigations on the elliptic interface problems was conducted by Babuska in 1970 using the finite element method [3], whereby he formulated the interface problem as an equivalent minimization problem and incorporated all the jump conditions into the cost functions. This method and a technique developed by Xu [91], require the exact evaluation of integrals on the boundary and the internal interface, which is usually difficult to obtain for an arbitrary interfaces or boundaries.

In the 1970-1980s, Peskin [50, 63, 64] developed an immersed boundary method to model blood flow in the heart. In the immersed boundary method, the complicated time-varying geometric boundary is regarded as being immersed in the fluid, and the Navier-Stokes equation is solved on a simple Cartesian grid. The presence of the embedded boundary is modeled via a singular source localized on the interface. This singular source is then approximated and distributed over a small neighborhood near the interface, usually through some discrete delta functions. The immersed boundary method has had success in modeling flows in complicated time-dependent geometry [17, 92] due to its flexibility, efficiency, and robustness. Recently, Tornberg and Engquist have proposed and analyzed a class of regularization methods for treating the singular source terms [79]. An extensive review of the immersed boundary method for turbulent flow simulations can be found in [33]. However, the immersed boundary method is typically first order accurate in higher dimensions and the smearing of the jumps along the interface might prevent its application to circumstances where sharp jumps and/or the resolution of the solution at the interface are of concern.

The success of these pioneering works and the identification of more interface problems have motivated the emergence of a variety of interface methods in the last decade, among them a major advance in this field was due to LeVeque and Li [42]. These authors proposed the immersed interface method (IIM) for solving elliptic equations with discontinuous coefficients and singular sources. In the IIM, local corrections of finite difference schemes on irregular grid points where the discretized Laplacian operator involves nodes from both sides of the interface, are pursued throughout the domain. This is achieved by incorporating interface jump conditions into local Taylor expansions of the operator on irregular points, from which FD schemes with genuinely first-order accurate truncation error can be derived. The resulting scheme is of second order accuracy and preserves the jumps at the interface. For 2-D or 3-D problems, a local coordinate is typically required to offer a better representation of the jump conditions since they are given in the direction normal to the interface. The IIM is robust and efficient, and has been successfully applied to a variety of interface-related problems, such as moving interface problems [30], elliptic irregular domain problems [14] and 3-D interface problems [13], to name a few. The reader is referred to a recent review [45] or Li's monograph [47] on IIM method for more details about these applications.

Various extensions and further improvements of the IIM have been considered in the literature [2, 6, 7, 14, 29–32, 35, 38, 41, 48, 49, 52, 69, 71, 81, 83, 90], including the formulations in polar coordinates [49], the formulation based on the finite element method [44] and the formulation proposed by Dumett and Keener [14] for solving anisotropic elliptic boundary value problems. On the other hand, the problem of convergence and efficiency of the IIM has attracted many research interests because the original IIM [42] typically leads to a matrix of non-symmetric coefficients though the original problem is self-adjoint and strictly elliptic. This reduces the number of standard fast solvers that can be utilized with IIM, and convergence may not be rigorously guaranteed [31]. To address this issue, a maximum principle preserving IIM was proposed by Li and Ito [46] to attain a diagonally dominant albeit still non-symmetric coefficient matrix. Specially designed multigrid solvers can then be employed to speed up the convergence of the maximum principle preserving IIM [1, 40]. For interface problems with piecewise constant coefficients, a fast IIM was constructed [43] through introduction of an unknown jump condition for  $[u_n]$ . Together with the elliptic equation, this unknown will also be solved numerically. The success of the fast IIM lies in the fact that it models the jump conditions in u(x)and its gradient as correction terms for the standard finite difference approximation to the equation. The latter is solved with fast Poisson solvers whilst the correction term is updated using the solution of u(x). A satisfactory convergence was found from this iterative procedure. Motivated by the fast IIM [43], an explicit jump IIM was developed by Wiegmann and Bube [90]. Recently, a decomposed IIM for elliptic equations with variable coefficients was proposed by Berthelsen [6]. The linear systems in two IIMs [89] and [6] are all symmetric and diagonally dominant, allowing the use of conventional fast Poisson solvers.

Another popular sharp interface scheme using the Cartesian grid is the ghost fluid method (GFM) originally developed for treating contact discontinuities in the inviscid Euler equations by Osher and his coworkers [18]. The GFM is typically first order accurate for interface problems, including the elliptic one [51] and could be of second order accuracy for elliptic irregular domain problems. In the flavor of the level set method which gives an implicit representation of the interface, the interface jump conditions are captured implicitly by extending values across the interface into a ghost fluid. On irregular grid points, when the FD discretized Laplacian refers to a node from the other side of the interface, a ghost fluid value instead of the real one will be supplied. Such a jump condition capturing procedure is directly incorporated into the numerical discretization in a way that the symmetry of the FD coefficient matrix is maintained, allowing the use of standard fast solvers. In higher dimensions, the jump in the normal derivative is correctly captured through a projection to Cartesian coordinate directions in the GFM, while the jump in the tangential derivatives is neglected [51]. In this way the GFM can be applied dimension by dimension. The GFM is very simple and robust, and its practical extensions to complex interface problems such as 3D moving interfaces or the multiphase Navier-Stokes equations are promising. Recently, an interesting jump condition capturing FD scheme was

constructed by Wang [84] by using a body-fitting curvilinear coordinate system.

It is noted that there also exist numerous studies in the literature about the quite relevant elliptic irregular domain problem [36, 59, 60]. One way to solve this problem is to embed the irregular domain to a larger regular computational domain, and reformulate the original boundary conditions as interface jump conditions. A simple Cartesian grid can then be adopted, so that various fast algebraic solvers developed in the literature can be utilized. This essentially converts an elliptic irregular domain problem into an elliptic interface problem. Due to this close relationship, the methods originally designed for one type of problems may be extended for another one. However, in the immersed boundary problems, no solution is sought outside the domain boundary, whereas, in immersed interface problems, interface jump conditions couple the solutions on both sides of the interface. Moreover, an immersed interface problem has two interface conditions by definition while an immersed boundary problem has only one 'interface condition' coming from the boundary condition of the original problem. In this dissertation, we will primarily focus on solving the elliptic immersed interface problems. The application of our method will be extended to a few immersed boundary problems.

To solve an elliptic problem with an irregular domain or interface, a body-fitting grid can be employed [3, 10, 11]. Nevertheless, for certain geometrically complex domains, the construction of a good body-fitting mesh remains a nontrivial and time-consuming task, even though considerable progress has been made. Furthermore, a considerable increase of computational difficulties will be encountered for moving interface problems, where a moving mesh method is required to regenerate or deform the grid during the simulation. Therefore, numerous modified finite difference methods that are based only on a simple Cartesian grid have been chosen to solve the elliptic interface-related problems in the literature [1, 6, 17, 18, 30, 31, 33, 36, 40, 42, 44–46, 50, 51, 59, 60, 63, 64, 84, 89, 92]. One obvious advantage of Cartesian grid methods is that there is no computational cost for grid generation. A Cartesian grid also allows the use of simple data structures and a standard FD stencil over a ma-

jority of the domain. Moreover, many contemporary software packages, such as fast Poisson solvers, multigrid, level set method, etc. are mainly developed for a Cartesian grid, and thus could be taken advantage of. On the other hand, in order to properly maintain the accuracy at the interface, some extra numerical work needs to be done near the interface in a Cartesian FD method. In general the growth of computational cost due to these extra operations is very small compared to the solution of the linear system as the modification of numerical methods occurs only in a small region near the interface.

The matched interface and boundary (MIB) method to be formulated in this dissertation is based on fictitious points, which are regarded as the smooth extension of the subdomains separated by an internal interface. The solution values at these fictitious points are determined by hierarchically matching two given interface conditions. The idea of fictitious points or ghost domain was also used in the literature for solving elliptic irregular domain problems by Mayo [59, 60] in the 1980s. The ghost cell is introduced outside of the (inner) domain as a fictitious domain. Similar ideas have been explored in Refs. [36, 92], but the fictitious values are smoothly extended or extrapolated according to the jump conditions. Our MIB formulation differs from these in the sense that fictitious values on the different subdomains are solved pairwisely and simultaneously. Most importantly, the determination of fictitious values are independent of the coordinate direction and the discretization of the elliptic equation. The resultant flexibility makes it possible for the MIB method to handle elliptic problems with irregular interfaces.

### **1.2** Illustration of MIB method: 1-D example

First proposed by Zhao and Wei [94] for the solution of the Maxwell's equation with straight material interface, the MIB method manages to implicitly enforce both interface conditions by modifying the discretization schemes near the dielectric interface. A major ingredient of the MIB method can be illustrated with the following 1-D example. Consider a uniform mesh with grid spacing h and let the interface be located at x = a where  $x_i \leq a \leq x_{i+1}$  for some *i*, as in Figure 1.1. Suppose the diffusion coefficient is  $\beta^-$  in the left subdomain and  $\beta^+$  in the right subdomain. For a one-dimensional elliptic equation (1.3) with interface conditions (1.4)

$$(\beta u_x)_x = \rho(x), \tag{1.3}$$

$$[u] = u^{+}(a) - u^{-}(a), \ [\beta u_{x}] = \beta^{+} u_{x}(a) - \beta^{-} u_{x}(a),$$
(1.4)

a straightforward application of the finite difference scheme says at  $x_i$ 

$$\frac{\beta_{i+1/2}u_{i+1} - 2(\beta_{i+1/2} + \beta_{i-1/2})u_i + \beta_{i-1/2}u_{i-1}}{h^2} = \rho(x_i), \quad (1.5)$$

where  $\beta_{i+1/2}$  and  $\beta_{i-1/2}$  are the values of diffusion coefficient at positions  $x_{i+1/2}$  and  $x_{i-1/2}$ , respectively. This discretization, based on Taylor expansions of smooth functions, renders a second-order convergence only if both u(x) and  $\beta(x)$  are continuous, which is not the case for the elliptic interface problems.



**Figure 1.1:** Illustration of MIB method for 1-D problem. The dash line indicates the position of interface x = a, where u(x) has a discontinuity. Real solutions u(x) at grid points are marked in black and the fictitious values f(x) are marked in green. Fictitious values  $f_{i+1}$  and  $f_{i+2}$  are the smooth extension of the solution in left subdomain;  $f_{i-1}$  and  $f_i$  are the smooth extension of the solution in right subdomain. These fictitious values are solved by matching two interface conditions simultaneously.

The degradation of the convergence of standard finite difference on elliptic interface problems is attributed to failure to ensure the interface conditions. With the MIB method, one modifies the finite difference scheme near the interface by first smoothly extending the subdomains separated by the interface and then applying the standard finite difference scheme on the extended subdomains. As seen from Figure 1.1, a second-order central difference scheme at  $x_i$  involves grid points  $x_{i-1}, x_i$  and  $x_{i+1}$ , but  $x_{i+1}$  is located on the other side of the interface. Since on the left subdomain, only the difference scheme at  $x_i$  involves the grid point on the other subdomain, it is sufficient to smoothly extend the left subdomain up to  $x_{i+1}$  to accommodate the discretization at  $x_i$ . Similarly, the right subdomain would be left-extended up to  $x_i$ . The solution values on these two extended domains are called fictitious values f(x)to distinguish them from the real solution values u(x) at the same place. These two fictitious values  $f_i$  and  $f_{i+1}$  can be solved from an approximation of the interface conditions (1.4)

$$(w_{0,i}^{+}f_{i} + w_{0,i+1}^{+}u_{i+1} + w_{0,i+2}^{+}u_{i+2}) - (w_{0,i-1}^{-}u_{i-1} + w_{0,i}^{-}u_{i} + w_{0,i+1}^{-}f_{i+1})$$

$$= [u] \qquad (1.6)$$

$$\beta^{+}(w_{1,i}^{+}f_{i} + w_{1,i+1}^{+}u_{i+1} + w_{1,i+2}^{+}u_{i+2}) - \beta^{-}(w_{1,i-1}^{-}u_{i-1} + w_{1,i}^{-}u_{i} + w_{1,i+1}^{-}f_{i+1})$$

$$= [\beta u_{x}], \qquad (1.7)$$

where  $(w_{0,i}^+f_i + w_{0,i+1}^+u_{i+1} + w_{0,i+2}^+u_{i+2})$  is the interpolation of  $u^+(a)$  on  $u_{i+1}, u_{i+2}$ and fictitious value  $f_i$  with corresponding interpolation weights  $w_{0,i+1}^+, w_{0,i+2}^+$  and  $w_{0,i}^+$ .  $(w_{1,i}^+f_i + w_{1,i+1}^+u_{i+1} + w_{1,i+2}^+u_{i+2})$  is the approximation of  $u_x^+(a)$  at the same set of nodes via finite difference. Similar approximations to  $u^-(x)$  and  $u_x^-(a)$  are conducted on the nodes  $u_{i-1}, u_1$  and fictitious value  $f_{i+1}$  with respective interpolation weights or finite difference coefficients, which are calculated with Fornberg's algorithm [21]. The equations (1.6 and 1.7 are linear in  $f_i$  and  $f_{i+1}$ :

$$w_{0,i}^{+}f_{i} - w_{\overline{0},i+1}^{-}f_{i+1} = [u] - w_{0,i+1}^{+}u_{i+1} - w_{0,i+2}^{+}u_{i+2} + w_{\overline{0},i-1}^{-}u_{i-1} + w_{\overline{0},i}^{-}u_{i},$$
  

$$\beta^{+}w_{1,i}^{+}f_{i} - \beta^{-}w_{\overline{1},i+1}^{-}f_{i+1} = [\beta u_{x}] - \beta^{+}(w_{1,i+1}^{+}u_{i+1} + w_{1,i+2}^{+}u_{i+2})$$
  

$$+\beta^{-}(w_{\overline{1},i-1}^{-}u_{i-1} + w_{\overline{1},i}^{-}u_{i}).$$

The fictitious values  $f_i$  and  $f_{i+1}$  are solved to be the linear combinations of real solution values  $u_{i-1}, u_i, u_{i+1}, u_{i+2}$  and the prescribed jumps  $[u], [\beta u_x]$ :

$$\begin{split} f_i &= \mathbf{C}_1^i u_{i-1} + \mathbf{C}_2^i u_i + \mathbf{C}_2^i u_{i+1} + \mathbf{C}_4^i u_{i+2} + \mathbf{C}_5^i [u] + \mathbf{C}_6^i [\beta u_x], \\ f_{i+1} &= \mathbf{C}_1^{i+1} u_{i-1} + \mathbf{C}_2^{i+1} u_i + \mathbf{C}_2^{i+1} u_{i+1} + \mathbf{C}_4^{i+1} u_{i+2} + \mathbf{C}_5^{i+1} [u] + \mathbf{C}_6^{i+1} [\beta u_x], \end{split}$$

where the vectors are

$$\mathbf{C}^{i} = \frac{-\beta^{-}w_{1,i+1}^{-}\mathbf{K}_{1} + w_{0,i+1}^{-}\mathbf{K}_{2}}{D}$$

$$\mathbf{C}^{i+1} = \frac{-\beta^{+}w_{1,i}^{+}\mathbf{K}_{1} + w_{0,i}^{+}\mathbf{K}_{2}}{D}$$

$$\mathbf{K}_{1} = (w_{0,i-1}^{-}, w_{0,i}^{-}, -w_{0,i+1}^{+}, -w_{0,i+2}^{+}, 1, 0)$$

$$\mathbf{K}_{2} = (\beta^{-}w_{1,i-1}^{-}, \beta^{-}w_{1,i}^{-}, -\beta^{+}w_{0,i+1}^{+}, -\beta^{+}w_{0,i+2}^{+}, 0, 1)$$

and  $D = -\beta^- w_{1,i+1}^- w_{0,1}^+ + \beta^+ w_{0,i+1}^- w_{1,i}^+$ . These two fictitious values are supplied to the standard central difference schemes at  $x_i$  and  $x_{i+1}$  such that

$$\frac{\frac{\beta_{i-1/2}u_{i-1} - 2(\beta_{i-1/2} + \beta_{i+1/2})u_i + \beta_{i+1/2}f_{i+1}}{h^2} = \rho(x_i), \text{ at } x_i,$$

$$\frac{\frac{\beta_{i+1/2}f_i - 2(\beta_{i+1/2} + \beta_{i+3/2})u_i + \beta_{i+3/2}u_{i+2}}{h^2} = \rho(x_{i-1}), \text{ at } x_{i+1}$$

are actually conducted on smooth subdomains, and thus have a better convergence property than (1.5), as illustrated by Table 1.1.

More fictitious values can be solved to support higher order discretization of the equation near the interface via iterative enforcing of the interface conditions as only two fictitious values can be solved from two interface conditions. Following the above procedure, we first solve  $f_i$  and  $f_{i+1}$  from

$$\left(w_{0,i}^{+}f_{i} + \sum_{k=i+1}^{i+4} w_{0,k}^{+}u_{k}\right) - \left(\sum_{k=i-3}^{i} w_{0,k}^{-}u_{k} + w_{0,i+1}^{-}f_{i+1}\right) = [u] \quad (1.8)$$

$$\beta^{+}(w_{1,i}^{+}f_{i} + \sum_{k=i+1}^{i+4} w_{1,k}^{+}u_{k}) - \beta^{-}(\sum_{k=i-3}^{i} w_{1,k}^{-}u_{k} + w_{1,i+1}^{-}f_{i+1}) = [\beta u_{x}] \quad (1.9)$$

which approximate the interface conditions (1.4) at fifth-order accuracy. We then solve  $f_{i-1}$  and  $f_{i+2}$  from

$$(w_{0,i-1}^{+}f_{i-1} + w_{0,i}^{+}f_{i} + \sum_{k=i+1}^{i+4} w_{0,k}^{+}u_{k}) - (\sum_{k=i-3}^{i} w_{0,k}^{-}u_{k} + w_{0,i+1}^{-}f_{i+1} + w_{0,i+2}^{-}f_{i+2}) = [u], \qquad (1.10)$$
  
$$\beta^{+}(w_{1,i-1}^{+}f_{i-1} + w_{1,i}^{+}f_{i} + \sum_{k=i+1}^{i+4} w_{1,k}^{+}u_{k}) - (\sum_{k=i-3}^{i} w_{1,k}^{-}u_{k} + w_{1,i+1}^{-}f_{i+1} + w_{1,i+2}^{-}f_{i+2}) = [\beta u_{x}]. \qquad (1.11)$$

Using two fictitious values at either side of the interface is consistant with a fourthorder central finite difference in the vicinity of the interface, for example,

$$u_{xx} = -\frac{f_{i+1}}{12h^2} + \frac{4u_i}{3h^2} - \frac{u_{i-1}}{4h^2} + \frac{4u_{i-2}}{3h^2} - \frac{u_{i-3}}{12h^2} \quad \text{at} \quad x_{i-1},$$
(1.12)

$$u_{xx} = -\frac{J_{i+2}}{12h^2} + \frac{4J_{i+1}}{3h^2} - \frac{u_i}{4h^2} + \frac{4u_{i-1}}{3h^2} - \frac{u_{i-2}}{12h^2} \quad \text{at} \quad x_i.$$
(1.13)

$$u_{xx} = -\frac{u_{i+3}}{12h^2} + \frac{4u_{i+2}}{3h^2} - \frac{u_{i+1}}{4h^2} + \frac{4f_{i-1}}{3h^2} - \frac{f_{i-2}}{12h^2} \quad \text{at} \quad x_{i+1}, \qquad (1.14)$$

$$u_{xx} = -\frac{u_{i+4}}{12h^2} + \frac{4u_{i+3}}{3h^2} - \frac{u_{i+2}}{4h^2} + \frac{4u_{i+1}}{3h^2} - \frac{f_i}{12h^2} \quad \text{at} \quad x_{i+2}, \tag{1.15}$$

to achieve fourth-order convergence, as shown in Table 1.1.

In addition to the convergence rate of the scheme, we are also interested in the relation between the accuracy of the MIB scheme and the magnitude of the jump in the diffusion coefficient  $\beta$ . Considering a second-order MIB scheme for which only

**Table 1.1:** Numerical convergence test for the equation  $(\beta u_x)_x = f(x)$ . The computational domain is [0, 2] and the interface is located at x = a = 0.74174243050441595.  $\beta = 1, u(x) = \sin(3x)$  if  $x \le a$  and  $\beta = 10, u(x) = \cos(3x)$  if x > a.

	2 <sup>nd</sup> MIB		4 <sup>th</sup> N	4 <sup>th</sup> MIB	
$n_{x}$	$L_{\infty}$	Order	$L_{\infty}$	Order	
20	2.80E-2		1.18E-4		
40	6.89E-3	2.02	8.80E-5	3.75	
80	1.64E-3	2.07	5.37E-6	4.03	
160	3.98E-4	2.04	2.88E-7	4.22	
320	1.02E-4	1.96	1.79E-8	4.01	

one fictitious value is defined on either side of the interface, i.e.,  $f_i$  and  $f_{i+1}$ , the interface conditions (1.4) can be approximated by

$$(w_{0,i}^{+}f_{i} + w_{0,i+1}^{+}u_{i+1} + w_{0,i+2}^{+}u_{i+2}) - (w_{0,i-1}^{-}u_{i-1} + w_{0,i}^{-}u_{i} + w_{0,i+1}^{-}f_{i+1}) = [u], \qquad (1.16)$$

$$\beta^{+}(w_{1,i}^{+}f_{i} + w_{1,i+1}^{+}u_{i+1} + w_{1,i+2}^{+}u_{i+2}) - \beta^{-}(w_{1,i-1}^{-}u_{i-1} + w_{1,i}^{-}u_{i} + w_{1,i+1}^{-}f_{i+1}) = [\beta u_{x}].$$
(1.17)

On the other hand, interface conditions (1.4) can also be approximated by using the *exactly* extended values  $u_i^+$  and  $u_{i+1}^-$ , respectively defined at  $x_i$  and  $x_{i+1}$ 

$$(w_{0,i}^{+}u_{i}^{+} + w_{0,i+1}^{+}u_{i+1} + w_{0,i+2}^{+}u_{i+2} + C_{0}^{+}h^{3}) - (w_{0,i-1}^{-}u_{i-1} + w_{0,i}^{-}u_{i} + w_{0,i+1}^{-}u_{i+1}^{-} + C_{0}^{-}h^{3}) = [u],$$

$$\beta^{+}(w_{1,i}^{+}u_{i}^{+} + w_{1,i+1}^{+}u_{i+1} + w_{1,i+2}^{+}u_{i+2} + C_{1}^{+}h^{2}) -$$

$$(1.18)$$

$$\beta^{-}(w_{1,i-1}^{-}u_{i-1} + w_{1,i}^{-}u_{i} + w_{1,i+1}^{-}u_{i+1}^{-} + C_{1}^{-}h^{2}) = [\beta u_{x}], \qquad (1.19)$$

where  $C_0^-, C_0^+, C_1^-$  and  $C_1^+$  are the coefficients of the leading truncation errors. Note that fictitious values  $f_i$  and  $f_{i+1}$  are the approximation to the exact values  $u_i^+$  and  $u_{i+1}^-$  on fictitious domains, respectively. We denote the difference between these fictitious values and the extended exact values as  $\epsilon_i$  and  $\epsilon_{i+1},$  i.e,

$$\epsilon_i = u_i^+ - f_i \tag{1.20}$$

$$\epsilon_{i+1} = u_{i+1} - f_{i+1}. \tag{1.21}$$

By subtracting Eq. (1.16) from Eq. (1.18), and Eq. (1.17) from Eq. (1.19), we notice that  $\epsilon_i$  and  $\epsilon_{i+1}$  satisfy the following two equations:

$$w_{0,i}^{+}\epsilon_{i} - \bar{w_{0,i+1}}\epsilon_{i+1} = (C_{0}^{-} - C_{0}^{+})h^{3}, \qquad (1.22)$$

$$\beta^{+}w_{1,i}^{+}\epsilon_{i} - \beta^{-}w_{1,i+1}^{-}\epsilon_{i+1} = (\beta^{-}C_{1}^{-} - \beta^{+}C_{1}^{+})h^{2}.$$
(1.23)

Solving for  $\epsilon_i$  we obtain

$$\epsilon_{i} = \frac{-\beta^{-}(C_{0}^{-} - C_{0}^{+})w_{1,i+1}^{-}h^{3} - w_{0,i+1}^{-}(C_{1}^{-}\beta^{-} - C_{1}^{+}\beta^{+})h^{2}}{-\beta^{-}w_{0,i}^{+}w_{1,i+1}^{-} + \beta^{+}w_{1,i}^{+}w_{0,i+1}^{-}}.$$
 (1.24)

Since  $w_{0,i+1}^- \sim O(1), w_{0,i}^+ \sim O(1), w_{1,i+1}^- \sim O(1/h)$  and  $w_{1,i}^+ \sim O(1/h)$ , it follows that  $\epsilon_i \sim O(h^3)$ . Moreover, since

$$\epsilon_{i} = \frac{-(C_{0}^{-} - C_{0}^{+})w_{1,i+1}^{-}h^{3} - w_{0,i+1}^{-}(C_{1}^{-} - C_{1}^{+}\frac{\beta^{+}}{\beta^{-}})h^{2}}{-w_{0,i}^{+}w_{1,i+1}^{-} + \frac{\beta^{+}}{\beta^{-}}w_{1,i}^{+}w_{0,i+1}^{-}},$$
(1.25)

as  $\beta^+/\beta^-$  increases,  $\epsilon_i$  asymptotically approaches  $C(h^3)$ , where the constant C depends only on  $C_1^+$  and  $w_{1,i}^+$ . In other words, for large jumps in the diffusion coefficient, the approximation error depends only on the solution of the problem. This proves that the present method is robust to a large contrast of  $\beta$  at the interface.

## Chapter 2

# MIB Method: 2-D Formulation and Numerical Experiments

This chapter is devoted to the formulation and test of the MIB scheme for solving 2-D elliptic equations with irregular (curved) interfaces. Compared with the treatment of interface in 1-D or a straight interface in 2-D, it is much more intricate to treat irregular interfaces because the local topology of irregular points varies from point to point. Therefore, the procedure outlined for regular interfaces cannot be directly applied. A second-order 2-D MIB scheme will be described first followed by its generalization to higher order schemes. We will then present an alternative interpolation formulation of the MIB method without the explicit use of fictitious domains. We show that the new formulation is essentially equivalent to the improved fictitious domain formulation. It is believed that this alternative formulation offers not only a better understanding of the MIB method, but also provides insights into other similar high order interface methods that might be constructed. Extensive numerical experiments are performed to demonstrate the high accuracy and robustness of the MIB method, as well as its flexibility in handling irregular interfaces.

## 2.1 Second-order 2-D MIB scheme for irregular interface

Before proceeding to the construction of the MIB scheme for the discretization of  $(\beta u_x)_x$  or  $(\beta u_y)_y$ , we have to identify the irregular grid points since only these points necessitate special care when the standard central difference scheme is applied to the whole domain. We define a grid point to be irregular if all the discretization points participating a standard central finite difference scheme at this point are not on the same side of the interface. For example, in a second order 2D scheme, an irregular point has at least one of its 4 nearest neighbor grid points lying on the other side of the interface. Note that the number of irregular points increases when a higher order FD scheme is employed.

Assume as before that there are two given conditions associated with the interface, i.e.,

$$[u] = u^{+} - u^{-} = \phi(x, y)$$
 (2.1)

$$[\beta u_n] = \beta^+ u_n^+ - \beta^- u_n^- = \psi(x, y)$$
(2.2)

and also assume that both  $\phi(x, y)$  and  $\psi(x, y)$  are  $C^1$  continuous along the interface  $\Gamma$ . When considering the interface which is not always aligned with the x- or y- mesh lines, as what shown by Figure 2.10, one more interface condition can be attained by differentiating Eq. (2.1) along the tangential direction of the interface, i.e.  $[u_{\tau}] = \phi_{\tau}(x, y)$ . Hence for a point  $(x_o, y_o)$  on the interface, we have three jump conditions,

$$[u] = u^{+} - u^{-} = \phi(x_{o}, y_{o})$$
(2.3)

$$[u_{\tau}] = u_{\tau}^{+} - u_{\tau}^{-} = \phi_{\tau}(x_{o}, y_{o})$$
(2.4)

$$[\beta u_n] = \beta^+ u^+ - \beta^- u^- = \psi(x_o, y_o).$$
(2.5)

where the normal vector of the interface is  $\vec{n} = (\cos \theta, \sin \theta)$  and  $u_{\tau}$  is the derivative

in the tangential direction  $\vec{\tau} = (-\sin\theta, \cos\theta)$ , while  $0 \le \theta < 2\pi$  is the angle between the positive x- direction and the vector  $\vec{n}$ . Considering these relations, these three local interface conditions can be reformulated as

$$[u] = u^{+} - u^{-} = \phi(x_{o}, y_{o})$$
(2.6)

$$[u_{\tau}] = (-u_x^+ \sin \theta + u_y^+ \cos \theta) - (-u_x^- \sin \theta + u_y^- \cos \theta) = \rho(x_o, y_o)$$
(2.7)

$$[\beta u_n] = \beta^+ (u_x^+ \cos\theta + u_y^+ \sin\theta) - \beta^- (u_x^- \cos\theta + u_y^- \sin\theta) = \psi(x_o, y_o).$$
(2.8)

In the MIB approach, the implementation of jump conditions is disassociated with



Figure 2.1: Three typical situations for an interface crossing the mesh lines. In the left chart, the interface passes the x- mesh line at point A and y- mesh line at point B. The interface conditions are approximated at point A and point B to generate the difference scheme for  $(\beta u_x)_x$  and  $(\beta u_y)_y$ , respectively. In the middle chart, the interface conditions are approximated at point A to obtain the difference scheme only for  $(\beta u_x)_x$  and the regular central difference is used for  $(\beta u_y)_y$ . In the right chart, the difference scheme for  $(\beta u_y)_y$  is set up from the approximation of the interface conditions at point B while term  $(\beta u_x)_x$  is treated with the regular central difference.

the discretization of the elliptic equation. Also  $(\beta u_x)_x$  and  $(\beta u_y)_y$  will be treated separately. Therefore, we only need to illustrate how to locally recover the second order accuracy of the standard 3-point finite difference scheme for  $(\beta u_x)_x$ . The modeling for  $(\beta u_y)_y$  can be achieved similarly. Considering an interface point  $o = (x_o, y_o)$ which is located at the intersection of interface  $\Gamma$  and the x mesh line, see Figure 2.2, one fictitious point on each side of  $(x_o, y_o)$  is required in the present MIB modeling. To estimate these two fictitious values we will discretize  $u^+, u^-, u_x^+$  and  $u_x^-$  involved in the jump conditions by using a 1-D grid partition as in the regular interface cases. However, it remains a problem to deal with the two derivatives  $u_y^+$  and  $u_y^-$  in the jump conditions (2.7) and (2.8) at the interface point  $(x_o, y_o)$ . We overcome this difficulty essentially via two steps. First, by using jump conditions (2.7) and (2.8), we eliminate one y derivative that is more difficult to discretize near the interface. Second, we carry out the discretization for the remaining y derivative by using a onesided FD scheme on the auxiliary points whose values are obtained by interpolation. This is usually possible in practice. If  $u_y^+$  is easier to be evaluated, we will cancel  $u_y^-$ 



**Figure 2.2:** Irregular point (i, j) and the interface. The interface crosses the *x*-mesh line at  $(x_0, y_0)$ . The fictitious values are  $f_{i,j}$  and  $f_{i+1,j}$  (green). The vertical dash line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (o, j+2), (o, j+1) and (o, j) right at  $(x_0, y_0)$ . The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at  $(x_0, y_0)$ .

from (2.7) and (2.8) to attain

$$[u] = u^{+} - u^{-}, \quad \text{and} \quad [\beta u_{n}] - \beta^{-} \tan \theta [u_{\tau}] = C_{x}^{+} u_{x}^{+} - C_{x}^{-} u_{x}^{-} + C_{y}^{+} u_{y}^{+}, \quad (2.9)$$

where  $C_x^+ = \beta^+ \cos \theta + \beta^- \tan \theta \sin \theta$ ,  $C_x^- = \beta^- \cos \theta + \beta^- \tan \theta \sin \theta$  and  $C_y^+ = \beta^+ \sin \theta - \beta^- \sin \theta$ . If otherwise  $u_y^-$  is easier to be calculated, one shall use the following jump conditions derived from Eqs. (2.6), (2.7) and (2.8) by canceling  $u_y^+$ :

$$[u] = u^{+} - u^{-}, \quad \text{and} \quad [\beta u_n] - \beta^{+} \tan \theta [u_{\tau}] = C_x^{+} u_x^{+} - C_x^{-} u_x^{-} - C_y^{-} u_y^{-}, \quad (2.10)$$

where  $C_x^+ = \beta^+ \cos \theta + \beta^+ \tan \theta \sin \theta$ ,  $C_x^- = \beta^- \cos \theta + \beta^+ \tan \theta \sin \theta$  and  $C_y^- = \beta^- \sin \theta - \beta^+ \sin \theta$ .

Cancellation of  $u_x^+$  or  $u_x^-$  from Eqs. (2.7) and (2.8) is required when modeling  $(\beta u_y)_y$ , which leads to two other combinations of jump conditions. One combination is

$$[u] = u^{+} - u^{-}, \quad \text{and} \quad [\beta u_{n}] + \beta^{-} \cot \theta [u_{\tau}] = C_{x}^{+} u_{x}^{+} + C_{y}^{+} u_{y}^{+} - C_{y}^{-} u_{y}^{-}, \quad (2.11)$$

where  $C_x^+ = (\beta^+ - \beta^-) \cos \theta$ ,  $C_y^+ = \beta^- \cos \theta \cot \theta + \beta^+ \sin \theta$  and  $C_y^- = \beta^- (\cos \theta \cot \theta + \sin \theta)$ ; and the other is

$$[u] = u^{+} - u^{-} \quad \text{and} \quad [\beta u_{n}] + \beta^{+} \cot \theta [u_{\tau}] = C_{x}^{-} u_{x}^{-} + C_{y}^{+} u_{y}^{+} - C_{y}^{-} u_{y}^{-}.$$
(2.12)

It is easy to check that here  $C_x^- = (\beta^+ - \beta^-) \cos \theta$ ,  $C_y^- = \beta^+ \cos \theta \cot \theta + \beta^- \sin \theta$ and  $C_y^+ = \beta^+ (\cos \theta \cot \theta + \sin \theta)$ . Either Eq. (2.11) or Eq. (2.12) can be chosen to formulate the discretization scheme for  $(\beta u_y)_y$ , depending on whether  $u_x^+$  or  $u_x^-$  is easier to be evaluated.

In the case that either  $\tan \theta$  or  $\cot \theta$  is undefined in choosing jump conditions, i.e., whenever the outer normal direction is aligned to the x- or y- direction, the interface is locally perpendicular to the mesh line so that it can be treated as a straight interface locally. The procedure introduced in Section (1.2) for the 1-D problem can be directly employed to handle these irregular points.

It is seen that for irregular interfaces each irregular point should have its own local MIB representation because of different local topology, while for regular interface cases, in contrast, only one global representation is needed throughout the whole domain. Some typical topology near irregular points is depicted in Figure 2.1.

We first consider the detailed discretization of jump conditions (2.9) at the interface point  $(x_o, y_o)$  for the topology given in Figure 2.2, where the interface  $\Gamma$  passes through point  $(x_o, y_o)$  between two irregular points (i, j) and (i + 1, j). Four nodes along the  $j^{th}$  mesh line, i.e., (i - 1, j), (i, j), (i + 1, j) and (i + 2, j), are required to approximate x-derivatives in jump conditions (2.9). In order to evaluate  $u_y^+$  at  $(x_o, y_o) = (x_o, y_j)$ , we add an auxiliary y- mesh line (dash line) through  $(x_o, y_o)$ . Three auxiliary grid points on this auxiliary line will be employed to approximate  $u_y^+$ . Two of these three points, (o, j + 1) and (o, j + 2), are on the positive side of the interface, while the third auxiliary point (o, j) is right on the interface.

Refer again to Figure 2.2, where point (i - 1, j) and (i, j) are on the same side of the interface while (i + 1, j), (i + 2, j), (o, j + 1) and (o, j + 2) are on the other side. We deploy two fictitious points at (i, j) and (i + 1, j), the corresponding fictitious unknowns are denoted as  $f_{i,j}$  and  $f_{i+1,j}$ , respectively. The two conditions in Eq. (2.9) are then approximated to be

$$w_{0,i-1}^{-}u_{i-1,j} + w_{0,i}^{-}u_{i,j} + w_{0,i+1}^{-}f_{i+1,j} + [u] = w_{0,i}^{+}f_{i,j} + w_{0,i+1}^{+}u_{i+1,j} + w_{0,i+2}^{+}u_{i+2,j}$$
(2.13)  
$$C_{x}^{-}(w_{1,i-1}^{-}u_{i-1,j} + w_{1,i}^{-}u_{i,j} + w_{1,i+1}^{-}f_{i+1,j}) + [\beta u_{n}] - \beta^{-} \tan \theta[u_{\tau}] = C_{x}^{+}(w_{1,i}^{+}f_{i,j} + w_{1,i+1}^{+}u_{i+1,j} + w_{1,i+2}^{+}u_{i+2,j}) + C_{y}^{+}(p_{1,j}^{+}u_{0,j}^{+} + p_{1,j+1}^{+}p_{0,j+1} + p_{1,j+2}^{+}u_{0,j+2})$$
(2.14)

where superscripts, - and +, signify that the FD approximation is on the - and + side of the interface, respectively. Here w and p are FD weights for approximation along the x- and y- directions, respectively. Their first subscript (0 or 1) represents that it is for interpolation or for the first-order differentiation, while their second subscript is the node index. For example,  $w_{0,i-1}^-, w_{0,i}^-$  and  $w_{0,i+1}^-$  are the interpolation weights of  $u(x_o, y_o)^-$  at grid points (i - 1, j), (i, j) and (i + 1, j).

As aforementioned,  $u_y^+$  shall be approximated on a one-sided finite difference stencil comprised of three auxiliary points, i.e.,  $u_{o,j}^+$ ,  $u_{o,j+1}$  and  $u_{o,j+2}$ , which need further numerical treatments. To relate  $u_{o,j}^+$  with function values at real grid points we adopt the relation

$$u_{o,j}^{+} = u_{o,j}^{-} + [u] = w_{0,i-1}^{-} u_{i-1,j} + w_{0,i}^{-} u_{i,j} + w_{0,i+1}^{-} f_{i+1,j} + [u].$$
(2.15)

The other two auxiliary values,  $u_{o,j+1}$  and  $u_{o,j+2}$  are interpolated on three grid points around them to ensure an accuracy of  $O(h^3)$ . These interpolation points are chosen *a priori*, and should be on the same side of the interface. For the geometry in Figure 2.2 we choose interpolation points to be (i, j + 1), (i + 1, j + 1), (i + 2, j + 1) and (i, j + 2), (i + 1, j + 2), (i + 2, j + 2) for  $u_{o,j+1}$  and  $u_{o,j+2}$ , respectively. As a result of these interpolations,  $u_{o,j+1}$  and  $u_{o,j+2}$  become known variables in jump condition (2.14).

There are only two unknowns in Eqs. (2.13) and (2.14), i.e., fictitious values  $f_{i,j}$ and  $f_{i+1,j}$ . To solve for the representation of these fictitious values in terms of the real function values and the known jumps, we introduce two expansions

$$f_{i,j} = \mathbf{C}^i \cdot \mathbf{U}, \tag{2.16}$$

$$f_{i+1,j} = \mathbf{C}^{i+1} \cdot \mathbf{U}, \qquad (2.17)$$

where  $\mathbf{C}^{i} = (\mathbf{C}_{1}^{i}, \mathbf{C}_{2}^{i}, \cdots, \mathbf{C}_{9}^{i})$  and  $\mathbf{C}^{i+1} = (\mathbf{C}_{1}^{i+1}, \mathbf{C}_{2}^{i+1}, \cdots, \mathbf{C}_{9}^{i+1})$  are the expansion coefficients of the two fictitious unknowns with respect to 6 real unknowns and 3 jumps which are also given by the vector

$$\mathbf{U} = (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{o,j+1}, u_{o,j+2}, [u], [\beta u_n], [u_\tau]).$$

Inserting these expansions of  $f_{i,j}$  and  $f_{i+1,j}$  to Eqs. (2.13) and (2.14), we end up with a small linear system for vectors  $\mathbf{C}^i$  and  $\mathbf{C}^{i+1}$ :

$$w_{0,i+1}^{-} \mathbf{C}^{i+1} \cdot \mathbf{U} - w_{0,i}^{+} \mathbf{C}^{i} \cdot \mathbf{U} = \mathbf{K}_{1} \cdot \mathbf{U} \quad (2.18)$$

$$(C_x^- w_{1,i+1}^- - C_y^+ p_{1,j}^+ w_{0,i+1}^-) \mathbf{C}^{i+1} \cdot \mathbf{U} - C_x^+ w_{1,i}^+ \mathbf{C}^i \cdot \mathbf{U} = \mathbf{K}_2 \cdot \mathbf{U}, \quad (2.19)$$

or equivalently

$$w_{0,i+1}^{-}\mathbf{C}^{i+1} - w_{0,i}^{+}\mathbf{C}^{i} = \mathbf{K}_{1}$$
 (2.20)

$$(C_x^- w_{1,i+1}^- - C_y^+ p_{1,j}^+ w_{0,i+1}^-) \mathbf{C}^{i+1} - C_x^+ w_{1,i}^+ \mathbf{C}^i = \mathbf{K}_2.$$
(2.21)
where

$$\begin{split} \mathbf{K}_{1} &= (-w_{0,i-1}^{-}, -w_{0,i}^{-}, w_{0,i+1}^{+}, w_{0,i+2}^{+}, 0, 0, -1, 0, 0), \\ \mathbf{K}_{2} &= (-C_{x}^{-}w_{1,i-1}^{-} + C_{y}^{+}p_{1,j}^{+}w_{0,i-1}^{-}, -C_{x}^{-}w_{1,i}^{-} + C_{y}^{+}p_{1,j}^{+}w_{0,i}^{-}, C_{x}^{+}w_{1,i+1}^{+}, C_{x}^{+}w_{1,i+2}^{+}, \\ & C_{y}^{+}p_{1,j+1}^{+}, C_{y}^{+}p_{1,j+2}^{+}, C_{y}^{+}p_{1,j}^{+}, -1, \beta^{-}\tan\theta). \end{split}$$

The solution of this  $2 \times 2$  linear system is readily available with Cramer's formula:

$$\mathbf{C}^{i} = \frac{w_{0,i+1}^{-}\mathbf{K}_{2} - (C_{x}^{-}w_{1,i+1}^{-} - C_{y}^{+}p_{1,j}^{+}w_{0,i+1}^{-})\mathbf{K}_{1}}{-C_{x}^{+}w_{1,i}^{+}w_{0,i+1}^{-} + w_{0,i}^{+}(C_{x}^{-}w_{1,i+1}^{-} - C_{y}^{+}p_{1,j}^{+}w_{0,i+1}^{-})}$$
(2.22)

$$\mathbf{C}^{i+1} = \frac{-C_x^+ w_{1,i}^+ \mathbf{K}_1 + w_{0,i}^+ \mathbf{K}_2}{-C_x^+ w_{1,i}^+ w_{0,3}^- + w_{0,i}^+ (C_x^- w_{1,i+1}^- - C_y^+ p_{1,j}^+ w_{0,i+1}^-)}.$$
 (2.23)

Before finishing the solution of the fictitious values, we need to distribute the expansion weights of  $f_{i,j}$  and  $f_{i+1,j}$  at auxiliary points  $u_{o,j+1}$  and  $u_{o,j+2}$  to their respective interpolation nodes defined above. Suppose

$$u_{o,j+1} = \mathbf{I}^{j+1} \cdot (u_{i,j+2}, u_{i+1,j+1}, u_{i+2,j+1})$$
$$u_{o,j+2} = \mathbf{I}^{j+2} \cdot (u_{i,j+1}, u_{i+1,j+1}, u_{i+2,j+1}),$$

where  $\mathbf{I}^{j+1} = (\mathbf{I}_1^{j+1}, \mathbf{I}_2^{j+1}, \mathbf{I}_3^{j+1})$  and  $\mathbf{I}^{j+2} = (\mathbf{I}_1^{j+2}, \mathbf{I}_2^{j+2}, \mathbf{I}_3^{j+2})$  are the interpolation coefficients in the vector form. The expansion weights of  $f_{i,j}$  and  $f_{i+1,j}$  could then be distributed onto these interpolation points and the final expressions of  $f_{i,j}$  and  $f_{i+1,j}$  are:

$$f_{i,j} = \mathbf{C}_{1}^{i} u_{i-1,j} + \mathbf{C}_{2}^{i} u_{i,j} + \mathbf{C}_{3}^{i} u_{i+1,j} + \mathbf{C}_{4}^{i} u_{i+2,j} + \mathbf{C}_{5}^{i} (\mathbf{I}_{1}^{j+1} u_{i,j+1} + \mathbf{I}_{2}^{j+1} u_{i+1,j+1} + \mathbf{I}_{3}^{j+1} u_{i+2,j+1}) + \mathbf{C}_{6}^{i} (\mathbf{I}_{1}^{j+2} u_{i,j+2} + \mathbf{I}_{2}^{j+2} u_{i+1,j+2} + \mathbf{I}_{3}^{j+2} u_{i+2,j+2}) + \mathbf{C}_{7}^{i} [u] + \mathbf{C}_{8}^{i} [\beta u_{n}] + \mathbf{C}_{9}^{i} [u_{\tau}]$$
(2.24)

$$f_{i+1,j} = \mathbf{C}_{1}^{i+1} u_{i-1,j} + \mathbf{C}_{2}^{i+1} u_{i,j} + \mathbf{C}_{3}^{i+1} u_{i+1,j} + \mathbf{C}_{4}^{i+1} u_{i+2,j} + \mathbf{C}_{5}^{i+1} (\mathbf{I}_{1}^{j+1} u_{i,j+1} + \mathbf{I}_{2}^{j+1} u_{i+1,j+1} + \mathbf{I}_{3}^{j+1} u_{i+2,j+1}) + \mathbf{C}_{6}^{i+1} (\mathbf{I}_{1}^{j+2} u_{i,j+2} + \mathbf{I}_{2}^{j+2} u_{i+1,j+2} + \mathbf{I}_{3}^{j+2} u_{i+2,j+2}) + \mathbf{C}_{7}^{i+1} [u] + \mathbf{C}_{8}^{i+1} [\beta u_{n}] + \mathbf{C}_{9}^{i+1} [u_{\tau}].$$
(2.25)

One can therefore discretize  $(\beta u_x)_x$  at irregular points (i, j) and (i + 1, j) as at the regular point:

$$(\beta u_x)_x = \frac{\beta^-_{i-\frac{1}{2},j}}{h^2} u_{i-1,j} - \frac{\beta^-_{i-\frac{1}{2},j} + \beta^-_{i+\frac{1}{2},j}}{h^2} u_{i,j} + \frac{\beta^-_{i+\frac{1}{2},j}}{h^2} f_{i+1,j} \quad \text{at } (i,j)$$
  
$$(\beta u_x)_x = \frac{\beta^+_{i+\frac{1}{2},j}}{h^2} f_{i,j} - \frac{\beta^+_{i+\frac{1}{2},j} + \beta^+_{i+\frac{3}{2},j}}{h^2} u_{i+1,j} + \frac{\beta^+_{i+\frac{3}{2},j}}{h^2} u_{i+2,j} \quad \text{at } (i+1,j)$$

by substituting the above expansions to  $f_{i,j}$  and  $f_{i+1,j}$ . The known terms involving given jumps  $[u], [u_{\tau}]$  and  $[\beta u_x]$  should be collected and combined into the right-hand vector B in the ultimate linear system  $A\mathbf{x} = B$  from which the elliptic equation is eventually solved.



**Figure 2.3:** Irregular point (i, j) and the interface. The interface crosses the y- mesh line at  $(x_o, y_o)$ . The dash horizontal line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (i - 2, o), (i - 1, o) and (i, o) right at  $(x_o, y_o)$ . The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at the intersect point  $(x_o, y_o)$ .

The difference scheme for  $(\beta u_y)_y$  can be generated following the procedure similar

to that for  $(\beta u_x)_x$ . Let (i, j) and (i, j + 1) be a pair of irregular points and between them the interface  $\Gamma$  intersects the  $i^{th}$  mesh line at  $(x_o, y_o)$ , see Figure 2.3. We need two fictitious points,  $f_{i,j}$  at point (i, j) and  $f_{i,j+1}$  at (i, j + 1), to facilitate the discretization of  $(\beta u_y)_y$  at (i, j) and (i, j+1). Jump conditions, Eq. (2.12), are chosen and approximated at the intersect point  $(x_o, y_o)$ . Considering the fact that we do not have grid points to directly approximate  $u_x^-$ , a horizontal auxiliary mesh line passing through  $(x_o, y_o)$  is introduced and three auxiliary points, (i-2, o), (i-1, o) and (i, o), are deployed on which a one-sided finite difference approximation for  $u_x^-$  is formulated. The jump conditions (2.12) are then replaced by the following approximate equations:

$$p_{0,j-1}^{-}u_{i,j-1} + p_{0,j}^{-}u_{i,j} + p_{0,j+1}^{-}f_{i,j+1} + [u] =$$

$$p_{0,j}^{+}f_{i,j} + p_{0,j+1}^{+}u_{i,j+1} + p_{0,j+2}^{+}u_{i,j+2}, \qquad (2.26)$$

$$C_{y}^{-}(p_{1,j-1}^{-}u_{i,j-1} + p_{1,j}^{-}u_{i,j} + p_{1,j+1}^{-}f_{i,j+1}) + [\beta u_{n}] + \beta^{+}\cot\theta[u_{\tau}] =$$

$$C_{y}^{+}(p_{1,j}^{+}f_{i,j} + p_{1,j+1}^{+}u_{i,j+1} + p_{1,j+2}^{+}u_{i,j+2}) + C_{x}^{-}(w_{1,i-2}^{-}u_{i-2,o} + w_{1,i-1}^{-}u_{i-1,o} + w_{1,i}^{-}u_{i,o}^{-}). \qquad (2.27)$$

The notations used here follow the same nomenclature as that in Eqs. (2.13) and (2.14). Since  $u_{i,o}^-$  is defined at an auxiliary point on the interface rather than at a grid point, we relate it with the unknowns at grid points by an interpolation:

$$u_{i,o}^{-} = u_{i,o}^{+} - [u] = p_{0,j}^{+} f_j + p_{0,j+1}^{+} u_{j+1} + p_{0,j+2}^{+} u_{j+2} - [u].$$
(2.28)

Nevertheless the interpolation on points  $u_{i,j-2}$ ,  $u_{i,j}$  and  $f_{i,j+2}$  is also applicable. With this relation, Eq. (2.27) is changed into

$$C_{y}^{-}(p_{1,j-1}^{-}u_{i,j-1} + p_{1,j}^{-}u_{i,j} + p_{1,j+1}^{-}f_{i,j+1}) + [\beta u_{n}] + \beta^{+} \cot \theta[u_{\tau}]$$

$$= C_{y}^{+}(p_{1,j}^{+}f_{i,j} + p_{1,j+1}^{+}u_{i,j+1} + p_{1,j+2}^{+}u_{i,j+2})$$

$$+ C_{x}^{-}[w_{1,i-2}^{-}u_{i-2,o} + w_{1,i-1}^{-}u_{i-1,o}$$

$$+ w_{1,i}^{-}(p_{0,j}^{+}f_{i,j} + p_{0,j+1}^{+}u_{i,j+1} + p_{0,j+2}^{+}u_{i,j+2} - [u])]. \qquad (2.29)$$

Also, we are interested in the representations of the two fictitious values  $f_{i,j}$  and  $f_{i,j+1}$  with respect to the known jumps and real unknowns, i.e.,

$$f_{i,j} = \mathbf{C}^j \cdot \mathbf{U} \tag{2.30}$$

$$f_{i,j+1} = \mathbf{C}^{j+1} \cdot \mathbf{U} \tag{2.31}$$

where  $\mathbf{U} = (u_{i,j-1}, u_{i,j}, u_{i,j+1}, u_{i,j+2}, u_{o,i-2}, u_{o,i-1}, [u], [\beta u_n], [u_{\tau}])$  and the components of vector  $\mathbf{C}^j, \mathbf{C}^{j+1}$  are the expansion coefficients of  $f_{i,j}$  and  $f_{i,j+1}$  with respect to  $\mathbf{U}$ , respectively. A 2×2 linear system for  $\mathbf{C}^i$  and  $\mathbf{C}^{j+1}$  can be obtained by replacing  $f_{i,j}$  and  $f_{i,j+1}$  in Eqs. (2.26) and (2.29) with above representations, as follows

$$p_{0,j+1}^{-} \mathbf{C}^{j+1} \cdot \mathbf{U} - p_{0,j}^{+} \mathbf{C}^{j} \cdot \mathbf{U} = \mathbf{K}_{1} \cdot \mathbf{U}$$
(2.32)  
$$C_{y}^{-} p_{1,j+1}^{-} \mathbf{C}^{j+1} \cdot \mathbf{U} - (C_{y}^{+} p_{1,j}^{+} + C_{x}^{-} w_{1,i}^{-} p_{0,j}^{+}) \mathbf{C}^{j} \cdot \mathbf{U} = \mathbf{K}_{2} \cdot \mathbf{U}.$$
(2.33)

It is easy to verify that

$$\begin{split} \mathbf{K}_{1} &= (-p_{0,j-1}^{-}, -p_{0,j}^{-}, p_{0,j+1}^{+}, p_{0,j+2}^{+}, 0, 0, -1, 0, 0) \\ \mathbf{K}_{2} &= (-C_{y}^{-} p_{1,j-1}^{-}, -C_{y}^{-} p_{1,j}^{-}, C_{y}^{+} p_{1,j+1}^{+} + C_{x}^{-} w_{1,i}^{-} p_{0,j+1}^{+}, \\ & C_{y}^{+} p_{1,j+2}^{+} + C_{x}^{-} w_{1,i}^{-} p_{0,j+2}^{+}, C_{x}^{-} w_{1,i-2}^{-}, C_{x}^{-} w_{1,i-1}^{-}, -C_{x}^{-} w_{1,i}^{-}, -1, -\beta^{+} \cot \theta). \end{split}$$

After dropping the vector **U** from the above system, we obtain the solution of the system again by Cramer's formula:

$$\mathbf{C}^{j} = \frac{-(C_{y}^{+}p_{1,j}^{+} + C_{x}^{-}w_{1,i}^{-}p_{0,j}^{+})\mathbf{K}_{1} + p_{0,j}^{+}\mathbf{K}_{2}}{C_{y}^{-}p_{1,j+1}^{-}p_{0,j}^{+} - p_{0,j+1}^{-}(C_{y}^{+}p_{1,j}^{+} + C_{x}^{-}w_{1,i}^{-}p_{0,j}^{+})}$$
(2.34)

$$\mathbf{C}^{j+1} = \frac{-C_y^- p_{1,j+1}^- \mathbf{K}_1 + p_{0,j+1}^- \mathbf{K}_2}{C_y^- p_{1,j+1}^- p_{0,j+1}^+ (C_y^+ p_{1,j}^+ + C_x^- w_{1,i}^- p_{0,j}^+)}.$$
 (2.35)

The expansions of  $\mathbf{C}^{j}$  and  $\mathbf{C}^{j+1}$  solved from this linear system involve the unknowns at auxiliary points  $u_{o,i-2}$  and  $u_{o,i-1}$ , which are not regular points, so we also need to distribute the weights on these two points to the regular grid points. Considering the local topology we choose grid points  $u_{i-1,j-1}$ ,  $u_{i-1,j}$  and  $u_{i-1,j+1}$ for the interpolation at auxiliary points  $u_{o,i-1}$ , and  $u_{i-2,j-1}$ ,  $u_{i-2,j}$  and  $u_{i-2,j+1}$  for  $u_{o,i-2}$ . With these two interpolation relations:

$$u_{o,i-1} = \mathbf{I}_{1}^{i-1} u_{i-1,j-1} + \mathbf{I}_{2}^{i-1} u_{i-1,j} + \mathbf{I}_{3}^{i-1} u_{i-1,j+1}, \qquad (2.36)$$

$$u_{o,i-2} = \mathbf{I}_1^{i-2} u_{i-2,j-1} + \mathbf{I}_2^{i-2} u_{i-2,j} + \mathbf{I}_3^{i-2} u_{i-2,j+1},$$
(2.37)

it turns out that both  $f_{i,j}$  and  $f_{i,j+1}$  have the expansions in terms of unknowns at regular grid points and known jumps, as below

$$\begin{split} f_{i,j} &= \mathbf{C}_{1}^{j} u_{i,j-1} + \mathbf{C}_{2}^{j} u_{i,j} + \mathbf{C}_{3}^{j} u_{i,j+1} + \mathbf{C}_{4}^{j} u_{i,j+2} \\ &+ \mathbf{C}_{5}^{j} (\mathbf{I}_{1}^{i-1} u_{i-1,j-1} + \mathbf{I}_{2}^{i-1} u_{i-1,j} + \mathbf{I}_{3}^{i-1} u_{i-1,j+1}) \\ &+ \mathbf{C}_{6}^{j} (\mathbf{I}_{1}^{i-2} u_{i-2,j-1} + \mathbf{I}_{2}^{i-2} u_{i-2,j} + \mathbf{I}_{3}^{i-2} u_{i-2,j+1}) \\ &+ \mathbf{C}_{7}^{j} [u] + \mathbf{C}_{8}^{j} [\beta u_{n}] + \mathbf{C}_{9}^{j} [u_{\tau}] \end{split}$$
(2.38)  
$$f_{i,j+1} &= \mathbf{C}_{1}^{j+1} u_{i,j-1} + \mathbf{C}_{2}^{j+1} u_{i,j} + \mathbf{C}_{3}^{j+1} u_{i,j+1} + \mathbf{C}_{4}^{j+1} u_{i,j+2} \\ &+ \mathbf{C}_{5}^{j+1} (\mathbf{I}_{1}^{i-1} u_{i-1,j-1} + \mathbf{I}_{2}^{i-1} u_{i-1,j} + \mathbf{I}_{3}^{i-1} u_{i-1,j+1}) \\ &+ \mathbf{C}_{6}^{j+1} (\mathbf{I}_{1}^{i-2} u_{i-2,j-1} + \mathbf{I}_{2}^{i-2} u_{i-2,j} + \mathbf{I}_{3}^{i-2} u_{i-2,j+1}) \\ &+ \mathbf{C}_{7}^{j+1} [u] + \mathbf{C}_{8}^{j+1} [\beta u_{n}] + \mathbf{C}_{9}^{j+1} [u_{\tau}]. \end{aligned}$$
(2.39)

With these two fictitious values, the discretization of  $u_{yy}$  at point (i, j) and (i, j + 1)becomes straightforward

$$(\beta u_y)_y = \frac{\beta_{j-\frac{1}{2}}^-}{(\Delta y)^2} u_{i,j-1} - \frac{\beta_{j+\frac{1}{2}}^- + \beta_{j-\frac{1}{2}}^-}{(\Delta y)^2} u_{i,j} + \frac{\beta_{j+\frac{1}{2}}^-}{(\Delta y)^2} f_{j+1} \quad \text{at } (i,j)$$
  
$$(\beta u_y)_y = \frac{\beta_{i,j-\frac{1}{2}}^+}{(\Delta y)^2} u_{i,j} - \frac{\beta_{i+\frac{1}{2}}^+ + \beta_{i-\frac{1}{2}}^+}{(\Delta y)^2} u_{i,j} + \frac{\beta_{i+\frac{1}{2}}^+}{(\Delta y)^2} f_{j+1} \quad \text{at } (i,j+1).$$

The known terms involving  $[u], [u_{\tau}]$  and  $[\beta u_n]$  should also be collected and combined into the vector B in the final linear system Ax = B.

The finite difference schemes constructed above are applicable regardless whether

the intersect point of the interface with the mesh line is at the grid point. However, significant simplification can be obtained if the intersect point is a grid point. When it comes to such a case, the difference scheme for  $(\beta u_x)_x + (\beta u_y)_y$  can be generated in a single run rather than separately.



**Figure 2.4:** Irregular point (i, j) and the interface. The interface passes the grid point (i, j). No auxiliary lines and auxiliary points are needed. The jumps  $[u], [\beta u_n]$  and  $[u_{\tau}]$  are evaluated at the point (i, j).

Figure 2.4 shows such a situation where grid point (i, j) is an intersection point. Note that when  $[u] \neq 0$  the function on the interface is not well defined but from the computational point of view the interface itself can be treated either as in the interior of the interface or outside it. Here we regard it as in the interior of the interface, i.e.  $u(i, j) = u^-(i, j)$  if the grid point (i, j) is on the interface. In such a case, one only needs to take care of the difference scheme at point (i, j) since both (i - 1, j) and (i, j - 1) are now regular points. For (i + 1, j) and (i, j + 1), the jump in [u] can be directly incorporated to generate the difference schemes

$$(\beta u_x)_x = \frac{\beta_{i+\frac{3}{2},j}^+}{(\Delta x)^2} u_{i+2,j} - \frac{\beta_{i+\frac{3}{2},j}^+ + \beta_{i+\frac{1}{2},j}^+}{(\Delta x)^2} u_{i+1,j} + \frac{\beta_{i+\frac{1}{2},j}^+}{(\Delta x)^2} (u_{i,j} + [u])$$
(2.40)

at (i+1, j) and

$$(\beta u_y)_y = \frac{\beta_{i,j+\frac{3}{2}}^+}{(\Delta y)^2} u_{i,j+2} - \frac{\beta_{i,j+\frac{3}{2}}^+ + \beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} u_{i,j+1} + \frac{\beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} (u_{i,j} + [u])$$
(2.41)

at (i, j+1). For point (i, j) on the interface, however, two fictitious values,  $f_{i,j+1}$  and  $f_{i+1,j}$ , are required in order to formulate the difference scheme for  $(\beta u_x)_x + (\beta u_y)_y$  at the point (i, j). The expansion coefficients of these two fictitious values with respect to the unknowns at surrounding grid points and the known jumps would also be solved from the approximation equations of the jump conditions at point (i, j). However, since the grid point is now on the interface, the interface conditions adopted are slightly different from what we used earlier. On the one hand, the Lagrange interpolation representation of the interface relation for u, Eq. (2.6), does not involve either  $f_{i+1,j}$  or  $f_{i,j+1}$  at all. Thus this jump condition can not provide an approximate equation for the fictitious values. On the other hand, two targeted fictitious values make the approximation possible for all derivatives,  $u_x^+, u_x^-, u_y^+$  and  $u_y^-$ . Therefore the cancellation of one of Eqs. (2.7) and (2.8) is no longer necessary and these two conditions exactly provide two approximate equations for  $f_{i,j+1}$  and  $f_{i+1,j}$ , as follows

$$-(w_{1,i-1}^{-}u_{i-1,j} + w_{1,i}^{-}u_{i,j} + w_{1,i+1}^{-}f_{i+1,j})\sin\theta$$

$$+ (p_{1,j-1}^{-}u_{i,j-1} + p_{1,j}^{-}u_{i,j} + p_{1,j+1}^{-}f_{i,j+1})\cos\theta + [u_{\tau}]$$

$$= -[w_{1,i}^{+}(u_{i,j} + [u]) + w_{1,i+1}^{+}u_{i+1,j} + w_{1,i+2}^{+}u_{i+2,j}]\sin\theta$$

$$+ [p_{1,j}^{+}(u_{i,j} + [u]) + p_{1,j+1}^{+}u_{i,j+1} + p_{1,j+2}^{+}u_{i,j+2}]\cos\theta, \qquad (2.42)$$

$$\beta^{-}(w_{1,i-1}^{-}u_{i-1,j} + w_{1,i}^{-}u_{i,j} + w_{1,i+1}^{-}f_{i+1,j})\cos\theta$$

$$+ \beta^{-}(p_{1,j-1}^{-}u_{i,j-1} + p_{1,j}^{-}u_{i,j} + p_{1,j+1}^{-}f_{i,j+1})\sin\theta + [\beta u_{n}]$$

$$= \beta^{+}[w_{1,i}^{+}(u_{i,j} + [u]) + w_{1,i+1}^{+}u_{i+1} + w_{1,i+2}^{+}u_{i+2})\cos\theta$$

$$+ \beta^{+}(p_{1,j}^{+}(u_{i,j} + [u]) + p_{1,j+1}^{+}u_{i,j+1} + p_{1,j+2}^{+}u_{i,j+2})\sin\theta. \qquad (2.43)$$

These two equations construct a  $2 \times 2$  linear system for vectors  $\mathbf{C}^{i+1}$  and  $\mathbf{C}^{j+1}$ ,

which are the respective expansion coefficients of fictitious values  $f_{i+1,j}$  and  $f_{i,j+1}$ , i.e.  $f_{i+1,j} = \mathbf{C}^{i+1} \cdot \mathbf{U}, f_{i,j+1} = \mathbf{C}^{j+1} \cdot \mathbf{U}$  with

$$\mathbf{U} = (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i,j-1}, u_{i,j+1}, u_{i,j+2}, [u], [\beta u_n], [u_{\tau}]).$$

The solution of this linear system is

$$-w_{1,i+1}^{-}\sin\theta\mathbf{C}^{i+1} + p_{1,j+1}^{-}\cos\theta\mathbf{C}^{j+1} = \mathbf{K}_{1}, \qquad (2.44)$$

$$\beta^{-} w_{1,i+1}^{-} \cos \theta \mathbf{C}^{i+1} + \beta^{-} p_{1,j+1}^{-} \sin \theta \mathbf{C}^{j+1} = \mathbf{K}_{2}, \qquad (2.45)$$

where

$$\begin{split} \mathbf{K_1} &= (w_{1,i-1}^- \sin\theta, (w_{1,i}^- - w_{1,i}^+) \sin\theta + (p_{1,j}^+ - p_{1,j}^-) \cos\theta, \\ &- w_{1,i+1}^+ \sin\theta, -w_{1,i+2}^+ \sin\theta, -p_{1,j-1}^- \cos\theta, p_{1,j+1}^+ \cos\theta, p_{1,j+2}^+ \cos\theta, \\ &p_{1,j}^+ \cos\theta - w_{1,i}^+ \sin\theta, 0, -1) \\ \mathbf{K_2} &= (-\beta^- w_{1,i-1}^- \cos\theta, (\beta^+ w_{1,i}^+ - \beta^- w_{1,i}^-) \cos\theta + (\beta^+ p_{1,j}^+ - \beta^- p_{1,j}^-) \sin\theta, \\ &\beta^+ w_{1,i+1}^+ \cos\theta, \beta^+ w_{1,i+2}^+ \cos\theta, -\beta^- p_{1,j-1}^- \sin\theta, \beta^+ p_{1,j+1}^+ \sin\theta, \beta^+ p_{1,j+2}^+ \sin\theta, \\ &\beta^+ (w_{1,i}^+ \cos\theta + p_{1,j}^+ \sin\theta), -1, 0), \end{split}$$

and

$$\mathbf{C}^{i+1} = -\frac{\beta^{-} \bar{p_{1,j+1}} \sin \theta \mathbf{K}_{1} - \bar{p_{1,j+1}} \cos \theta \mathbf{K}_{2}}{\beta^{-} \bar{w_{1,i+1}} \bar{p_{1,j+1}}}$$
(2.46)

$$\mathbf{C}^{j+1} = \frac{\beta^{-} w_{1,i+1}^{-} \cos \theta \mathbf{K}_{1} + w_{1,i+1}^{-} \sin \theta \mathbf{K}_{2}}{\beta^{-} w_{1,i+1}^{-} p_{1,j+1}^{-}}.$$
 (2.47)

Thus we obtain the full representations of  $f_{i+1,j}$  and  $f_{i,j+1}$ 

$$f_{i+1,j} = \mathbf{C}_{1}^{i+1}u_{i-1,j} + \mathbf{C}_{2}^{i+1}u_{i,j} + \mathbf{C}_{3}^{i+1}u_{i+1,j} + \mathbf{C}_{4}^{i+1}u_{i+2,j} + \mathbf{C}_{5}^{i+1}u_{i,j+1} + \mathbf{C}_{6}^{i+1}u_{i,j+2} + \mathbf{C}_{7}^{i+1}[u] + \mathbf{C}_{8}^{i+1}[u_{\tau}] + \mathbf{C}_{9}^{j+1}[\beta u_{n}]$$
(2.48)  
$$f_{i,j+1} = \mathbf{C}_{1}^{j+1}u_{i-1,j} + \mathbf{C}_{2}^{j+1}u_{i,j} + \mathbf{C}_{3}^{j+1}u_{i+1,j} + \mathbf{C}_{4}^{j+1}u_{i+2,j} + \mathbf{C}_{5}^{j+1}u_{i,j+1} + \mathbf{C}_{6}^{j+1}u_{i,j+2} + \mathbf{C}_{7}^{j+1}[u] + \mathbf{C}_{8}^{j+1}[u_{\tau}] + \mathbf{C}_{9}^{j+1}[\beta u_{n}]$$
(2.49)

and the difference scheme for  $(\beta u_x)_x + (\beta u_y)_y$  at the point (i, j) is

$$(\beta u_x)_x + (\beta u_y)_y = \frac{\beta_{i+\frac{1}{2},j}^-}{(\Delta x)^2} f_{i+1,j} + \frac{\beta_{i+\frac{1}{2},j}^- + \beta_{i-\frac{1}{2},j}^-}{(\Delta x)^2} u_{i,j} + \frac{\beta_{i-\frac{1}{2},j}^-}{(\Delta x)^2} u_{i-1,j} + \frac{\beta_{i,j+\frac{1}{2}}^-}{(\Delta y)^2} f_{i,j+1,j} + \frac{\beta_{i,j+\frac{1}{2}}^- + \beta_{i,j-\frac{1}{2}}^-}{(\Delta y)^2} u_{i,j} + \frac{\beta_{i,j-\frac{1}{2},j}^-}{(\Delta y)^2} u_{i,j-1}.$$
(2.50)

It is noted that the representation scheme described above has a local truncation error of  $O(h^3)$  so that the central finite difference scheme for the second-order derivatives have a local truncation error of O(h) at irregular points. Because the number of irregular points is of O(N) while the number of the total grid points is of  $O(N^2)$ , the global approximation error of the proposed MIB method is dominated by the error of  $O(h^2)$  at regular points, which has been verified in our numerical experiments.

**Comments 2.1.1.** The placement of auxiliarly points is flexible in the MIB method and it is not necessary to use a one-side difference scheme to approximate the partial derivative that is not in the primary direction. For example, considering the topology of the interface and the distribution of grid points in Figure 2.5, we can place two auxiliary points on the opposite side of the x-mesh line  $y = y_j$  hence  $y_x^+$  can be approximated with a central finite difference on these auxiliary points, in contrast to the situation in Figure 2.2 where both auxiliary points are placed on the same side of the mesh line. In this case, the expansion weights of fictitious values on  $u_{o,j-1}$  and  $u_{o,j+1}$  will be distributed to u(i, j - 1), u(i + 1, j - 1), u(i + 2, j - 1) and u(i, j + 1), u(i + 1, j + 1), u(i + 2, j + 1), respectively.



**Figure 2.5:** Irregular point (i, j) and the interface. The interface crosses the xmesh line at  $(x_0, y_0)$ . The fictitious values are  $f_{i,j}$  and  $f_{i+1,j}$  (green). The vertical dashed line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (o, j - 1), (o, j + 1) and (o, j) right at  $(x_0, y_0)$ . The jumps  $[u], [\beta u_n]$  and  $[u_\tau]$ are evaluated at  $(x_0, y_0)$ .

**Comments 2.1.2.** In case that the interface crosses the mesh line at a grid point, it is possible to solve the fictitious values rather than to formulate the difference scheme at that grid point directly as did in Eq. (2.42) to Eq. (2.50). For three fictitious values  $f_{i,j}, f_{i+1,j}$  and  $f_{i,j+1}$  in Figure 2.4,  $f_{i,j}$  can be directly represented as  $u_{i,j} = [u]$  since it is on the grid point (i, j) which is regarded as inside of the interface by default. Therefore the fictitious value  $f_{i+1,j}$  is be determined using the second relation in  $Eq.(2.9), i.e., [\beta u_n] - \beta^- \tan \theta[u_\tau] = C_x^+ u_x^+ - C_x^- u_x^- + C_y^+ u_y^+$ . The approximation to this relation presents an equation for  $f_{i+1,j}$  as follows

$$[\beta u_n] - \beta^- \tan \theta[u_\tau] - C_y^+ \left( p_{1,j}^+(u_{i,j} + [u]) + p_{1,j+1}^+ u_{i,j+1} + p_{i,j+2}^+ u_{i,j+2} \right) = C_x^+(w_{1,i}^+(u_{i,j} + [u]) + w_{1,i+1}^+ u_{i+1,j} + w_{i+2,j}^+) - C_x^-(w_{1,i-1}^+ u_{i-1,j} + w_{1,i+1}^+ f_{i+1,j}). \quad (2.51)$$

The third fictitious value  $f_{i,j+1}$  is solved from the approximation to the second relation

in Eq.(2.11)

$$[\beta u_n] + \beta^- \cot \theta[u_\tau] - C_x^+ \left( w_{1,i}^+(u_{i,j} + [u]) + w_{1,i+1}^+ u_{i+1,j} + w_{i+2,j}^+ u_{i+2,j} \right) = C_y^+(p_{1,j}^+(u_{i,j} + [u]) + p_{1,j+1}^+ u_{i,j+1} + p_{i,j+2}^+) - C_y^-(p_{1,j-1}^+ u_{i,j-1} + p_{1,j}^+ u_{i,j} + p_{1,j+1}^+ f_{i,j+1}).$$
(2.52)

The central finite difference scheme at grid point (i, j) can then be formulated using these fictitious values.

# 2.2 Higher order 2-D MIB scheme for irregular interfaces



**Figure 2.6:** A typical stencil used in constructing a fourth-order scheme for  $u_{xx}$  and  $u_x$ . There are two pairs of fictitious points in this case:  $f_{i,j}, f_{i+1,j}$  and  $f_{i-1,j}, f_{i+2,j}$ .

The generalization of the proposed MIB method to higher order convergence is quite straightforward. In the present approach, a high order scheme means the use of standard high order finite difference discretization in the whole computational domain. In the vicinity of the interface, irregular grid points are first identified according to the finite difference discretization. Fictitious points are then created at the same locations of the irregular points. The function values at fictitious points are determined by enforcing the interface jump conditions. For high order schemes, the same set of interface conditions is repeatedly used to determine the required set of fictitious values. At each irregular grid point, this procedure is exactly the same as what was described for the straight interfaces.



**Figure 2.7:** A typical stencil used in constructing a sixth order scheme for  $u_{xx}$  and  $u_x$ . Now there are three pairs of fictitious points in this case:  $f_{i,j}, f_{i+1,j}$  and  $f_{i-1,j}, f_{i+2,j}$  as well as  $f_{i-2,j}, f_{i+3,j}$ .

Let us consider a fourth-order case. We start with the interface condition (2.9) and the stencil shown in Figure 2.6 to derive a fourth-order scheme for  $u_{xx}$  and  $u_x$ by using following steps:

- 1. Use  $u_{i-3,j}, \dots, u_{i,j}$  and  $f_{i+1,j}$  as a stencil to approximate  $u^-$  and  $u_x^-$ . For  $u^+$ and  $u_x^+$  we choose the stencil  $f_{i,j}, u_{i+1,j}, \dots, u_{i+4,j}$ . Here,  $u_y^+$  is discretized on auxiliary points  $(o, j), \dots, (o, j+4)$ , to ensure an accuracy of  $O(h^5)$ .
- 2. Solve the  $2 \times 2$  linear system resulting from Step 1 for the representations of

 $f_{i,j}$  and  $f_{i+1,j}$ .

- Use u<sub>i-3,j</sub>,..., u<sub>i,j</sub> and f<sub>i+1,j</sub>, f<sub>i+2,j</sub> to approximate u<sup>-</sup> and u<sub>x</sub><sup>-</sup>, and use f<sub>i-1,j</sub>, f<sub>i,j</sub> and u<sub>i+1,j</sub>,..., u<sub>i+4,j</sub> to approximate u<sup>+</sup> and u<sub>x</sub><sup>+</sup>. Use the same approximation for u<sub>y</sub><sup>+</sup> as Step 1. Note that at this moment, both f<sub>i,j</sub> and f<sub>i+1,j</sub> are known.
- 4. Solve the 2 × 2 linear system resulting from Step 3 for the representations of  $f_{i-1,j}$  and  $f_{i+2,j}$ .
- 5. Substitute appropriate terms of  $f_{i-1,j}$ ,  $f_{i,j}$ ,  $f_{i+1,j}$ ,  $f_{i+2,j}$  for the values at irregular points when the standard fourth-order central discretization of  $u_x$  or  $u_{xx}$ crosses the interface.

This iterative procedure is systematic and is of arbitrarily high order provided there are enough number of regular points available on both sides of the interface. For example, a sixth order discretization of  $u_{xx}$  and  $u_x$  can be constructed by using the stencil illustrated in Figure 2.7. In this case, we successively solve the representations for three pairs of fictitious values  $(f_{i,j}, f_{i+1,j})$ ,  $(f_{i-1,j}, f_{i+2,j})$  and  $(f_{i-2,j}, f_{i+3,j})$ . It is noted that in constructing high order MIB methods, one needs to ensure that the local truncation error at the irregular point is one order lower than the designed global order of accuracy. The nature of high order accuracy of these schemes can be appreciated from the numerical examples presented in Section 2.5.

### 2.3 Subtleness in domain extensions

We have not yet specified in sufficient detail how to determine fictitious values for a given topology of irregular points, which is a subtle issue indeed. At the first glance, fictitious values have to be solved along the direction of discretization, as they are needed in this discretization. In this regard, a fictitious value at a given point varies when it is obtained via smooth extension from different directions, because it is solved from the smooth continuation *only* in the direction associated with the original discretization. According to this association between the domain extension and the

discretization,  $f_{i+1,j}$  should only be solved along the x-direction, see Figure 2.8. Similarly,  $f_{i,j-1}$  should be determined twice, along the x- and y-directions, respectively. Unfortunately,  $f_{i,j-1}$  cannot be solved along the y-direction up to fourth-order accuracy due to the large curvature in Figure 2.8. It is this difficulty that has severely constrained the applicability of the MIB scheme in Ref. [96], where only one generic elliptic interface problem with relatively low curvatures was solved up to fourth-order accuracy.



**Figure 2.8:** Fictitious value  $f_{i,j+2}$  shall be used for the discretization of  $u_y, u_{yy}$  at grid point (i, j + 1). However,  $f_{i,j+2}$  can not be solved along the y-direction but it can be solved along x-direction, i.e., solved together with fictitious values  $f_{i-2,j+2}, f_{i-1,j+2}$  and  $f_{i+1,j+2}$ .

A close look at the fictitious values in Figure 2.9 reveals that the fictitious values must fall into one of the following five categories:

- Those can be solved only along one of the x- and y-directions and will be used for the discretization along the same direction.
- 2. Those can be solved only along one of the x- and y-directions, but will be used for the discretization along the other direction.
- 3. Those can be solved only along one of the x- and y-directions, and will be used for the discretization along both directions.

- 4. Those can be solved along both the x- and y-directions, and will be used for the discretization along either one, or both directions.
- 5. Those can not be solved along any direction, but will be used for the discretization along one or both directions.

It is seen that a fictitious value in either the second, the third or the fifth category would be deemed as unsolvable. As seen in Figure 2.8, to find four fictitious values  $f_{i,j}, \dots, f_{i,j+3}$  along the y-direction, grid points  $(x_i, y_{j-2})$  through  $(x_i, y_{j+1})$  are supposed to be on same side of the interface, whereas the other four grid points,  $(x_i, y_{j+2})$  through  $(x_i, y_{j+5})$ , must be on the other side. Similarly, to find four fictitious values  $f_{i,j-2}, \dots, f_{i,j+1}$ , by considering the extension in the y-direction, we need points  $(x_i, y_{j-4})$  through  $(x_i, y_{j-1})$  on one side and  $(x_i, y_j)$  through  $(x_i, y_{j+3})$ on the other side. It is obvious that the distribution of grid points with respect to the interface in Figure 2.8 does not satisfy such an assumption. In particular, it is unable to find fictitious values  $f_{i,j-2}$  through  $f_{i,j+3}$  by the extension in the y-direction along the *i*<sup>th</sup> mesh line since there are only two grid points in one subdomain on that mesh line. According to the above discussion, however, at least four grid points in either subdomain are needed in order to define a fourth-order extension. These fictitious values can be classified into the third type, and were unsolvable by using the practice of Ref. [96].

To enhance the flexibility of MIB method in handling irregular interfaces, we introduce a new concept, the disassociation between domain extension and discretization. This concept is established based on the following error analysis. If the approximation errors of fictitious values at a given point  $(x_i, y_j)$  obtained from the x- and y-directions are both of  $O(h^n)$  for a positive n, the difference between these two fictitious values must be of  $O(h^n)$ . Therefore, a fictitious value at an irregular point  $(x_i, y_j)$ , regardless the direction in its calculation, can also be used for the discretization in any direction involving the grid point. For example, fictitious values  $f_{i,j}$  through  $f_{i,j+3}$ are essentially the extension of the inner or the outer subdomain. Although they cannot be calculated through the extension in the y-direction, they indeed can be



Figure 2.9: The distribution of fictitious points (stars) for a  $40 \times 40$  mesh with irregular interface. Each fictitious domain roughly has two layers of fictitious points, supporting a fourth-order central difference scheme. The green stars represent the smooth continuation of the interior subdomain and the red stars denote the smooth continuation of the outer subdomain.

obtained by the extension in the x-direction. Particularly, one can show that  $f_{i,j+2}$  can be calculated considering the extension along the (j + 2)<sup>th</sup> horizontal mesh line. The known  $f_{i,j+2}$  can be used to facilitate the discretization, not only for  $u_x$  and  $u_{xx}$  at  $(x_{i-2}, y_{j+2})$  and  $(x_{i-1}, y_{j+2})$ , resepctively, but also for  $u_y$  and  $u_{yy}$  at  $(x_i, y_j)$  and  $(x_i, y_{j+1})$ , respectively. This new understanding makes it possible to solve the fictitious values of the second and third types, and significantly broadens the applicability of the MIB method to general interface geometry.

For the fictitious values of the fourth type, i.e., those that can be extended in both the x- and y-directions, such as  $f_{i-1,j-2}$ ,  $f_{i-1,j+3}$  in Figure 2.8, we may attain their values in the most convenient manner and use them for necessary discretization.

Finally, fictitious values of the fifth type may be made available in most cases by refining the mesh. For example, if there are only four grid points inside a circular interface, fictitious values cannot be solved for all higher order schemes, i.e., orders higher than two, on such a grid. However, when the grid size is doubled in both directions, the fourth-order fictitious values can be obtained. For this reason, the present MIB method is a robust high order approach for all curved interfaces. It was claimed in Ref. [96] that the proposed MIB scheme is of arbitrarily high order in principle. Indeed, the MIB procedure is systematic and it encompasses a variety of higher order schemes. However, the higher order convergence was demonstrated only for an interface problem on a special geometry due to the association between the domain extension and the discretization. It is believed that with the present understanding and extension technique, we are able to fully realize the higher-order potential of the MIB method for arbitrarily curved geometry.

It is noted that for a general interface the difference scheme generated by the MIB involves more nodes than the IIM and of course, more than the standard central difference scheme. The number of grid points and the distribution of the MIB stencils vary with the underlying mesh and the local extension of the interface relative to the mesh. Also, the resulting linear system is no longer symmetric and diagonally dominant, as in the case of the IIM. Nevertheless, all the iterative solvers we tested, including the successive over relaxation (SOR) or the preconditioned biconjugate gradient (PBCG), always yield the solution with a favorable convergence rate. Detailed comparison in terms of accuracy and CPU time is given in Section 2.5.

## 2.4 Interpolation formulation

The MIB method described in the preceding sections makes use of fictitious domains and values for the standard high order finite difference discretization of the governing equation near the interface. The fictitious values give a smooth continuation of the solution across the interface. Interface jump conditions are iteratively used to determine fictitious values. In this section, the possibility of the smooth continuation in terms of polynomial expansions is investigated. This interpolation formulation does not require the use of fictitious domains and values. To illustrate the idea, we start our discussion with a 1-D problem. The general principle is then applied to 2-D problems with curved interfaces. Particular attentions are paid to the relation between these formulations.

#### 2.4.1 One dimensional formalism

Consider the 1-D elliptic problem given in Eq. (1.3). The interface is located at  $x_i \leq a \leq x_{i+1}$ . In a second-order central finite difference scheme,  $x_i$  is the irregular point to the left of the interface, and  $x_{i+1}$  the irregular point to the right. On each side of the interface, we define a second-order polynomial

$$u^{-}(x) = a_{0}^{-} + \frac{a_{1}^{-}}{h}(x - x_{i}) + \frac{a_{2}^{-}}{h^{2}}(x - x_{i})^{2}, \qquad (2.53)$$

$$u^{+}(x) = a_{0}^{+} + \frac{a_{1}^{+}}{h}(x - x_{i+1}) + \frac{a_{2}^{+}}{h^{2}}(x - x_{i+1})^{2}, \qquad (2.54)$$

where the polynomial coefficients are scaled by h to reduce the condition number of the coefficient matrix. Obviously,  $a_0^- = u_i$  and  $a_0^+ = u_{i+1}$  and the remaining four coefficients can be determined from the expansion of the polynomials at two grid points, and from two interface jump conditions

$$-a_{1}^{-} + a_{2}^{-} = u_{i-1} - u_{i} \tag{2.55}$$

$$a_1^+ + a_2^+ = u_{i+2} - u_{i+1}$$
 (2.56)

$$(-a_1^+ x_r + a_2^+ x_r^2) - (a_1^- x_l + a_2^- x_l^2) = [u] - u_{i+1} + u_i$$
(2.57)

$$\beta^{+}(a_{1}^{+} - 2a_{2}^{+}x_{r}) - \beta^{-}(a_{1}^{-} + 2a_{2}^{-}x_{l}) = h[\beta u_{x}]$$
(2.58)

where  $x_l = \frac{a - x_i}{h}$ ,  $x_r = \frac{x_{i+1} - a}{h}$ . The first two equations, (2.55) and (2.56), are the realizations of  $u^-(x)$  at  $x_{i-1}$ , and  $u^+(x)$  at  $x_{i+2}$ , respectively. The last two equations, (2.57) and (2.58), are the approximations of the interface jump conditions with  $u^-(x)$  and  $u^+(x)$ . Eqs.(2.55-2.58) essentially provide an algebraic system for the coefficients of  $u^-(x)$  and  $u^+(x)$ , whose solutions are the representations of those four polynomial coefficients in terms of  $u_{i-1}, u_i, u_{i+1}, u_{i+2}, \phi$  and  $\psi$ . These polynomial coefficients

are solved by inverting the coefficient matrix of equation

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -x_{l} & -x_{l}^{2} & -x_{r} & x_{r}^{2} \\ -\beta^{-} & -2\beta^{-}x_{l} & \beta^{+} & -2\beta^{+}x_{r} \end{pmatrix} \begin{pmatrix} a_{1}^{-} \\ a_{2}^{-} \\ a_{1}^{+} \\ a_{2}^{+} \end{pmatrix} = \begin{pmatrix} u_{i-1} - u_{i} \\ u_{i+2} - u_{i+1} \\ [u] - u_{i+1} + u_{i} \\ h[\beta u_{x}] \end{pmatrix}.$$
 (2.59)

It can be seen from Eqs. (2.53) and (2.54) that

$$u_x^-(x_i) = \frac{a_1^-}{h}, u_{xx}^-(x_i) = \frac{2a_2^-}{h^2}, u_x^+(x_{i+1}) = \frac{a_1^+}{h}, \text{ and } u_{xx}^+(x_{i+2}) = \frac{2a_2^+}{h^2}.$$

The 1-D elliptic equation (1.3) can therefore be approximated at irregular points by

$$\beta^{-} \frac{2a_{2}^{-}}{h^{2}} = \rho(x_{i}) \quad \text{at} \quad x_{i},$$
 (2.60)

$$\beta^{+} \frac{2a_{2}^{+}}{h^{2}} = \rho(x_{i+1}) \text{ at } x_{i+1},$$
 (2.61)

if  $\beta$  is piecewise constant. Otherwise the approximation equations would be

$$\beta_x(x_i)\frac{a_1^-}{h} + \beta(x_i)\frac{2a_2^-}{h^2} = \rho(x_i) \quad \text{at} \quad x_i,$$
(2.62)

$$\beta_x(x_{i+1})\frac{a_1^+}{h} + \beta(x_{i+1})\frac{2a_2^+}{h^2} = \rho(x_{i+1}) \quad \text{at} \quad x_{i+1}.$$
(2.63)

The constants in the representations of  $a_1^-, a_2^-, a_1^+$  and  $a_2^+$  in the above equations, i.e., the terms involving given jumps  $\phi$  and  $\psi$ , should be moved to the right hand side. This finishes the establishment of a second-order interpolation scheme at both irregular points.

The above procedure can be easily generalized to construct higher order schemes. For example, to attain a fourth-order scheme at irregular points near the interface, one can start the formulation of the scheme by defining a fourth-order polynomial on each side of the interface

$$u^{-}(x) = a_{0}^{-} + \frac{a_{1}^{-}}{h}(x - x_{i}) + \frac{a_{2}^{-}}{h^{2}}(x - x_{i})^{2} + \frac{a_{3}^{-}}{h^{3}}(x - x_{i})^{3} + \frac{a_{4}^{-}}{h^{4}}(x - x_{i})^{4}, \quad (2.64)$$

$$u^{+}(x) = a_{0}^{+} + \frac{a_{1}^{+}}{h}(x - x_{i+1}) + \frac{a_{2}^{+}}{h^{2}}(x - x_{i+1})^{2} + \frac{a_{3}^{+}}{h^{3}}(x - x_{i+1})^{3} + \frac{a_{4}^{+}}{h^{4}}(x - x_{i+1})^{4}.$$
(2.65)

The first two coefficients  $a_0^-$  and  $a_0^+$  again have to be  $u_i$  and  $u_{i+1}$  respectively. The remaining eight coefficients are to be determined by using two interface jump conditions, and by expanding these two polynomials at six grid points, i.e., points  $x_{i-1}$ ,  $x_{i-2}$  and  $x_{i-3}$  for Eq. (2.64), and points  $x_{i+2}$ ,  $x_{i+3}$  and  $x_{i+4}$  for Eq. (2.65). We could therefore end up with eight linear algebraic equations

$$\begin{pmatrix} -3 & 9 & -27 & 81 & 0 & 0 & 0 & 0 \\ -2 & 4 & -8 & 16 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 4 & 8 & 16 \\ 0 & 0 & 0 & 0 & 3 & 9 & 27 & 81 \\ -x_l & -x_l^2 & -x_l^3 & -x_l^4 & -x_r & x_r^2 & -x_r^3 & x_r^4 \\ -\beta^- & -2\beta^- x_l & -3\beta^- x_l^2 & -4\beta^- x_l^3 & \beta^+ & -2\beta^+ x_r & 3\beta^+ x_r^2 & -4\beta^+ x_r^3 \end{pmatrix}$$
$$(a_1^-, a_2^-, a_3^-, a_4^-, a_1^+, a_2^+, a_3^+, a_4^+)^T$$
$$= (u_{i-3} - u_i, u_{i-2} - u_i, u_{i-1} - u_i, u_{i+2} - u_{i+1}, u_{i+3} - u_{i+1}, u_{i+4} - u_{i+1}, \\ [u] - u_{i+1} + u_i, h[\beta u_x])^T$$

where the first three equations are generated from  $u^{-}(x)$ , while the next three equations are generated from  $u^{+}(x)$ . The last two equations are the approximation of two interface jump conditions with polynomials  $u^{-}(x)$  and  $u^{+}(x)$  at x = a. By inverting the coefficient matrix, the representations of all the eight polynomial coefficients are solved, and expressions for  $u^{-}(x)$  and  $u^{+}(x)$  are fully determined. Unlike the second-order case, where each polynomial is used only once in formulating the difference scheme at either  $x_i$  or  $x_{i+1}$ , for the fourth-order case, each polynomial is applied twice since there are two irregular points on each side of the interface, i.e.,  $x_{i-1}$  and  $x_i$  on the left, and  $x_{i+1}$  and  $x_{i+2}$  on the right. These points are identified since the regular central difference scheme at these points involves the grid point(s) on the opposite side. In particular,  $u^-(x)$  would be used to formulate the difference scheme at the points  $x_{i-1}$  and  $x_i$ , while  $u^+(x)$  is used at  $x_{i+1}$  and  $x_{i+2}$ . Assuming a piecewise continuous  $\beta$ , the difference schemes at the two closest irregular points are

$$\beta_x(x_i)\frac{a_1^-}{h} + \beta(x_i)\frac{2a_2^-}{h^2} = \rho(x_i) \quad \text{at} \quad x_i,$$
 (2.66)

$$\beta_x(x_{i+1})\frac{a_1^+}{h} + \beta(x_{i+1})\frac{2a_2^+}{h^2} = \rho(x_{i+1}) \quad \text{at} \quad x_{i+1}.$$
(2.67)

At the other two irregular points, the difference schemes are more complicated since the derivatives of the polynomials involve more terms

$$\frac{\beta_x(x_{i-1})}{h}(a_1^- - 2a_2^- + 3a_3^- - 4a_4^-) + \frac{\beta(x_{i-1})}{h^2}(2a_2^- - 6a_3^- + 12a_4^-) = \rho(x_{i-1}) \text{ at } x_{i-1}, \qquad (2.68)$$
$$\frac{\beta_x(x_{i+2})}{h}(a_1^+ + 2a_2^+ + 3a_3^+ + 4a_4^+) + \frac{\beta(x_{i+2})}{h^2}(2a_2^+ + 6a_3^+ + 12a_4^+) = \rho(x_{i+2}) \text{ at } x_{i+2}. \qquad (2.69)$$

It is concluded from the above discussion that at irregular points near the interface, a difference scheme of arbitrary order can be formulated by following these steps:

1. Define a polynomial on each side of the interface

$$u^{-}(x) = u_{i} + \sum_{k=1}^{n} \frac{a_{k}^{-}}{h^{k}} (x - x_{i}), \qquad (2.70)$$

$$u^{+}(x) = u_{i+1} + \sum_{k=1}^{n} \frac{a_{k}^{+}}{h^{k}} (x - x_{i+1}), \qquad (2.71)$$

where n, the order of the polynomial, depends on the global accuracy required.

These polynomials are essentially the approximations of the solution in the vicinity of the interface.

- 2. Expand each polynomial at the nearest n-1 grid points to the interface on the same side. The total number of expansions is therefore 2n-2.
- 3. Approximate two interface jump conditions with these polynomials. Differentiation is needed in approximating  $[\beta u_x]$ , the jump in flux.
- 4. Compose a  $2n \times 2n$  linear algebraic system by combining 2n 2 expansion polynomials and two discretized interface jump conditions. Solve this linear system by inversion, and the solution is the representation of polynomials' coefficients in terms of the approximation solution at involved grid points and the given jumps.
- 5. Approximate the original elliptic equation at all the irregular points by differentiating the polynomials. Replace the polynomial coefficients with their representations. This gives rise to a finite difference scheme at the corresponding grid point.

#### 2.4.2 Two dimensional formalism

The new formulation of the MIB method for 2-D problems with curved interfaces can be accomplished similarly. The essential idea is to construct 1-D interpolation polynomials, instead of 2-D ones. Referring to Figure 2.10, we have two irregular points,  $x_i$  and  $x_{i+1}$ , along the x-axis for a second-order scheme. Two 1-D polynomials, in exactly the same form as Eqs. (2.53) and (2.54), are defined

$$u^{-}(x, y_{j}) = a_{0}^{-} + \frac{a_{1}^{-}}{h}(x - x_{i}) + \frac{a_{2}^{-}}{h^{2}}(x - x_{i})^{2}, \qquad (2.72)$$

$$u^{+}(x, y_{j}) = a_{0}^{+} + \frac{a_{1}^{+}}{h}(x - x_{i+1}) + \frac{a_{2}^{+}}{h^{2}}(x - x_{i+1})^{2}.$$
 (2.73)

These polynomials can be regarded as the 1-D approximation of the solution in the vicinity of the interface along  $y = y_j$  mesh line. Here, one has  $a_0^- = u_{i,j}$  and



**Figure 2.10:** (Same as Figure 2.2) Irregular point (i, j) and the interface. The interface crosses the x- mesh line at  $(x_0, y_0)$ . The vertical dashed line is the auxiliary line on which three auxiliary points (in empty circle) are defined: (o, j + 2), (o, j + 1) and (o, j) right at  $(x_0, y_0)$ . The jumps  $[u], [\beta u_n]$  and  $[u_\tau]$  are evaluated at  $(x_0, y_0)$ .

 $a_0^+ = u_{i+1,j}$  by definition. To solve for the rest four coefficients, these two polynomials are to be evaluated at  $(x_{i-1}, y_j)$  and  $(x_{i+2}, y_j)$ , which gives two equations. The other two equations are obtained from the approximations of interface jump conditions (2.74) and (2.75) [96]

$$[u] = u^+ - u^-, (2.74)$$

$$[\beta u_n] - \beta^- \tan \theta[u_\tau] = C_x^+ u_x^+ + C_x^- u_x^- + C_y^+ u_y^+, \qquad (2.75)$$

where  $\theta$  is the angle between the normal vector and the x-axis,  $C_x^+ = \beta^+ \cos \theta + \beta^- \tan \theta \sin \theta$ ,  $C_x^- = -\beta^- (\cos \theta + \tan \theta \sin \theta)$ , and  $C_y^+ = (\beta^+ - \beta^-) \sin \theta$ . Although the approximation to Eq. (2.74) is trivial, special care has to be taken in treating Eq. (2.75), which couples two directions. Here we approximate Eq. (2.75) as

$$\frac{C_x^+}{h}(a_1^+ - 2a_2^+x_r) + \frac{C_x^-}{h}(a_1^- + 2a_2^-x_l) + (2.76)$$

$$C_y^+(p_{1,j}^+u_{o,j}^+ + p_{1,j+1}^+u_{o,j+1} + p_{1,j+2}^+u_{o,j+2}) = [\beta u_n] - \beta^- \tan\theta[u_\tau].$$

where p is the FD weight in the y-direction, and  $x_l = \frac{x_o - x_i}{h}, x_r = \frac{x_{i+1} - x_o}{h}$ . Note that here auxiliary values  $u_{o,j}^+, u_{o,j+1}$  and  $u_{o,j+2}$  are also introduced to calculate  $u_y^+$  as done in the fictitious domain formulation [96], where

$$u_{o,j}^{+} = u_{x_{o},y_{o}}^{-} + [u] = (u_{i,j} + a_{1}^{-}x_{l} + a_{2}^{-}x_{l}^{2}) + [u].$$
(2.77)

The final algebraic system for four polynomial coefficients is

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -x_l & -x_l^2 & -x_r & x_r^2 \\ \frac{1}{h}C_x^- + C_y^+ p_{1,j}^+ x_l & \frac{2}{h}C_x^- x_l + C_y^+ p_{1,j}^+ x_l^2 & \frac{1}{h}C_x^+ & -\frac{2}{h}C_x^+ x_r \end{pmatrix} \begin{pmatrix} a_1^- \\ a_2^- \\ a_1^+ \\ a_2^+ \end{pmatrix} = \begin{pmatrix} u_{i-1,j} - u_{i,j} \\ u_{i+2,j} - u_{i+1,j} \\ \phi - u_{i+1,j} + u_{i,j} \\ [\beta u_n] - \beta^- \tan \theta[u_\tau] - C_y^+ \{ p_{1,j}^+ (u_{i,j} + [u]) + p_{1,j+1}^+ u_{o,j+1} + p_{1,j+2}^+ u_{o,j+2} \} \end{pmatrix}.$$

Desirable difference schemes for  $u_x$  or  $u_{xx}$  at two irregular points can be obtained from the direct differentiation of these two polynomials after all of their coefficients are determined. Following a similar procedure, one can determine polynomials  $u^-(x_i, y)$ and  $u^+(x_i, y)$ , which are then used to approximate  $u_y^-$ ,  $u_{yy}^-$ ,  $u_y^+$  and  $u_{yy}^+$  at corresponding irregular points.

High order 2-D interpolation schemes can be established in a manner similar to what was described for 1-D high order schemes. However, for general interfaces with large curvatures, the construction of a high order interpolation MIB scheme is subject to the same difficulty as that in the fictitious domain formulation, i.e., there are not sufficient grid points to support high order polynomials along all the xand y-directions on one side of an interface, see  $x = x_i$  mesh line in Figure 2.8. It normally takes four grid points to determine a fifth-order polynomial  $u^-(x_i, y)$ , which is required for calculating  $u_y$  and  $u_{yy}$  at four-irregular points,  $x_{i,j-2}$ ,  $x_{i,j-1}$ ,  $x_{i,j+2}$  and  $x_{i,j+3}$ , in a fourth-order scheme. One solution to this problem is to make use of two interface intersection points, which would provide four jump conditions. However, a simple remedy is to make use of two other polynomials found in the x-direction, namely,  $u^-(x, y_{j+2})$  and  $u^-(x, y_{j-1})$  to supply function values at  $x_{i,j-1}$  and  $x_{i,j+2}$ . These two computed function values, together with the original function values  $u_{i,j}$ and  $u_{i,j+1}$ , as well as two interface jump conditions, can determine  $u^-(x_i, y)$ . As usual, this determination is sought in coupling with the determination of  $u^+(x_i, y)$ , whose solution has no additional problem. Moreover, we can choose either one of the two interface intersection points along the  $x = x_i$  mesh line.

The two formulations also differ from each other in dealing with large curvatures. The fictitious domain approach avoids determining a fictitious value along a discretization direction whenever there are insufficient number of grid values along the direction inside an interface. It makes use of grid values in the other direction to determine the fictitious values. However, the interpolation formulation cannot avoid determining polynomials in all directions at an irregular point because it has to calculate both the x- and y-derivatives at the irregular point. It therefore makes use of other polynomials to provide function values outside the interface to complete the determination of each required polynomial. These two approaches might involve different sets of grid values in dealing with a given situation.

#### 2.4.3 Comparison of two formulations

In this subsection, we discuss the similarities and difference of the fictitious domain formulation and the interpolation formulation mainly based on 1-D cases. In terms of similarities, both formulations share the same set of irregular points for a given interface geometry and given order of the scheme. Both approaches utilize only the standard (high order) central finite difference discretization and the lowest order interface jump conditions. The essential equivalence between the two formulations would be obvious if the fictitious value formulation was also casted in polynomial expressions. Referring to Figure 1.1, bridged by the fictitious value  $f_{i+1}$ , the solution in the left vicinity of the interface is actually approximated by a Lagrange interpolation polynomial

$$u^{-}(x) = \sum_{k=0}^{1} L_{k} u_{i-1+k} + L_{2} f_{i+1}$$
(2.78)

to an accuracy of  $O(h^3)$  or

$$u^{-}(x) = \sum_{k=0}^{3} L_{k} u_{i-3+k} + L_{4} f_{i+1}$$
 (2.79)

to an accuracy of  $O(h^5)$ . These two polynomials, although not explicitly constructed, are actually approximated in seeking the fictitious values. Moreover, a very important common feature is that, in either 1D or higher dimensions, both approaches make use of only 1-D polynomials. This treatment significantly simplifies the scheme.

Comparing Eq. (2.78) with Eq. (2.53), or Eq. (2.78) with Eq. (2.64) it can be found that with the fictitious domain approach one only needs to determine two parameters, i.e, the fictitious values  $f_i$  and  $f_{i+1}$ , to resolve the approximate solutions in both the left and the right vicinities of the interface. With the new formulation, however, one has to solve for all the six polynomial coefficients to determine the approximate solution.

Since approximate polynomials are explicitly determined in the new formulation, they are then directly differentiated to provide the approximation to the first and second derivatives in the elliptic equation. In the fictitious domain approach, instead of differentiating approximate polynomials implicitly determined with the first pair of fictitious values  $f_i$  and  $f_{i+1}$ , a formal central difference scheme involving  $f_i$  or  $f_{i+1}$ is adopted at each irregular point to approximate the partial derivative. Moreover, to support a fourth or higher order scheme, more fictitious values are needed, and they have to be solved progressively after determining  $f_i$  and  $f_{i+1}$ . With the new formulation, however, all the coefficients in the high order polynomials are solved simultaneously.

## 2.5 Numerical experiments on irregular interfaces

The performance of the MIB scheme for 2-D elliptic problems with irregular interfaces is to be examined in this section with a couple of examples. We will primarily focus on the second-order and fourth-order MIB method in our numerical experiments, while the last case is devoted to the validation of high order MIB schemes. Numerical results are compared to the analytical solutions of the equations, in terms of both numerical accuracy and computational efficiency. The IIM of LeVeque and Li [42] is regenerated for a comparison in some test cases. The performance of our IIM code has been verified with that in the literature [42] and is found to be similar to a later version of the IIM given by Li and Ito [46]. The PBCG solver [65] is adopted to solve the linear system due to its efficiency and the simplicity in implementation. The standard  $L_{\infty}$  norm error measurement is employed in this section.

Example 2.5.1. We consider a 2-D Poisson equation

$$(\beta u_x)_x + (\beta u_y)_y = f(x, y)$$
(2.80)

defined in a square  $[-1,1] \times [-1,1]$  with a circular interface  $r^2 \equiv x^2 + y^2 = \frac{1}{4}$  inside. Following [42], the exact solution is designed to be

$$u(x,y) = \begin{cases} x^2 + y^2 & r \le 0.5\\ \frac{1}{4}(1 - \frac{1}{8b} - \frac{1}{b}) + (\frac{r^4}{2} + r^2)/b & otherwise \end{cases}$$

with the diffusion coefficient

$$\beta(x,y) = \begin{cases} 2 & r \le 0.5 \\ b & otherwise \end{cases}$$

$n_x  imes n_y$	Second-ord	ler MIB	IIM		
	$L_{\infty}$	Order	$L_{\infty}$	Order	
$20 \times 20$	2.852 (-4)		2.167 (-3)		
$40 \times 40$	7.707 (-5)	1.9	5.000 (-4)	2.1	
$80 \times 80$	2.069 (-5)	1.9	1.131 (-4)	2.1	
$160 \times 160$	5.131 (-6)	2.0	2.748 (-5)	2.0	
320 × 320	1.257 (-6)	2.0	6.781 (-6)	2.0	

**Table 2.1:** Numerical efficiency test of the 2-D Poisson equation in Example 2.5.1.  $b = 10, [u] = 0, [\beta u_n] = -0.75.$ 

Such a designated solution forces the discontinuous inhomogeneous term f(x, y) to be

$$f(x,y) = \begin{cases} 8.0 & r \le 0.5\\ 8(x^2 + y^2) + 4.0 & otherwise \end{cases}$$

Let b = 10 such that u(x, y) is continuous throughout the domain and  $[\beta u_n] = -0.75$  on the interface. The computed result with a 20 × 20 mesh is plotted in Figure 2.11. Table 2.1 lists the computed error of the second-order MIB scheme in a comparison with the results of the IIM. Both methods have very clear second-order accuracy. The MIB delivers a more accurate result than that of the IIM. It is found that the CPU times used for generating the local immersed grids are almost the same for both methods although the underlying algorithms are different. The CPU times used for solving the linear algebraic equation systems are not compared because the IIM method has its own optimal solver [46], while no such solver is available for the MIB yet. It is expected that the IIM method is faster since it involves fewer irregular nodes.

**Example 2.5.2.** We want to use our forth-order MIB method to solve the same Poisson equation in example (2.5.1) with a designated finite jump in [u] at the interface:

$$u(x,y) = \begin{cases} x^2 + y^2 - 1 & r \le 0.5 \\ \frac{1}{4}(1 - \frac{1}{8b} - \frac{1}{b}) + (\frac{r^4}{2} + r^2)/b & otherwise \end{cases}$$

It can be checked that on the interface [u] = 1 and  $[\beta u_n] = -0.75$  with b = 10.



Figure 2.11: Computed solution (upper) and the error (lower) for the 2-D Poisson equation in Example 2.5.1.  $\beta = 10, [u] = 0, [\beta u_n] = -0.75$ .

Because the analytical solution is a fourth-order polynomial, it is expected that a fourth-order method shall render a numerical solution of the machine error. The error plot in Figure 2.12 exactly illustrates this prediction, where the maximum error is around  $10^{-14}$ .



Figure 2.12: The computed solution (upper) and the error (lower) for Example 2.5.2.

**Example 2.5.3.** In this example we solve the Laplace equation  $u_{xx} + u_{yy} = 0$ , which is defined in the square  $[-1, 1] \times [-1, 1]$  and has the following analytical solution:

$$u(x,y) = \begin{cases} e^x \cos(y) & r \le 0.5\\ 0 & otherwise \end{cases}$$

The jumps in u and  $u_n$  along the interface  $r = \frac{1}{2}$  can be evaluated from this solution. Note that here we have unit diffusion coefficient throughout the whole domain. Table 2.2 gives the computational results of the second-order MIB and the IIM for a comparison. Again, as expected, a steady second-order convergence is validated for both methods. Moreover, the MIB scheme is favored over IIM due to the smaller numerical error on all the 5 successively refined meshes. The computed result in Figure 2.13 sharply features designated discontinuity along the circular interface. **Table 2.2:** Numerical efficiency test of the 2-D Laplace equation in Example 2.5.3.  $\beta = 1$ .

$nx \times ny$	Second-ord	ler MIB	IIM	IIM		
	$L_{\infty}$	Order	$L_{\infty}$	Order		
$20 \times 20$	1.015 (-4)		4.389 (-4)			
$40 \times 40$	2.511 (-5)	2.0	1.079 (-4)	2.0		
$80 \times 80$	6.369 (-6)	2.0	2.778 (-5)	2.0		
$160 \times 160$	1.608 (-6)	2.0	7.500 (-6)	1.9		
$320 \times 320$	3.714 (-7)	2.1	1.740 (-6)	2.1		

**Example 2.5.4.** We consider the Poisson equation with a computational domain  $[-1, 1] \times [-1, 1]$  and an elliptical interface

$$\left(\frac{x}{18/27}\right)^2 + \left(\frac{y}{10/27}\right)^2 = 1.$$



**Figure 2.13:** Computed solution (upper) and the error (lower) for the 2-D Laplace equation in Example 2.5.3.  $\beta \equiv 1$ .

The analytical solution and the coefficient  $\beta$  are given as the follows

$$u(x,y) = \begin{cases} e^x \cos(y) & \text{inside } \Gamma \\ 5 \exp\left(-x^2 - \frac{y^2}{2}\right) & \text{otherwise} \end{cases}$$
$$\beta(x,y) = \begin{cases} b & \text{inside } \Gamma \\ 1 & \text{otherwise.} \end{cases}$$

Two cases are considered, one with b = 10 and the other with b = 1000. The latter shows a strong discontinuity in the coefficient  $\beta$  and demands more iterations in solving the linear system as it is ill-conditioned due to the large jump in  $\beta$ . The lower accuracy for the case with b = 1000 can be attributed to the larger jump in the coefficient.

	b = 10			b = 1000				
$n_x = n$	4 <sup>th</sup> MIB		2 <sup>nd</sup> MIB		4 <sup>th</sup> MIB		2 <sup>nd</sup> MIB	
$n_y$	$L_{\infty}$	Order	$L_{\infty}$	Order	$L_{\infty}$	Order	$L_{\infty}$	Order
40	3.92E-5		5.21E-3		6.19E-3		2.76E-2	
80	2.92E-6	3.75	1.49E-3	1.8	2.65E-4	4.55	7.52E-3	1.9
160	1.70E-7	4.10	3.75E-4	2.0	1.33E-5	4.32	2.17E-3	2.0
320	8.57E-9	4.31	7.80E-5	2.3	6.73E-7	4.30	4.84E-4	2.2

**Table 2.3:** Numerical convergence test and accuracy test for Example 2.5.4.

To validate the asymptotic behavior of the approximation error with the increasing of the jump in  $\beta$ , a series of  $\beta^+$  are chosen for numerical tests. The results are collected in Table 2.4. It can be seen that for moderate magnitude the numerical error is increasing while for large jumps the error is almost a constant.

Example 2.5.5. To examine the resolution of the proposed method, a Helmholtz-like

**Table 2.4:** Robustness test of high order MIB scheme with Example 2.5.4.  $\beta^- = 1$ .  $n_x = n_y = 40$ .

$\beta^+$	10	20	100	500	$10^{3}$	$10^{4}$	$10^{5}$	$10^{8}$
$L_{\infty}$	3.92E-5	7.10E-5	1.89E-4	3.04E-4	3.28E-4	3.53E-4	3.56E-4	3.56E-4



Figure 2.14: The computed solution (upper) and the error (lower) for Example 2.5.4 with second-order method and  $n_x = n_y = 40$ .

equation

$$\nabla \cdot (\beta \nabla u(x, y)) + k^2(x, y)u(x, y) = q(x, y)$$

is considered, where  $k(x,y) = \kappa \sigma(x,y)$  is the dielectric function describing macro-

scopically the properties of the medium in which the wave propagates. Both  $\beta$  and k are discontinuous at the interface. The analytical solution to the equation is designed to be highly oscillatory

$$u(x,y) = \begin{cases} x^2 + y^2 & r \le 0.5\\ \sin(\kappa x)\cos(\kappa y) & otherwise, \end{cases}$$

where  $\kappa$  can be tuned to produce solutions of desired frequency. The computational domain and the interface are the same as those in Example(2.5.2), while

$$eta = \begin{cases} 10 & r \leq 0.5 \\ 1 & otherwise, \end{cases}$$

and

$$\sigma(x,y) = \begin{cases} 1 & r \leq 0.5 \\ \sqrt{10} & otherwise. \end{cases}$$

It is generally believed that a high order method usually comes with high resolution, which is absolutely needed in solving the Maxwell's equation or the Helmholtz equation for high frequency wave propagation and scattering. In this example two different frequencies,  $\kappa = 2$  and  $\kappa = 12$ , are considered as in Table 2.5 where the solution of the present fourth-order scheme is compared with that of the second-order MIB scheme. It is observed that the order of convergence agrees with the theoretical analysis for both schemes. Also observed is that the low frequency solution can be well approximated by both schemes. However, for the high frequency case, considerable differences can be found in the two schemes. In particular, for the low frequency case, a sparse mesh ( $20 \times 20$ ) is sufficient for both schemes to produce a result of moderate accuracy, i.e., around  $10^{-3}$ . For the high frequency problem, since the solution admits large gradients and is anisotropic, one has to use a very dense mesh (320 × 320) to resolve the fast variation in the solution if one sticks to a second-order method. It is noted that by using the fourth-order MIB scheme, nevertheless, an 80 × 80 mesh can provide a sufficient resolution. It is therefore anticipated that with the proposed high-order interface method, significant saving on computing time can be achieved for problems that involve both material interface and high frequency oscillations.

**Example 2.5.6.** This example was introduced in Ref. [43], and is adopted here to examine the flexibility of MIB in dealing with complex interfaces. The analytical solution to the equation, the coefficient  $\beta$  and a jigsaw puzzle-like interface  $\Gamma$  are given below

$$u(x,y) = \begin{cases} e^{x}(y^{2} + x^{2}\sin(y)) & inside \ \Gamma \\ -(x^{2} + y^{2}) & otherwise, \end{cases}$$
  
$$\beta(x,y) = \begin{cases} 1 & inside \ \Gamma \\ 10 & otherwise, \end{cases}$$
  
$$\Gamma : \qquad \begin{cases} x(\theta) = & 0.6\cos(\theta) - 0.3\cos(3\theta) \\ y(\theta) = & 1.5 + 0.7\sin(\theta) - 0.07\sin(3\theta) + 0.2\sin(7\theta). \end{cases}$$

A discretization of the interface is plotted in Figure 2.9. The computed solution and the error for a 100  $\times$  100 mesh are plotted in Figure 2.16 and the numerical error in terms of the  $L_2$  norm are collected in Table 2.6, together with the data of the second-order counterpart. We note that the predicted convergence rate for both methods are verified, whereas the fourth-order method gives a much more accurate result. The maximum error occurs at the irregular points near the interface where the local truncation error is one-order lower than that at regular points.

**Example 2.5.7.** This is another standard test case for testing numerical methods designed for solving elliptic interface problems. The interface is parametrized with


Figure 2.15: The computed solution (upper) and the error (lower) for Example2.5.5.

the polar angle  $\theta$ , as

$$r = \frac{1}{2} + \frac{\sin(\theta)}{7}.$$

The exact solution to the problem can be arbitrarily designed. Here we choose

$$u(x,y) = \begin{cases} \exp(x^2 + y^2) & \text{inside } \Gamma \\ 0.1(x^2 + y^2)^2 - 0.01 \ln(2\sqrt{x^2 + y^2}) & \text{otherwise} \end{cases}$$
$$\beta(x,y) = \begin{cases} 1 & \text{inside } \Gamma \\ 10 & \text{otherwise} \end{cases}$$

This solution also prescribes the Dirichlet boundary condition of the problem. The non-homogeneous term of the Poisson equation can also be derived from the exact solution. Figure 2.16 plots the computed solution with a mesh of  $100 \times 100$ . Table 2.7 shows the results of the numerical accuracy tests on three successively refined meshes, in comparison with the second-order MIB method.

In the last example, we will scrutinize the convergence of the fourth-order and sixth-order MIB methods. We note that the present method is particularly favorable if the immersed inner boundary is convex. In such cases, there is always an adequate number of grid points to support a given high order approximation of the interface conditions.

**Example 2.5.8.** Here we choose the immersed inner boundary as a circle of radius  $\frac{1}{2}$  with its center located at the origin. The computational domain is  $[-1, 1] \times [-1, 1]$ , excluding the inner circle. The exact solution of the Poisson equation is set to be  $u(x, y) = 5 \exp(-x^2 - y^2/2)$  on the computational domain.

We plot the numerical solution, computed on a  $40 \times 40$  grid, in Figure 2.17 with a negative sign (i.e., -u(x, y)) to illustrate the immersed inner boundary. The accuracy and convergence order of our MIB method are presented in Table 2.8. For high order methods, it takes more time in the generation of the local difference schemes because of the iterative nature of the current algorithm, as given in Table 2.8. However, this time is still negligible compared to the time spent on the solution of the main linear system. It is seen that the result obtained by using the sixth order scheme at the grid of  $40 \times 40$  is about 30 times more accurate than that obtained by using the second-order scheme at the grid of  $640 \times 640$ , while consuming only a fraction (1/473) of CPU time. In order for the second method to reach the same accuracy of the sixth order scheme at the grid of  $40 \times 40$ , a grid larger than  $2600 \times 2600$  has to been used, which approximately costs 300000 seconds CPU time. Therefore, the sixth order scheme is about 28000 times more efficient than the second-order scheme in terms of CPU time. It is noted that the CPU time used for the generation of the local finite difference scheme (i.e, the immersed grids) is very small compared to that for solving the linear algebraic equation system.

	NON NTO ACON	in in the		No Noter				
		к "	= 2			א א	12	
nx = ny	4 <sup>th</sup> N	AIB	2 <sup>nd</sup> N	AIB	$4^{\rm th}$ N	<b>AIB</b>	2 <sup>nd</sup> N	<b>AIB</b>
	$L_{\infty}$	Order	$L_{\infty}$	Order	$L_{\infty}$	Order	$L_{\infty}$	Order
20	3.91E-5		3.51E-3		2.13E-1		7.80E-1	
40	1.04E-5	1.91	1.41E-3	1.32	2.86E-2	1.90	1.50E-1	2.34
80	6.34E-7	4.04	3.70E-4	1.93	1.92E-3	4.55	3.31E-2	2.18
160	4.37E-8	3.86	1.01E-4	1.87	2.04E-4	4.32	1.06E-2	1.64
320	2.78E-9	3.97	2.36E-5	2.10	8.43E-6	4.60	1.95E-3	2.44

Table 2.5: Numerical convergence test and accuracy test for Example 2.5.5.



Figure 2.16: Upper: the computed solution of Example(2.5.6); Lower: computed solution of Example(2.5.7). Both are computed with mesh  $n_x = n_y = 100$ .

n — n	4 <sup>th</sup> MIB		2 <sup>nd</sup> MIB	
$n_x - n_y$	$L_2$	Order	$L_2$	Order
100	2.10E-8		3.78E-5	
200	1.54E-9	3.77	1.06E-5	1.83
400	5.11E-11	<b>3</b> .95	2.49E-6	2.09

2. . 1

۰.

 Table 2.6: Numerical convergence test for Example 4.

 Table 2.7: Numerical convergence test for Example 5.

	4 <sup>th</sup> M	IIB	2 <sup>nd</sup> N	ЛIВ
$n_x = n_y$	$L_2$	Order	$L_2$	Order
100	1.29E-7		5.03E-5	
200	1.17E-8	3.46	1.35E-5	1.90
400	7.09E-10	4.04	3.41E-6	1.99



**Figure 2.17:** The solution of the high order MIB method for the 2-D Poisson equation in Example 2.5.8. -u(x, y) is plotted.

1	Set	cond-ord	er MIB	Four	rth-orde	r MIB	Six	th-orde	r MIB
$n_{x} \times n_{y}$	$L_{\infty}$	Order	CPU	$L_{\infty}$	Order	CPU	$L_{\infty}$	Order	CPU
$20 \times 20$	1.20 (-3)		0.155(0.008)	2.47 (-5)		0.552(0.02)			
$40 \times 40$	2.85 (-4)	2.07	1.450(0.026)	1.51 (-6)	4.03	3.714(0.07)	3.48 (-8)		10.94(0.15)
$80 \times 80$	6.55 (-5)	2.12	11.18(0.089)	6.16 (-8)	4.61	25.85(0.24)	3.42 (-10)	6.64	90.67(0.55)
$160 \times 160$	1.67 (-5)	1.97	90.26(0.227)	2.88 (-9)	4.41	249.8(0.89)	4.62 (-12)	6.21	480.9(1.90)
$320 \times 320$	4.10 (-6)	2.03	595.0(0.503)	1.33(-10)	4.44	2760(1.65)			
$640 \times 640$	1 02 (-6)	2 01	5174(2.15)						

on equation (Case 6). The numbers in the	
sixth-order MIB schemes for the Poissc	neration of local difference schemes.
of the second-, fourth- and	times in seconds for the gen
Table 2.8: Comparison	parentheses are the CPU

### Chapter 3

# MIB Method: 3-D Formulation and Numerical Experiments

# 3.1 Second-Order 3-D MIB Scheme for Irregular Interfaces

Consider a 3-D elliptic interface problem

$$\nabla \cdot (\beta \nabla u(r)) - \kappa(r)u(r) = \rho(r) \tag{3.1}$$

with interface conditions

$$[u] = u^+ - u^-, (3.2)$$

$$[\beta u_{\xi}] = \beta^{+} u_{\xi}^{+} - \beta^{-} u_{\xi}^{-}.$$
(3.3)

Here the coefficient function  $\kappa(r)$  and the source function  $\rho(r)$  are given. The linear Poisson-Boltzmann equation to be derived in the next chapter is a special case of Eq. (3.1). As the condition (3.3) is defined in the normal direction  $\xi$ , a local coordinate transformation is needed to relate this normal gradient  $u_{\xi}$  to the partial derivatives with respect to the three Cartesian coordinates  $u_x, u_y, u_z$ ,

$$(u_{\xi}, u_{\eta}, u_{\zeta})^{\mathrm{T}} = \mathbf{A} \cdot (u_x, u_y, u_z)^{\mathrm{T}}$$
(3.4)

where the superscript T denotes the transpose of the transformation matrix  $\mathbf{A}$  which is defined as

$$\mathbf{A} = \begin{pmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta & \sin\psi \\ -\sin\theta & \cos\theta & 0 \\ -\sin\psi\cos\theta & -\sin\psi\sin\theta & \cos\psi \end{pmatrix}.$$
 (3.5)

The coordinate transformation represented by A is accomplished through two steps. In the first step, the x - y plane is rotated with respect to the z- axis by an angle  $\theta$  such that the normal vector would be located on the new x - z plane. Denoting this new coordinate by (x', y', z'), in the second step, the x' - z' plane is rotated with respect to the y'-axis by an angle  $\psi$  such that the normal vector is aligned with the x'- axis after rotation. The coordinate after the second rotation is essentially  $(\xi, \eta, \zeta)$ , since

$$\begin{pmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta & \sin\psi\\ -\sin\theta & \cos\theta & 0\\ -\sin\psi\cos\theta & -\sin\psi\sin\theta & \cos\psi \end{pmatrix} = \begin{pmatrix} \cos\psi & 0 & \sin\psi\\ 0 & 1 & 0\\ -\sin\psi & 0 & \cos\psi \end{pmatrix} \times \begin{pmatrix} \cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1 \end{pmatrix}, \quad (3.6)$$

where in the right-hand side the first matrix is the transformation matrix for the second rotation with respect to the y'- axis and the second matrix represents the first rotation with respect to the z-axis. For points located on the contact surface, it is pretty easy to obtain these two angles  $\theta, \psi$  as in this case  $\xi$  is actually the outer normal direction of a sphere with the given center. Details for computing these angles

will be presented in the next section.

With local transformation (3.4), interface condition (3.3) can be recasted as

$$[\beta u_{\xi}] = \beta^{+} (\cos\psi\cos\theta u_{x}^{+} + \cos\psi\sin\theta u_{y}^{+} + \sin\psi u_{z}^{+}) - \beta^{-} (\cos\psi\cos\theta u_{x}^{-} + \cos\psi\sin\theta u_{y}^{-} + \sin\psi u_{z}^{-}).$$
(3.7)

Nevertheless, one can not simply use (3.7) and (3.2) as the two conditions to solve for the fictitious values  $f_{i,j,k}$ ,  $f_{i+1,j,k}$  in the *x*-direction because Eq. (3.7) involves four partial derivatives in the *y*- and *z*- directions which can not be approximated using only the grid points in the *x*- mesh line. To reduce the number of partial derivatives in the interface conditions, we can differentiate interface condition (3.2) with respect to the tangent and binormal coordinates respectively to provide two additional interface conditions

$$u_{\eta}^{+} - u_{\eta}^{-} = (-\sin\theta u_{x}^{+} + \cos\theta u_{y}^{+}) - (-\sin\theta u_{x}^{-} + \cos\theta u_{y}^{-}), \qquad (3.8)$$

$$u_{\zeta}^{+} - u_{\zeta}^{-} = (-\sin\psi\cos\theta u_{x}^{+} - \sin\psi\sin\theta u_{y}^{+} + \cos\psi u_{z}^{+})$$
$$- (-\sin\psi\cos\theta u_{x}^{-} - \sin\psi\sin\theta u_{y}^{-} + \cos\psi u_{z}^{-}), \qquad (3.9)$$

and then cancel any two derivatives which are not in the primary direction from Eqs. (3.7),(3.8) and (3.9). It turns out that there are 12 possible combinations for such cancellation, 4 for x- being the primary direction, 4 for y- and 4 for z-, which are listed as follows

1. Cancel  $u_y^-$  and  $u_z^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] - \beta^{-} \tan \psi[u_{\zeta}] + C_{\eta}[u_{\eta}] = C_{x}^{+} u_{x}^{+} + C_{x}^{-} u_{x}^{-} + C_{y}^{+} u_{y}^{+} + C_{z}^{+} u_{z}^{+}$$
(3.10)

$$C_{\eta} = -\beta^{-} \tan \theta / \cos \psi$$

$$C_{x}^{+} = \beta^{+} \cos \psi \cos \theta + \beta^{-} \tan \psi \sin \psi / \cos \theta + \beta^{-} \cos \psi \sin \theta \tan \theta$$

$$C_{x}^{-} = -\beta^{-} / (\cos \psi \cos \theta)$$

$$C_{y}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

$$C_{z}^{+} = (\beta^{+} - \beta^{-}) \sin \psi$$

2. Cancel  $u_y^+$  and  $u_z^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{+} \tan[u_{\zeta}] = C_{x}^{+} u_{x}^{+} + C_{x}^{-} u_{x}^{-} + C_{y}^{+} u_{y}^{+} + C_{z}^{-} u_{z}^{-}$$
(3.11)

where

$$C_{\eta} = -\beta^{-} \cos \psi \tan \theta - \beta^{+} \tan \psi \sin \psi \tan \theta$$

$$C_{x}^{+} = \beta^{+} \cos \psi \cos \theta + \beta^{-} \cos \psi \sin \theta \tan \theta + \beta^{+} \tan \psi \sin \psi / \cos \theta$$

$$C_{x}^{-} = -(\beta^{+} \tan \psi \sin \psi + \beta^{-} \cos \psi) / \cos \theta$$

$$C_{y}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

$$C_{z}^{-} = (\beta^{+} - \beta^{-}) \sin \psi$$

3. Cancel  $u_y^-$  and  $u_z^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{-} \tan \psi[u_{\zeta}] = C_{x}^{+} u_{x}^{+} + C_{x}^{-} u_{x}^{-} + C_{y}^{-} u_{y}^{-} + C_{z}^{+} u_{z}^{+}$$
(3.12)

$$C_{\eta} = -\beta^{+} \cos \psi \tan \theta - \beta^{-} \tan \psi \sin \psi \tan \theta$$

$$C_{x}^{+} = (\beta^{+} \cos \psi + \beta^{-} \tan \psi \sin \psi) / \cos \theta$$

$$C_{x}^{-} = -\beta^{-} \cos \psi \cos \theta - \beta^{+} \cos \psi \sin \theta \tan \theta - \beta^{-} \tan \psi \sin \psi / \cos \theta$$

$$C_{y}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

$$C_{z}^{+} = (\beta^{+} - \beta^{-}) \sin \psi$$

4. Cancel  $u_y^+$  and  $u_z^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{+} \tan \psi[u_{\zeta}] = C_{x}^{+} u_{x}^{+} + C_{x}^{-} u_{x}^{-} + C_{y}^{-} u_{y}^{-} + C_{z}^{-} u_{z}^{-}$$
(3.13)

where

$$C_{\eta} = -\beta^{+} \tan \theta / \cos \psi$$

$$C_{x}^{+} = \beta^{+} / (\cos \psi \cos \theta)$$

$$C_{x}^{-} = -\beta^{-} \cos \psi \cos \theta - \beta^{+} \cos \psi \sin \theta \tan \theta - \beta^{+} \tan \psi \sin \psi / \cos \theta$$

$$C_{y}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

$$C_{z}^{-} = (\beta^{+} - \beta^{-}) \sin \psi$$

5. Cancel  $u_x^-$  and  $u_z^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{-} \tan \psi[u_{\zeta}] = C_{y}^{+}u_{y}^{+} + C_{y}^{-}u_{y}^{-} + C_{x}^{+}u_{x}^{+} + C_{z}^{+}u_{z}^{+}$$
(3.14)

$$C_{\eta} = \beta^{-} \cot \theta / \cos \psi$$

$$C_{y}^{+} = \beta^{+} \cos \psi \sin \theta + \beta^{-} \cos \psi \cos \theta \cot \theta + \beta^{-} \tan \psi \sin \psi / \sin \theta$$

$$C_{y}^{-} = -\beta^{-} / (\cos \psi \sin \theta)$$

$$C_{x}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{z}^{+} = (\beta^{+} - \beta^{-}) \sin \psi$$

6. Cancel  $u_x^-$  and  $u_z^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\zeta}] + C_{\eta}[u_{\eta}] - \beta^{+} \tan \psi[u_{\zeta}] = C_{y}^{+}u_{y}^{+} + C_{y}^{-}u_{y}^{-} + C_{x}^{+}u_{x}^{+} + C_{z}^{-}u_{z}^{-}$$
(3.15)

where

$$C_{\eta} = \beta^{-} \cos \psi \cot \theta + \beta^{+} \tan \psi \sin \psi \cot \theta$$

$$C_{y}^{+} = \beta^{+} \cos \psi \sin \theta + \beta^{-} \cos \psi \cos \theta \cot \theta + \beta^{+} \tan \psi \sin \psi / \sin \theta$$

$$C_{y}^{-} = -(\beta^{-} \cos \psi + \beta^{+} \tan \psi \sin \psi) / \sin \theta$$

$$C_{x}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{z}^{-} = (\beta^{+} - \beta^{-}) \sin \psi$$

7. Cancel  $u_x^+$  and  $u_z^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{-} \tan \psi[u_{\zeta}] = C_{y}^{+}u_{y}^{+} + C_{y}^{-}u_{y}^{-} + C_{x}^{-}u_{x}^{-} + C_{z}^{+}u_{z}^{+}$$
(3.16)

$$C_{\eta} = \beta^{+} \cos \psi \cot \theta + \beta^{-} \tan \psi \sin \psi \cot \theta$$

$$C_{y}^{+} = (\beta^{+} \cos \psi + \beta^{-} \tan \psi \sin \psi) / \sin \theta$$

$$C_{y}^{-} = -\beta^{-} \cos \psi \sin \theta - \beta^{+} \cos \psi \cos \theta \cot \theta - \beta^{-} \tan \psi \sin \psi / \sin \theta$$

$$C_{x}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{z}^{+} = (\beta^{+} - \beta^{-}) \sin \psi$$

8. Cancel  $u_x^+$  and  $u_z^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] - \beta^{+} \tan \psi[u_{\zeta}] = C_{y}^{+} u_{y}^{+} + C_{y}^{-} u_{y}^{-} + C_{x}^{-} u_{x}^{-} + C_{z}^{-} u_{z}^{-}$$
(3.17)

where

$$C_{\eta} = \beta^{+} \cot \theta / \cos \psi$$

$$C_{y}^{+} = \beta^{+} / (\cos \psi \sin \theta)$$

$$C_{y}^{-} = -\beta^{-} \cos \psi \sin \theta - \beta^{+} \cos \psi \cos \theta \cot \theta - \beta^{+} \tan \psi \sin \psi / \sin \theta$$

$$C_{x}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{z}^{-} = (\beta^{+} - \beta^{-}) \sin \psi$$

9. Cancel  $u_x^-$  and  $u_y^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] + C_{\zeta}[u_{\zeta}] = C_{z}^{+}u_{z}^{+} + C_{z}^{-}u_{z}^{-} + C_{x}^{+}u_{x}^{+} + C_{y}^{+}u_{y}^{+}$$
(3.18)

$$C_{\eta} = 0$$

$$C_{\zeta} = \beta^{-} \cot \psi$$

$$C_{z}^{+} = \beta^{+} \sin \psi + \beta^{-} \cos \psi \cot \psi$$

$$C_{z}^{-} = -\beta^{-} / \sin \psi$$

$$C_{x}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{y}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

10. Cancel  $u_x^-$  and  $u_y^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] + C_{\zeta}[u_{\zeta}] = C_{z}^{+}u_{z}^{+} + C_{z}^{-}u_{z}^{-} + C_{x}^{+}u_{x}^{+} + C_{y}^{-}u_{y}^{-}$$
(3.19)

where

$$C_{\eta} = -(\beta^{+} - \beta^{-})\cos\psi\sin\theta\cos\theta$$

$$C_{\zeta} = \beta^{+}\cot\psi\sin^{2}\theta + \beta^{-}\cot\psi\cos^{2}\theta$$

$$C_{z}^{+} = \beta^{+}\sin\psi + \beta^{+}\cos\psi\cot\psi\sin^{2}\theta + \beta^{-}\cos\psi\cot\psi\cos^{2}\theta$$

$$C_{z}^{-} = -\beta^{-}\sin\psi - \beta^{+}\cos\psi\cot\psi\sin^{2}\theta - \beta^{-}\cos\psi\cot\psi\cos^{2}\theta$$

$$C_{x}^{+} = (\beta^{+} - \beta^{-})\cos\psi\cos\theta$$

$$C_{y}^{-} = (\beta^{+} - \beta^{-})\cos\psi\sin\theta$$

11. Cancel  $u_x^+$  and  $u_y^-$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] + C_{\zeta}[u_{\zeta}] = C_{z}^{+}u_{z}^{+} + C_{z}^{-}u_{z}^{-} + C_{x}^{-}u_{x}^{-} + C_{y}^{+}u_{y}^{+}$$
(3.20)

$$C_{\eta} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta \cos \theta$$

$$C_{\zeta} = \beta^{+} \cot \psi \cos^{2} \theta + \beta^{-} \cot \psi \sin^{2} \theta$$

$$C_{z}^{+} = \beta^{+} \sin \psi + \beta^{+} \cos \psi \cot \psi \cos^{2} \theta + \beta^{-} \cos \psi \cot \psi \sin^{2} \theta$$

$$C_{z}^{-} = -\beta^{-} \sin \psi - \beta^{+} \cos \psi \cot \psi \cos^{2} \theta - \beta^{-} \cos \psi \cot \psi \sin^{2} \theta$$

$$C_{x}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{y}^{+} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

12. Cancel  $u_x^+$  and  $u_y^+$  from Eqs. (3.7),(3.8) and (3.9) to have

$$[\beta u_{\xi}] + C_{\eta}[u_{\eta}] + C_{\zeta}[u_{\zeta}] = C_{z}^{+}u_{z}^{+} + C_{z}^{-}u_{z}^{-} + C_{x}^{-}u_{x}^{-} + C_{y}^{-}u_{y}^{-}$$
(3.21)

where

$$C_{\eta} = 0$$

$$C_{\zeta} = \beta^{+} \cot \psi$$

$$C_{z}^{+} = \beta^{+} / \sin \psi$$

$$C_{z}^{-} = -\beta^{-} \sin \psi - \beta^{+} \cos \psi \cot \psi$$

$$C_{x}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \cos \theta$$

$$C_{y}^{-} = (\beta^{+} - \beta^{-}) \cos \psi \sin \theta$$

One of these twelve conditions will be chosen in addition to Eq.(3.2) for the solution of the a pair of fictitious values. The details of this solution will be illustrated in the x- direction with interface conditions (3.2) and (3.10). The solution with the other conditions can be accomplished similarly.

Consider a situation as shown in Figure 3.1, where the interface intersects a xmesh line at  $s = (x_o, y_o, z_o)$  between (i, j, k) and (i + 1, j, k), which are then classified as irregular points. The solution of interior domain will be extended to the point



**Figure 3.1:** Local topology around irregular point (i, j, k). The interface crosses the x-mesh line at the point  $S = (x_o, y_o, z_o)$  between (i, j, k) and (i + 1, j, k), which are two irregular points, on which two fictitious values,  $f_{i,j,k}$  and  $f_{i+1,j,k}$  are defined (marked with green dots). Two auxiliary lines (dashed line) are sketched passing through S, one on the x - y plane and the other on the x - z plane. Two auxiliary points (in empty circle) (o, j, k + 1) and (o, j, k + 2) are placed on the auxiliary line on the x - z plane to facilitate the discretization of  $u_z^-$ ; Also, two auxiliary points (o, j, k + 1) and (o, j, k + 2) are placed on the auxiliary line on the x - y plane to facilitate the discretization of  $u_y^-$ .

(i + 1, j, k) as  $f_{i+1,j,k}$  whereas the exterior domain will be continuated to the point (i, j, k) as  $f_{i,j,k}$ . With these two fictitious values the interface condition (3.2) can be approximated to be

$$(w_{0,i}^{+}f_{i,j,k} + w_{0,i+1}^{+}u_{i+1,j,k} + w_{0,i+2}^{+}u_{i+2,j,k}) - (w_{0,i-1}^{-}u_{i-1,j,k} + w_{0,i}^{-}u_{i,j,k} + w_{0,i+1}^{-}f_{i+1,j,k}) = [u].$$

$$(3.22)$$

In a similar way we can also get the approximations to the partial derivatives  $u_x^$ and  $u_x^+$  in Eq. (3.10). The approximation of the other two derivatives  $u_y^-$  and  $u_z^$ at the point  $(x_o, y_o, z_o)$ , however, remains a major problem because  $(x_o, y_o, z_o)$  is not located on any y- mesh line or z- mesh line. To overcome the difficulty due to the lack of real grid points for finite difference, we add two auxiliary lines passing  $(x_o, y_o, z_o)$ , one on the x - y plane and the other on the x - z plane, and set two auxiliary points on each auxiliary line to support a one-sided finite difference for  $u_y^$ and  $u_z^-$ . This makes it possible to approximate Eq. (3.10) as

$$[\beta u_{\xi}] - \beta^{-} \tan \psi[u_{\zeta}] + C_{\eta}[u_{\eta}] = C_{x}^{+}(w_{1,i}^{+}f_{i,j,k} + w_{1,i+1}^{+}u_{i+1,j,k} + w_{1,i+2}^{+}u_{i+2,j,k}) + C_{x}^{-}(w_{1,i-1}^{-}u_{i-1,j,k} + w_{1,i}^{-}u_{i,j,k} + w_{1,i+1}^{-}f_{i+1,j,k}) + C_{y}^{-}(p_{o,j}u_{o,j,k} + p_{o,j-1}u_{o,j-1,k} + p_{o,j-2}u_{o,j-2,k}) + C_{z}^{-}(q_{o,k}u_{o,j,k} + q_{o,k+1}u_{o,j,k+1} + q_{o,k+2}u_{o,j,k+2}),$$

$$(3.23)$$

where  $p_{o,:}$  are the finite difference weights for  $u_y^-$  at auxiliary nodes (o, j, k), (o, j - 1, k), (o, j - 2, k) and  $q_{o,:}$  are the finite difference weights for  $u_z^-$  at auxiliary nodes (o, j, k), (o, j, k + 1) and (o, j, k + 2). Here  $u^+(o, j, k)$  is the limiting value of the solution u(x, y, z) at the point S from the + side of the interface. Values  $f_{i,j,k}$  and  $f_{i+1,j,k}$  can be solved from Eqs. (3.22) and (3.23) as the linear combinations of the solution values at the real grid points  $u_{i-1,j,k}, u_{i,j,k}, u_{i,j,k}, u_{i,j,k}$  and the auxiliary points

$$u_{o,j,k}, u_{o,j-1,k}, u_{o,j-2,k}, u_{o,j,k}, u_{o,j,k+1}, u_{o,j,k+2}$$

as well as the four given jumps  $[u], [\beta u_{\xi}], [u_{\eta}], [u_{\zeta}]$ . These values are given in the following general form

$$\begin{aligned} f_{i,j,k} &= C_1^i u_{i-1,j,k} + C_2^i u_{i,j,k} + C_3^i u_{i+1,j,k} + C_4^i u_{i+2,j,k} \\ &+ C_5^i u_{o,j,k} + C_6^i u_{o,j+1,k} + C_7^i u_{o,j+2,k} + C_8^i u_{o,j,k} + C_9^i u_{o,j,k-1} \\ &+ C_{10}^i u_{o,j,k-2} C_{11}^i [u] + C_{12}^i [\beta u_{\xi}] + C_{13}^i [u_{\eta}] + C_{14}^i [u_{\zeta}], \end{aligned}$$
(3.24)  
$$f_{i+1,j,k} &= C_1^{i+1} u_{i-1,j,k} + C_2^{i+1} u_{i,j,k} + C_3^{i+1} u_{i+1,j,k} + C_4^{i+1} u_{i+2,j,k} \\ &+ C_5^{i+1} u_{o,j,k} + C_6^{i+1} u_{o,j+1,k} + C_7^{i+1} u_{o,j+2,k} + C_8^{i+1} u_{o,j,k} + C_9^{i+1} u_{o,j,k-1} \\ &+ C_{10}^{i+1} u_{o,j,k-2} + C_{11}^{i+1} [u] + C_{12}^{i+1} [\beta u_{\xi}] + C_{13}^{i+1} [u_{\eta}] + C_{14}^{i+1} [u_{\zeta}]. \end{aligned}$$
(3.25)

In these two linear representations of the fictitious values, the solution values at auxiliary points have not yet been defined. In MIB, we obtain these values by interpolating the available normal grid points on the same subdomain. For example, u(o, j - 1, k) can be interpolated on u(i - 1, j - 1, k), u(i, j - 1, k) and u(i + 1, j - 1, k), and u(o, j - 2, k) can be interpolated on u(i - 1, j - 2, k), u(i, j - 2, k) and u(i + 1, j - 2, k). Similarly, auxiliary values u(o, j, k+1) and u(o, j, k+2) are interpolated at u(i-2, j, k+1), u(i - 1, j, k + 1), u(i, j, k + 1) and u(i - 1, j, k + 2), u(i, j, k + 2), u(i + 1, j, k + 2), respectively. It is also noticed that the auxiliary value  $u^+(0, j, k)$  can be interpolated as  $(w_{0,i}^+f_{i,j,k} + w_{0,i+1}^+u_{i+1,j,k} + w_{0,i+2}^+u_{i+2,j,k})$  or  $(w_{0,i-1}^-u_{i-1,j,k} + w_{0,i}^-u_{i,j,k} + w_{0,i+1}^-f_{i+1,j,k}) + [u]$ , see Eq.(3.22). The expansion coefficients of the fictitious values on these auxiliary points can therefore be distributed to their respective interpolation nodes. Let

$$\begin{split} u(o, j, k) &= (w_{0,i-1}^{-}u_{i-1,j,k} + w_{0,i}^{-}u_{i,j,k} + w_{0,i+1}^{-}f_{i+1,j,k}) + [u], \\ u(o, j-1, k) &= \mathbf{I}_{j-1} \cdot (u(i-1, j-1, k), u(i, j-1, k), u(i+1, j-1, k))^{T}, \\ u(o, j-2, k) &= \mathbf{I}_{j-2} \cdot (u(i-1, j-2, k), u(i, j-2, k), u(i+1, j-2, k))^{T}, \\ u(o, j, k+1) &= \mathbf{I}_{k+1} \cdot (u(i-2, j, k+1), u(i-1, j, k+1), u(i, j, k+1))^{T}, \\ u(o, j, k+2) &= \mathbf{I}_{k+2} \cdot (u(i-1, j, k+2), u(i, j, k+2), u(i+1, j, k+2))^{T}, \end{split}$$

where I is the vector consisting of corresponding interpolation weights. Then

$$\begin{aligned} f_{i,j,k} &= C_{1}^{i} u_{i-1,j,k} + C_{2}^{i} u_{i,j,k} + C_{3}^{i} u_{i+1,j,k} + C_{4}^{i} u_{i+2,j,k} \\ &+ (C_{5}^{i} + C_{8}^{i}) (w_{0,i-1}^{-} u_{i-1,j,k} + w_{0,i}^{-} u_{i,j,k} + w_{0,i+1}^{-} f_{i+1,j,k} + [u]) \\ &+ C_{6}^{i} \cdot \mathbf{I}_{j-1} \cdot (u(i-1,j-1,k), u(i,j-1,k), u(i+1,j-1,k))^{T} \\ &+ C_{7}^{i} \cdot \mathbf{I}_{j-2} \cdot (u(i-1,j-2,k), u(i,j-2,k), u(i+1,j-2,k))^{T} \\ &+ C_{9}^{i} \cdot \mathbf{I}_{k+1} \cdot (u(i-2,j,k+1), u(i-1,j,k+1), u(i,j,k+1))^{T} \\ &+ C_{10}^{i} \cdot \mathbf{I}_{k+2} \cdot (u(i-1,j,k+2), u(i,j,k+2), u(i+1,j,k+2))^{T} \\ &+ C_{9}^{i} [u] + C_{10}^{i} [\beta u_{\zeta}] + C_{11}^{i} [u_{\eta}] + C_{12}^{i} [u_{\zeta}], \end{aligned}$$

$$(3.26)$$

$$f_{i+1,j,k} = C_1^{i+1} u_{i-1,j,k} + C_2^{i+1} u_{i,j,k} + C_3^{i+1} u_{i+1,j,k} + C_4^{i+1} u_{i+2,j,k} + (C_5^{i+1} + C_8^{i+1}) (w_{0,i-1}^{-} u_{i-1,j,k} + w_{0,i}^{-} u_{i,j,k} + w_{0,i+1}^{-} f_{i+1,j,k} + [u]) + C_6^{i+1} \cdot \mathbf{I}_{j-1} \cdot (u(i-1,j-1,k), u(i,j-1,k), u(i+1,j-1,k))^T + C_7^{i+1} \cdot \mathbf{I}_{j-2} \cdot (u(i-1,j-2,k), u(i,j-2,k), u(i+1,j-2,k))^T + C_9^{i+1} \cdot \mathbf{I}_{k+1} \cdot (u(i-2,j,k+1), u(i-1,j,k+1), u(i,j,k+1))^T + C_{10}^{i+1} \cdot \mathbf{I}_{k+2} \cdot (u(i-1,j,k+2), u(i,j,k+2), u(i+1,j,k+2))^T + C_{11}^{i+1} [u] + C_{12}^{i+1} [\beta u_{\xi}] + C_{13}^{i+1} [u_{\eta}] + C_{14}^{i+1} [u_{\zeta}].$$
(3.27)

This finished the solution of fictitious values  $f_{i,j,k}$  and  $f_{i+1,j,k}$  in the x-direction. The standard finite difference scheme for  $u_{xx}$  at the points (i, j, k) and (i + 1, j, k)will then be modified accordingly by incorporating these fictitious values

$$u_{xx} = \beta^{-} \frac{f_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h^{2}} \quad \text{at} \ (i,j,k), \tag{3.28}$$

$$u_{xx} = \beta^{+} \frac{u_{i+2,j,k} - 2u_{i+1,j,k} + f_{i-1,j,k}}{h^{2}} \quad \text{at} \ (i+1,j,k), \tag{3.29}$$

and replacing  $f_{i,j,k}$  and  $f_{i+1,j,k}$  with their respective representations in Eq.(3.26) or Eq.(3.27).

Although solved in the x-direction, the fictitious value  $f_{i,j,k}$  or  $f_{i+1,j,k}$  represents the smooth continuation of the solution on the respective subdomain to the current irregular point. Therefore it can be used for the formulation of difference schemes for any partial derivatives rather than  $u_{xx}$ . In particular, if the central difference scheme of  $u_{yy}$  at some irregular point in the subdomain  $\Omega^+$  involves the grid point (i, j, k), the fictitious value  $f_{i,j,k}$  can be directly supplied. This makes it possible to handle an interface with large Gaussian curvature such that a fictitious value can only be solved in one particular direction while it will be used for the central difference scheme in other direction(s).

**Comments 3.1.1.** It is noted that  $\tan \theta$ ,  $\cot \theta$ ,  $\tan \phi$  or  $\cot \phi$  is undefined for particular values of  $\theta$  or  $\phi$ . This happens when the normal vector at the interface is on a coordinate plane, i.e., on the x - y, x - z or y - z plane. One can then apply the MIB

method formulated for 2-D problems in Chapter 2 to solve for the fictitious values in these cases.

Further reduction to the 1-D problem might happen if the normal vector is aligned with a coordinate direction. The original two interface condition instead of reorganized ones are to be used to solve for the fictitious values in this case, as described in Section 1.2.



**Figure 3.2:** Local topology with interface intersecting at a grid point (i, j, k). Two fictitious values,  $f_{i,j,k}$  and  $f_{i+1,j,k}$  are defined (marked with green dots).

**Comments 3.1.2.** In case that the interface crosses the mesh line at a grid point, the related fictitious values can also be solved in a similar manner as described in Commment 2.1.2. In Figure 3.2, for example, one fictitious value  $f_{i,j,k} = u(i, j, k) + [u]$  and the other fictitious value  $f_{i+1,j,k}$  can be solved with interface condition 3.13, in which  $u_y^-$  is to be directly approximated using  $u_{i,j,k}, u_{i,j-1,k}, u_{i,j-2,k}, u_z^-$  is to be approximated using  $u_{i,j,k}, u_{i+1,j,k}$  can be approximated using  $u_{i-1,j,k}, u_{i,j,k}, f_{i+1,j,k}$  and  $u_x^+$  is to be approximated using  $f_{i,j,k}, u_{i+1,j,k}, u_{i+2,j,k}$ . Auxiliary lines and points are therefore no longer needed.

### **3.2** Numerical Experiments

Example 3.2.1. We consider a 3-D Poisson equation

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f(x, y, z)$$

$$(3.30)$$

defined in a cube  $[-1,1] \times [-1,1] \times [-1,1]$  with a spherical interface

$$r^2 \equiv x^2 + y^2 + z^2 = \frac{1}{4}.$$

The exact solution is designed to be

$$u(x, y, z) = \begin{cases} (x^2 + y^2 + z^2)^2 & r \le 0.5\\ 2(x^2 + y^2 + z^2)^2 & otherwise \end{cases}$$
(3.31)

with the diffusion coefficient

$$\beta(x,y) = \begin{cases} 1 & r \le 0.5 \\ 80 & otherwise \end{cases}$$
(3.32)

Such a designated solution forces the discontinuous inhomogeneous term f(x, y, z) to be

$$f(x,y) = \begin{cases} 20(x^2 + y^2 + z^2) & r \le 0.5\\ 3200(x^2 + y^2 + z^2) & otherwise \end{cases}.$$
 (3.33)

Table(3.1) lists the global errors in the infinity norm.

 Table 3.1: Numerical convergence test for Example(3.2.1)

$n_x = n_y$	$L_{\infty}$	Order
20	1.56E-2	
40	4.16E-3	1.91
80	8.93E-4	2.22

Example 3.2.2. We consider a 3-D Poisson equation

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f(x, y, z)$$
(3.34)

defined in a cube  $[-1,1] \times [-1,1] \times [-1,1]$  with an ellipsoidal interface

$$\frac{x^2}{0.49} + \frac{y^2}{0.16} + \frac{z^2}{0.09} = 1.$$

The exact solution is designed to be

$$u(x, y, z) = \begin{cases} \sin(x)\cos(y)\sin(z) & r \le 0.5\\ \exp(x^2 + y^2 + z^2) & otherwise \end{cases}$$
(3.35)

with the diffusion coefficient

$$\beta(x,y) = \begin{cases} 1 & r \le 0.5 \\ & & , \\ 10 & otherwise \end{cases}$$
(3.36)

from which one can derive the inhomogeneous term f(x) and the jumps [u] and  $[\beta u_n]$ required for the solution of the Poisson equation.

Table(3.2) lists the global errors in the infinity norm.

 Table 3.2: Numerical convergence test for example(3.2.2)

$n_x = n_y$	$L_{\infty}$	Order
20	9.33E-3	
40	2.42E-3	1.95
80	5.98E-4	2.02

### 3.3 Comparison of MIB and IIM

It is interesting to examine the fundamental difference (or similarities if any) between MIB and IIM since both of them manage to impose continuity conditions across the interface by locally modifying the finite difference schemes according to the interface conditions. MIB uses only the zeroth- and first-order interface conditions, while IIM, by contrast, derives a number of additional second-order interface conditions. In IIM, as the solution on either side of the interface is approximated by a multi-dimensional Taylor expansion with respect to the solution at the interface, it allows (actually requires) one to choose in the whole domain the grid points near the host irregular point. For 2-D problem, one needs 10 conditions in IIM to fully determine the 10 coefficients in two third-order Taylor polynomials of two variables. These 10 conditions are chosen to be 4 interface conditions (2 original ones plus 2 derived second-order ones) and 6 independent expansions of Taylor polynomials at 6 grid points. More grid points can be chosen to provide an under-determined linear system which essentially provides necessary space for the optimization of these polynomial coefficients. Such flexibility, however, does not exist for the MIB method as here one always chooses a primary direction to solve for the fictitious values and this is equivalent to the determination of 1-D polynomials, as illustrated in Section 2.4 of interpolation formulation. As a result, one has to choose different sets of grid points for the approximation of partial derivatives with respect to x and those with respect to y, therefore MIB has more grid points involved in the finite difference schemes at irregular points than IIM and the resultant linear system might be more ill-conditioned.

On the other hand, as the fictitious values are always solved in one direction, the generalization of MIB to higher order is straightforward and the small linear system for fictitious values is always invertible. For the IIM method, however, the generalization to higher order might be hindered by the lack of a sufficient number of interface conditions for the determination of more coefficients in higher order multi-dimensional polynomials. For example, to determine two fifth-order 2-D Taylor polynomials, a total of 28 coefficients are to be determined. It remains a problem to derive a sufficient number of high order interface conditions, and the linear system for the coefficients of these high-order multi-dimensional polynomials may be not invertible.

### Chapter 4

# Poisson-Boltzmann Equation and Interface Methods

In this chapter, we will discuss the Poisson-Boltzmann equation from the perspective of implicit solvent modeling, of which the fundamental theoretical underpinning is the Debye-Hückel theory describing the distribution of the electrostatic potential  $\phi(x)$  in 3-D space. The importance of the electrostatic potential motivated many interests of scientists over the years and reviews of these efforts appear on a frequent basis. In the first section, we will discuss the major considerations and critical assumptions in the derivation of the Poisson-Boltzmann equation. Three most important mathematical issues will be highlighted in the next section. One of them, the discontinuity of the dielectric constant, will be further addressed, but this section is mainly devoted to the brief review of the analytical solution and the numerical methods for the Poisson-Boltzmann equation. A short discussion on the dielectric interface will be given in the last section as this interface is the position where discontinuity occurs and therefore is critical to the formulation of the MIB method and to the solution of the Poisson-Boltzmann equation.

# 4.1 Implicit Solvent Modeling and the Poisson-Boltzmann Equation

The importance of electrostatic interactions to chemistry, physics, material science and biology has been well established. Because of the crucial roles of electrostatic properties in investigating electrostatic binding and solvation energies [23], electrostatic steering of ligands toward proteins [67], protein conformational change [73, 85], folding stability [73, 95] and many other important subjects, the accurate evaluation of electrostatic properties has therefore always been a major concern in molecular/structural biology. This task is further complicated by the consideration of the solvent environment as most biochemical reactions of biomolecules occur in salty solution or at interfaces between lipid bilayers and aqueous phases. A complete energetic description of biomolecular processes therefore has to include the surrounding aqueous solvent environment, either explicitly or implicitly [56]. With an explicit model the biomolecule(s) to be studied are embedded in a large number of solvent molecules. The interaction between two charges is described by Coulomb's law and the total electrostatic energy of the system is derived as the sum of these mutual Coulomb interactions. The explicit model might provide the most accurate and detailed description of the solvent but it also significantly increases the computational cost as the number of solvent particles can be more than 90% for a real simulation [4]. Some quantities, such as solvation energy, converge slowly as one has to average over all degrees of freedom of all solvent particles. These critical limitations reduce the time scale, the size of the molecule and the amount of sampling that can be achieved in practical simulations. An implicit solvent model, however, describes the solvent as a structureless dielectric continuum and replaces the solvent interactions with an equivalent energetic term based on the mean field behavior. Similarly, the biomolecule itself can also be modeled as dielectric media with partial charges at atomic positions. The electrostatic potential induced by the partial charges inside the molecule and the free ions inside the solvent is therefore governed by a Poisson equation:

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) = -\frac{4\pi}{\epsilon_0}(\rho^f(x) + \sigma(x))$$
(4.1)

where  $\epsilon(x)$  is the spatial dependent dielectric constant and  $\epsilon_0$  is the dielectric permittivity of the vacuum. Here  $\rho^f(x)$  is the fixed charge density inside the molecule and  $\sigma(x)$  is the charge density of the free ions among the solvent. The distribution of free ions depends on the electrostatic field. Therefore the charge density inside the solvent is a function of  $\phi(x)$ . In particular, one can relate the density of the ionic particles subjected to the external field with  $\sigma_0(x)$ , the ionic density in the absence of external electrostatic potential field, via

$$\sigma(x) = e^{-E(x)/kT} \sigma_0(x), \tag{4.2}$$

where a Boltzmann distribution  $e^{-E(x)/kT}$  is used to describe the redistribution of the charge density in the background of potential of mean force (PMF) E(x), which represents the average effect of the entire system on a single particle. This PMF can be further related with the average electrostatic potential if one assumes that the tendency of free ions toward regions of low electrostatic potential energy is dependent of the magnitude and sign of its charge. With this crucial assumption, usually referred to as Gouy-Chapman or Debye-Hückel theories of ion distributions, the ionic potential of the mean force is replaced by the product of average electrostatic potential and the charge of the ion. Then the Poisson equation becomes the Poisson-Boltzmann equation:

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) = -\frac{4\pi}{\epsilon_0} \left[ \rho^f(x) + \sum_i c_i z_i q e^{-z_i q \phi(x)/kT} \lambda(x) \right], \tag{4.3}$$

where  $c_i$  and  $z_i$  are the bulk concentration and valence of ion *i*, respectively. *q* is the unit(proton) charge, *k* is the Boltzmann constant, *T* is the absolute temperature and  $\lambda(x)$  is the accessibility of the ion at position *x*. There are a number of variations of

this primitive Poisson-Boltzmann equation under different assumptions. For instance, if the potential is weak, equation (4.3) can be linearized to be

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) = -\frac{4\pi}{\epsilon_0} \left[ \rho^f(x) - \frac{\sum_i c_i z_i^2 q^2 \phi(x)}{kT} \lambda(x) \right]; \tag{4.4}$$

while for symmetric ionic solution with bulk concentration c and valency z, equation (4.3) can be reduced to be

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) = -\frac{4\pi}{\epsilon_0} \left[ \rho^f(x) - 2czq \sinh\left(\frac{zq\phi(x)}{kT}\right)\lambda(x) \right].$$
(4.5)

The so-called Debye length

$$l_D = \sqrt{\frac{\epsilon_0 kT}{8\pi c z^2 q^2}}$$

can be derived from the linear Poisson-Boltzmann equation for symmetric ionic solution

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) + \frac{8\pi\epsilon z^2 q^2}{\epsilon_0 kT}\phi(x)\lambda(x) = -\frac{4\pi}{\epsilon_0}\rho^f(x)\lambda(x), \tag{4.6}$$

where  $\frac{8\pi cz^2q^2}{\epsilon_0 kT}$  has a dimension of  $1/(length)^2$ . Moreover, it can be found that

$$\nabla \cdot (\nabla \phi(x)) + \frac{8\pi c z^2 q^2}{\epsilon_0 k T} \phi(x) = 0, \qquad (4.7)$$

admits a general solution

$$\phi(x) = A e^{-x/l} D \tag{4.8}$$

which describes the exponential decay of the electrostatic potential in ionic solution. Here  $k_D = 1/l_D$  is the Debye screening constant.

There are a number of significant assumptions in the above derivation of the Poisson-Boltzmann equation. The critical examination on these assumptions can provide not only the deep understanding of the PBE but also motivations to the modification of Poisson-Boltzmann theory for other important or specific circumstances. Below we will summarize these assumptions and the modifications or extensions concerning these assumptions:

1. Fixed charge density in biomolecules.

This assumption does not consider the re-arrangement of the polar and charged groups in the external electrostatic potential field. The average polarizability of the protein is measured by its dielectric constant which also macroscopically characterizes the relaxation of the protein structure in response to the charge perturbation. Therefore the neglection of polarizability of the charged groups in an implicit model would lead to considerable deviation of the dielectric constant inside the biomolecules. Shark [72] and Simonson [76] have demonstrated that a larger dielectric constant should be used in solving the Poisson-Boltzmann equation if polarizability is neglected.

2. Uniform dielectric constant in biomolecules.

The issue about the dielectric constant in biomolecules is closely related to the model of fixed charge density. As mentioned above, the dielectric constant of a protein is a macroscopic quantity to characterize its relaxation in response to charged perturbations. Therefore the hydrophobic core and the charged protein side chains at the protein/solvent interface might have different response and different dielectric constant, as the latter are usually undergoing large, mutually uncorrelated and anisotropic motions. In this regard, molecular dynamics (MD) simulations are usually used for the detailed prediction of the protein dielectric properties in solution. Simonson illustrated that the Fröhlich-Kirkwood theory could provide a consistent prediction with the MD simulation if the charge portion of the side chains are treated as part of the solvent and assign the remaining part of the protein a uniform dielectric constant of 3-4 [76]. A more complicated inhomogeneous dielectric map can be generated by assigning every amino acid a different dielectric constant based on its unique intrinsic polarizability [75, 77].

There are more physical considerations of the effects of polarizability on the di-

electric function besides the layered dielectric model for proteins. For example, a local dielectric function  $\epsilon(r, r')$  (depending on the response point r and the perturbation point r') or  $\epsilon(|r - r'|)$  (depending on their separation only) rather than a single dielectric constant can be supplied to the Poisson-Boltzmann equation to introduce the polarization induced by the external field, the fluctuations of electronic polarization within an individual molecule, and the local field in a cluster of non-overlapping molecules [34]. Non-local dielectric functions have been used to account for the effects of solvent shells surrounding charged species in work on electron transfer, the double layer at electrode-solvent interface and intermolecular interactions near interfaces.

#### 3. Singular charge density in biomolecules.

The physical modeling of charge distribution inside biomolecules remains an open question. It appears that there is a tendency to use more physical finitesize models such as homogeneously charged sphere, the Fermi two-parameter charge distribution or the Guassian charge distribution instead of the usual point charge model [80]. The advantages of these finite-size models to the solution of the Poisson-Boltzmann equation, however, have not been explored in depth. Particularly, because the characteristic size in these finite-size models is of the same scale as the root-mean-square (RMS) radius of the nucleus

$$\sqrt{\langle R^2 \rangle} = (0.836A^{1/3} + 0.570) \times 10^{-15} meters$$

it is very difficult to provide a discrete description of these finite-size models on a mesh of size ranging 0.1Å to 0.5Å. It might be possible to investigate the consequence of these finite-size models in the framework of the boundary element method, whereby the Green function for the point charge can be simply replaced by the smooth potential corresponding to a finite-size model.

4. Boltzmann distribution of the ions in the solvent.

When a biomolecule with charged surface is immersed in an electrolyte solution

(ionic solution, for example), ions of opposite charges to the surface are electrostatically attracted to the surface while ions of like charges are repelled. The counter ions are not bounded to the surface but remain dispersed and mobile in the vicinity of the surface due to their thermal motions. According to the Gouy-Chapman-Stern model, the region next to the surface is called the Stern layer where the distribution of ions is determined by the Coulomb interactions and special absorption. The region next to the stern layer is called the diffuse layer and ions in this layer can move freely (diffuse) in any direction. This spatial separation of ions is denoted as the electric double layer [37].

In addition to the local density distribution represented by the electric double layer, the packing effects caused by the finite size of the ion and solvent molecules can also be considerable and the resultant spatial variations of ionic density and strong correlation between ion-ion, ion-solvent or solvent-solvent molecules can not be modeled by a Boltzmann distribution. A number of modifications have been developed to address the deviation of real ionic distribution and the Boltzmann model. For instance, Eigen and Wicke [16] modified the Boltzmann distribution by introducing steric and confinement effects of ions, while Luo et.~al.~[54] conducted molecular dynamics simulations including the solvent and ionic structures to obtain the potential of mean force (PMF) of a single ion. This PMF is then supplied to the generalized Poisson-Boltzmann equation (combining Eq.(4.2) and Eq.(4.1))

$$\nabla \cdot (\epsilon(x)\nabla\phi(x)) = -\frac{4\pi}{\epsilon_0}(\rho^f(x) + e^{-E(x)/kT}\sigma_0(x)).$$
(4.9)

The solved real ionic distributions were found to agree with the x-ray reflectivity measurements.

# 4.2 Analytical Solutions and Numerical Techniques for the Poisson-Boltzmann Equation: A Brief Review

A close look at the Poisson-Boltzmann equation will reveal four major features, namely, the irregular interface between the solute and the solvent, the discontinuous dielectric function on this dielectric interface, the singular partial charge distribution and the nonlinearity in the case the strong potential. We are particularly interested in the handling of the dielectric constant with discontinuity at the irregular molecular surface, not only because dielectric properties of macromolecules are crucial to their stability and activity but also because this discontinuity poses one of the most difficult challenges in seeking the numerical solution of the Poisson-Boltzmann equation and all of the current methods for the Poisson-Boltzmann equation are lacking of specific treatments of this discontinuity. In particular, we are interested in the incorporation of these two conditions

$$\phi_p = \phi_w, \quad \epsilon_p \frac{\partial \phi_p}{\partial n} = \epsilon_w \frac{\partial \phi_w}{\partial n}, \tag{4.10}$$

at the dielectric interface into the numerical solution of the Poisson-Boltzmann equation. The first condition describes the continuity of the electrostatic potential across the interface while the second characterizes the balance of the derivative of the electrostatic potential field at the interface. The subscript p denotes protein and w denotes solvent. We will refer to the second as the potential flux continuity condition, which is not considered in any finite-difference or finite-element based Poisson-Boltzmann solvers.

Because of the complicated dielectric interface and charge distribution, the analytical solution to the Poisson-Boltzmann equation is not available for almost all the pratical systems <sup>1</sup> and a variety of computational techniques and software packages

<sup>&</sup>lt;sup>1</sup>There are very few systems having analytical solutions of linear Poisson-Boltzmann equation. To the author's best knowledge, they are:

have been developed for the numerical solution fo the Poisson-Boltzmann equation. These numerical techniques cover all the major discretization methods such as finite difference, finite element, boundary element and Fourier spectral methods. One of the first such Poisson-Boltzmann solvers was developed by Orttung [62] in 1979 based on the finite element method. After that a number of other Poisson-Boltzmann solvers emerged, such as a finite difference method by Warwicker and Watson [86] and a boundary element technique by Zauhar and Morgan [93]. The successful applications of these solvers, espically the method by Warwicker and Watson [86], on the biological macromolecules, motivated the establishment of the numerical technique of higher accuracy and efficiency. A list of major Poisson-Boltzmann solvers with a short description of each solver was given by Baker and is reproduced here.

The construction of accurate and efficient numerical algorithms is restricted mainly by the point charges and the discontinuous dielectric constant at the molecular surface. Despite the slow advancement of numerical techniques for the singular charge density, a number of techniques were devised to improve the accuracy of the solution near the dielectric interface. For example, APBS manages to discretize the equation on a mesh *a priori* refined in the vicinity of the molecular surface. Some robust *a posteriori* error estimation is used to drive the adaptive mesh refinement when multilevel meshes are used in APBS [27]. Similarly, Delphi [24] attmpts to improve the accuracy of the solution, especially in the vicinity of the dielectric interface, by first solving the Poisson-Boltzmann equation in a computational domain much larger than the molecule(s) enclosed. Then, the second round of computation is carried out on the smaller domain with a refined mesh. The boundary condition of this second

<sup>1.</sup> A single sphere with single or multiple point charges. They are to be discussed in this dissertation.

<sup>2.</sup> A single sphere with a uniform distributed charge on the surface [28]

<sup>3.</sup> Multiple point-charged molecule with complete solvent penetration [28].

<sup>4.</sup> Two centered charged spheres. It seems that an analytical solution might be obtained using the bispherical coordinate system. Despite a number of research articles [20, 61, 78] in the past decades, there is no simple solution of electrostatic potential in closed form for this case Some solutions in semi-analytical form are series of slow convergence and are too complicated to be implemented.

<sup>5.</sup> Systems with cylindrical geometries [57, 74].

run comes from the last computation. Such a 'focusing' procedure can be hierarchically implemented to acquire a sufficient resolution and accuracy near the molecular surface. All these techniques are based on the local mesh refinement and there is no special treatment of the interface conditions in the fundamental discretization scheme.

A noteworthy class of numerical techniques for the Poisson-Boltzmann equation is the boundary element method. In this approach, two boundary integrals, one in the interior of the molecule and the other in solvent, are defined. The electrostatic potential and its normal gradients are determined from the matching of both interface conditions on the molecular surface with these two integrals. Once the surface potential is solved, the potential in the entire domain can be analytically computed. Theoretically, boundary element approach is superior to all the other techniques because it allows the discretization of the equation in lower dimension, as both interface conditions are satisfied, because the singularity of the point charges are removed through boundary integrals, and because the far field boundary conditions are enforced exactly in an integral. The very first boundary element method proposed by Zauhar and Morgan [93] assumes that the ionic strength is zero but this assumption was soon removed in the formulation of Rashin [66]. The first boundary element approach based on Green's function for the Poisson-Boltzmann equation, however, probably is due to Juffer *et. al.* [66].

Compared with the wide applications of finite difference or finite element based Poisson-Boltzmann solvers, the applications of boundary element methods are found to be limited. This might be explained by a number of fundamental issues in the boundary element methods for the Poisson-Boltzmann equation. For example, the Green's function of the Poisson-Boltzmann equation may be difficult to find if the molecules have complex internal structures. Additional volume integrals have to be calculated if the boundary element method is used to solve the nonlinear Poisson-Boltzmann equation [9]. The final linear system might be ill-conditioned if weak formulation of the linear Poisson-Boltzmann equation or the numerical quadrature in the surface integral is not optimized. In particular, the matching of the two interface conditions with the surface integrals gives rise to  $4N^2$  influence coefficients where N is the number of surface elements. The calculation and storage of this large number of quantities severely limits the efficiency of the boundary element method. It has been demonstrated that the application of a fast multipole technique could reduce the computational complexity of these quantities to an order of  $N \log(N)$  [8]. It is expected that further investigation on the fast multipole or related method such as tree code, and on the fast calculation of the Maxwell stress tensor on the molecular surface [53] could provide necessary thrust to the application of the boundary element method on the numerical solution of the Poisson-Boltzmann equation.

#### 4.3 Dielectric Interface

The definition of dielectric interface is one of critical issues in the implicit solvent model. Any biomolecular surface, such as van der Waals surface, solvent accessible surface, solvent-excluded surface [12], Gaussian surface [88], multiresulution molecular surface [87], can be chosen as the dielectric interface in computation but the choice of which biomolecular surface to use as the dielectric interface is still open to debate. Most of these biomolecular surfaces consider only the constant van der Waals radii of the atoms and the radius of the probe which is the spherical model of the solvent molecule. They are defined *a priori* and are supplied into a particular implicit solvent model, the Poisson-Boltzmann equation for example, to determine the energetic state of the system. In contrast to these surface definitions, the biomolecular surface can be derived from the minimization of an energy functional of the solute cavity shape [15]. In this case, the biomolecular surface is the output of the implicit solvent model rather than its input.

The implementation of a well-defined molecular surface in the solution of the Poisson-Boltzmann equation depends on the underlying numerical technique. In finite difference based Poisson-Boltzmann solvers such as DelPhi, MEAD and PBEQ, the molecular surface is used to define the map of the dielectric function only and the position of the molecular surface is not explicitly computed. In PBEQ, for example, a set of probes centered at the van der Waals surface are defined, and any grid point enclosed in this envelope is regarded to be in the solute (otherwise it is in the solvent). In APBS with finite element discretization, nevertheless, the position of the molecular surface is explicitly calculated to provide the exact position for local mesh refinement. None of these Poisson-Boltzmann solvers calculates the normal direction of the molecular surface as they do not consider the discontinuity of the potential gradient at the molecular surface.

Few molecular surface generators calculate the normal direction of the molecular surface. We notice that the normal vector and other geometrical quantities can be derived from the spherical harmonic expansion of the molecular surface [58]. The slow convergence of such expansion toward an accurate description of the molecular surface prevents us from implementing the same molecular surface as other Poisson-Boltzmann solvers. The MSMS program due to Michel Sanner et. al. provides an efficient algorithm for the molecular surface, its normal directions, its total area and its triangulation [68]. We will implement the MSMS surface in our solution of the Poisson-Boltzmann equation. However, we do prefer the dielectric interface defined as the iso-surface of some analytical function such as a Gaussian surface or the recently developed multi-resolution model of the molecular surface. The former is the isosurface of additive approximate electron density centered at each partial charge and the latter is the iso-surface of the solution to a diffusion equation. Because the surface is a particular level set of an analytical function, the solvent accessibility and the surface geometry can be readily derived from the function [70]. The molecular surface derived from the minimization of an energy functional can also be described by an isosurface hence it can be easily implemented into our interface algorithm.
(G)	am-	
I(A)	dyn	
tigrie	astic	
mul	tocha	
raic	t), st	
ulgeb	SOF	
3), 8	tion(	
H(MG	laxa	
tigric	er re	
mul	e ov	
3E),	essiv	
n(PE	succ	D).
atio	(j)	s(M
equ	nt(C	amio
nnar	adie	dyn
oltzn	e gr	ular
n-Bo	ugat	olec
oisso	conj	I), n
1: P	3S),	(MN)
atior	lel(C	nics
Nota	-Seic	echa
SIS.	auss	ar m
solve	G.	ecula
ann	E(FE	mol
ltzm	men	2M),
1-Bo	e ele	ics(C
issol	finit	chan
r Po	í,	mec
Iajo	ce(F	tum
1: V	eren	quan
e 4.	diff.	D), (
Tabl	inite	cs(S.

Software package	Description	URL	Availability
APBS	Solves PBE in parallel with FD MG and FE AMG solvers	http://agave.wustl.edu/apbs/	Windows, All Unix, Free, Open Source
DelPhi	Solves PBE sequentially with highly optimized FD GS solver	http://trantor.bioc.columbia.edu/delphi/	SGI, LINUX, AIX, \$250 aca- demic
GRASP	Visualization program with emphasis on graphics; of- fers sequential calculation of qualitative PB poten- tials	http://trantor.bioc.columbia.edu/grasp	SGI, \$500 academic
MEAD	Solves PBE sequentially with a FD SOR solver	http://www.scripps.edu/bashford	Windows, All UNIX, Free, Open source
UHBD	Multi-purpose program with emphasis on SD; offers sequential FD SOR PBE solver	http://mccammon.ucsd.edu/uhbd.html	All UNIX,\$300 academic
MacroDox	Multi-purpose program with emphasis on SD; offers sequential FD SOR PBE solver	http:///prin.chem.tntech.edu/macrodox.html	SGI, Free, Open source
Jaguar	Multi-purpose program with emphasis on QM; offers sequential FE MG, SOR, and CG solvers	http://www.schrodinger.com/Products/jaguar.html	Most UNIX, commercial
CHARMM	multi-purpose program with emphasis on MD; of- fers sequential FD MG solver and can be linked with APBS	http://yuri.harvard.edu	All UNIX. \$600 academic

### Chapter 5

# Application of Interface Methods to the Poisson-Boltzmann Equation

In this chapter, the continuous conditions of the electrostatic potential and its flux are to be incorporated in the numerical solution of the Poisson-Boltzmann equation. We first present the implementation of MSMS surfaces in the MIB method. Our implementation can be generalized to other finite-difference-based Poisson-Boltzmann solvers. We will then solve the Poisson-Boltzmann equation with the MIB method formulated in Chapter 3. The results of the MIB method will be compared with those of the PBEQ and the APBS, with special concerns in the convergence rate, the accuracy of the electrostatic potential near the dielectric interface and the accuracy of the electrostatic solvation energy. In addition to the MIB method, the immersed interface method is reproduced, which not only has its own merit as an alternative interface method for the Poisson-Boltzmann equation, but also helps us to confirm our MIB results.

## 5.1 Implementation of molecular surface in MIB and IIM

The implicit enforcement of the interface conditions at the molecular surface in the MIB method necessitates an explicitly defined molecular surface such that its intersection points with the mesh lines can be calculated with a sufficient accuracy being consistent with the accuracy of the approximation to the Poisson-Boltzmann equation. We choose the MSMS program as our molecular surface generator since it is one of a few software packages that explicitly calculate both the molecular surface coordinates and associated normal directions. MSMS gives the triangulation of the molecular surface, which is a first-order approximation to the analytical molecular surface. A decision has to be made on whether the triangulation surface or the analytical surface is to be used. By using a triangulated surface it means that the intersections of the surface triangles with the mesh lines will be calculated and used in the MIB formulation. These intersections deviate from the exact intersections of the molecular surface and the mesh lines by some small displacements whose magnitude depend on the density of the triangulation. By using these approximate intersect points instead of the exact intersect point one could get simplification and computational efficiency in locating points but at the possible reduction of accuracy in the solution of electrostatic potential. In the numerical experiments comparison will be made between these two types of molecular surfaces: the exact molecular surface and its approximation via triangulation. We will show that a fairly good approximation to the analytical molecular surface can be given by the MSMS surface and the error between these two surfaces can be reduced effectively by using appropriate triangulation density such that the numerical error due to the finite difference becomes dominant.

Starting with an MSMS triangulation, we are going to determine the accessibility of grid points to the solvent and the nearby mesh line/surface intersect point(s) for each irregular point. We will first calculate the possible intersect points of each triangle with the mesh lines. Let  $(x_i, y_i, z_i)$ , i = 1, ..., 3 be the three vertices of a surface triangle. We define

$$\begin{aligned} x_{min} &= \min\{x_1, x_2, x_3\}, & x_{max} &= \max\{x_1, x_2, x_3\}, \\ y_{min} &= \min\{y_1, y_2, x_3\}, & y_{max} &= \max\{y_1, y_2, y_3\}, \\ z_{min} &= \min\{z_1, z_2, z_3\}, & z_{max} &= \max\{z_1, z_2, z_3\}. \end{aligned}$$

The x-mesh lines  $(y_j, z_k)$  that might intersect with this triangle must lay inside the rectangles defined by above intervals for y and z, i.e.,  $y_{min} < y_j < y_{max}$  and  $z_{min} < z_k < z_{max}$ . The other two rectangles, determined by  $x_{min} < x < x_{max}$ ,  $y_{min} < y < y_{max}$  and  $x_{min} < x < x_{max}$ ,  $z_{min} < z < z_{max}$ , define respectively the z- and y- mesh lines that might intersect with the current triangle. The intersect points of mesh lines with the current triangle are calculated in two steps. In the first step, we calculate all the intersections of these mesh lines with the plane spanned by these three vertices

To find the intersect point of the mesh line  $(x_i, y_j)$  with this plane, for example, one can solve the linear equation of z obtained by replacing x, y with  $x_i, y_j$  respectively in equation (5.1), i.e.,

ŧ

$$\begin{vmatrix} x_i - x_1 & y_j - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0.$$
 (5.2)

The intersect point then has a coordinate  $(x_i, y_j, z)$ . In the second step, we examine whether these intersections are inside the triangle. Those points that are not located within the triangle must be discarded. The remaining points are the real intersect points of the mesh lines with this triangle. An intersect point is inside the triangle if



**Figure 5.1:** The illustration of mesh lines and triangle as well as their intersect points. If the intersect point happens to be a vertex, or passes through a side of the triangle, or the sum of three angles  $\theta_1, \theta_2$  and  $\theta_3$  is  $2\pi$ , then the intersect point is inside the triangle.

- the intersect point is overlapped with one of three vertices of the triangle,
- or the intersect point is located one side of the triangle,
- or the total of the three angles  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  is  $2\pi$ .

The normal vector of the triangle will be taken as the normal vector of the molecular surface at the intersect point. However, if the intersect point coincides with a vertex of the triangle, the normal vector at that vertex, which has been calculated by the MSMS, rather than that of the triangle will be taken as the normal vector of the molecular surface at the intersect point.

We will then determine the accessibility of each grid points based on all these intersect points of the mesh lines and the molecular surface. As we can always choose a sufficiently large computational domain to enclose the molecule, the grid points before the first intersect point or after the last intersect point are then always accessible and are marked by -1. Since a molecule is an enclosed simply connected or multiply connected 3-D domain, the number of intersect points of a mesh line with the molecular surface is alway even if the mesh line is never tangent to the molecular surface. This reflects the fact with any mesh line that penetrates into the molecular surface at some place must penetrate out of the surface again at the other place. If a mesh line is tangent to the molecular surface, the tangent point will be counted twice to make the total number of intersect points be even. Once the lists of intersect points on each mesh line are ready, all the intersect points on the same mesh line are ordered according to their major coordinate to make the determination of accessibility of grid points more efficient. Here the major coordinate means the x coordinate for intersect points on the same x-mesh line, the y coordinate for intersect points on the same y-mesh line and the z coordinate for intersect points on the same z-mesh line. The accessibility is determined on each mesh line. For example, to determine the accessibility of all grid points on mesh line  $(y_j, z_k)$ , all the intersect points of this mesh line with the molecular surface are increasingly ordered in an array, xint, of length n, which must be even. All the grid points  $(x_i, y_j, z_k)$  within  $xint_l$  and  $xint_{l+1}$  where l is odd will be regarded as inaccessible, while all the grid points within  $xint_l$  and  $xint_{l+1}$  where l is even are accessible, as shown in Figure 5.2. The accessibility at all grid points can be determined by examining all mesh lines along a single direction, i.e., either all x-mesh lines, or all y-mesh lines or all z-mesh lines. The irregular points can be found once the accessibility at all grid points are determined.

**Example 5.1.1.** To validate the MSMS surface implementation in our MIB method, we design a 3-D Poisson problem  $(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f(x, y, z)$  with internal interface defined by the molecular surface of cyclohexane(5.1). The discontinuous solution u(x, y, z) of the Poisson equation is designated to be

$$u(x, y, z) = \begin{cases} x^2 + y^2 + z^2 & \text{inside interface} \\ \frac{1}{\sqrt{x^2 + y^2 + z^2}} + \log(2\sqrt{x^2 + y^2 + z^2}) & \text{otherwise} \end{cases}$$
(5.3)

and the diffusion coefficient  $\beta = 1$  inside the cyclohexane and 80 outside. These definitions also prescribe f(x, y, z) as well as the jumps [u] and  $[\beta u_n]$ .

A second-order convergence is expected if the implementation of the MSMS surface is appropriate and the error in the MSMS surface is dominated by the intrinsic discretization error in the MIB method. The latter can be guaranteed by a triangulation of sufficiently high density. This prediction is verified by Table 5.1. More



Figure 5.2: 2-D diagram of the accessibility determined by chasing intersec points of mesh lines and molecular surface. Accessible grid points are marked by solid red circles and the inaccessible grid points are marked by empty black circles. For the upper mesh line  $(y = y_4)$  which has 6 intersection points with the molecular surface (the right most one is counted twice as the mesh line is tangent to the molecular surface at this point), the grid points (1.4) through (3.4) are accessible because they are located before the first intersection point; grid points (4.4) through (5.4) are inaccessible because they are located between the first and the second intersection point; grid points (6.4) through (8.4) are accessible again because they are located between the second and the third intersection point; grid points (9.4) and (10.4) are inaccessible again because they are located between the third and the fourth intersection point; grid points (1.14) through (15.4) are accessible secause they are located between the fourth and the fifth intersection point. Between the fifth and the sixth intersection points there are no grid points. After the sixth intersection point which is the last one on this mesh line, all the grid points are accessible.

importantly, Table 5.1 provides a guide line for the selection of surface triangulation. In particular, in order to get a second-order convergence up to a mesh size of 0.1Å, the density should be larger than 50. On the other hand, if one sticks to a mesh size of 0.5Å in the calculation of electrostatic potential for the real biomolecules, a triangulation density of 10 is sufficient since for this mesh size there is no essential difference between the solutions at density of 10 and density of 40. Assured by this observation, an MSMS surface with a density of 10 will be used for numerical solutions of the Poisson-Boltzmann equation with a mesh size of 0.5Å. When a finer mesh such as 0.2Å or 0.1Å is necessary for the examination of the convergence, we

Table 5.1: Positions	and radi	i of atoms	s of cyclol	nexane.
	x	y	z	vdW radius
	-2.0270	0.9540	-0.6510	1.7
	-1.6690	0.2340	0.6650	1.7
	-0.4530	-0.6870	0.4410	1.7
	0.7510	0.1480	-0.0400	1.7
	0.3930	0.8680	-1.3560	1.7
	-0.8230	1.7880	-1.1320	1.7
	-2.2840	0.2080	-1.4180	1.2
	-2.8880	1.6170	-0.4830	1.2
	-2.5270	-0.3680	0.9970	1.2
	-1.4260	0.9800	1.4350	1.2
	-0.1960	-1.1890	1.3850	1.2
	-0.7010	-1.4410	-0.3200	1.2
	1.0070	0.8940	0.7270	1.2
	1.6120	-0.5150	-0.2080	1.2
	0.1490	0.1210	<b>-2</b> .1260	1.2
	1.2510	1.4700	-1.6880	1.2
	-1.0810	2.2910	-2.0760	1.2
	-0.5750	2.5430	-0.3710	1.2

will instead use the MSMS surface with a density of 40. Our implementation of the

 Table 5.2: Numerical convergence test for Example (5.1.1).

0		I (*	/
Density of	Mesh	$L_{\infty}$	Order
triangulation	size(Å)	error	Order
	1/2	1.03(0)	
5	1/4	9.83(-1)	
	1/8	1.16(0)	
	1/2	1.38(-1)	
10	1/4	3.63(-2)	1.93
	1/8	1.65(-2)	1.14
	1/2	1.37(-1)	
20	1/4	3.54(-2)	1.95
	1/8	9.83(-3)	1.85
	1/2	1.38(-1)	
40	1/4	3.54(-2)	1.96
	1/8	9.07(-3)	1.96

MSMS surface for the MIB method is also applicable for the IIM method, i.e., we first find the intersect points of each mesh line with the interface and then determine the accessibility of each grid point. It is noticed, however, the IIM scheme at any irregular point only requires the geometry of a nearby interface point, which is not necessary to be the intersect point of the mesh line and the molecular surface. We choose one of the nearby intersect points as the required interface point since it produces minimum extra calculation.



Figure 5.3: The grid points and interface point needed for the formulation of the IIM scheme [13] at the irregular point(i, j, k). The IIM method manages to find the Taylor expansions of the solution accurate to  $O(h^3)$  at both sides of the interface, for which a total of 16 expansion coefficients are to be determined. Original IIM requires 10 grid points and 6 interface conditions at an interface point(p' or p) to solve for the 16 polynomial coefficients. These 10 points(red) are chosen to be (i, j, k), other 6 grid points directly connecting to (i, j, k) and 3 more grid points inside the cube, preferably at the other side of the interface. IIM with optimization employs all the 27 grid points (red and green) inside the cube to produce constrained quadratic optimization problems from which 16 polynomial coefficients are solved. See [13] for the detailed formulation of 3-D IIM scheme. It is not necessary to choose the interface point to be the nearest one as p'. For simplicity we choose p', the intersect point of the interface with the mesh line, as the interface point for the regular point (i, j, k).

#### 5.2 Numerical validation of interface method for the Poisson-Boltzmann equation

We consider a sphere of radius 2Å, centered at the origin and with charges inside whose number and positions are to be tuned. A sphere with single or multiple charges admits an analytical solution of electrostatic potential and has been used for evaluation and accuracy assessment for various Poisson-Boltzmann solvers. For noncentered charge or multiple charges one can seek the analytical solution as the expansion of spherical harmonic functions [39]. Both the exact boundary conditions and the exact values of the solvation energy can be calculated from these analytical solutions. The derivation of analytical solutions for multiple charges is given in Appendix A.

The first computation is on the sphere with a single unit charge. Table 5.3 lists the convergence of two interface methods in comparison with the PBEQ and the APBS. In addition to the electrostatic solvation energy, we also calculate the maximum absolute error e1, the maximum percentage error e2 and the average percentage error e3 of the electrostatic potential near the surface of the sphere, as

$$e_1 = \max |\phi(x, y, z) - \tilde{\phi}(x, y, z)|,$$
 (5.4)

$$e_2 = 100 \times \max \left| \frac{\phi(x, y, z) - \phi(x, y, z)}{\phi(x, y, z)} \right|,$$
 (5.5)

$$e_3 = 100 \times \sum^{N_{irr}} |\frac{\phi(x, y, z) - \tilde{\phi}(x, y, z)}{\phi(x, y, z)}| / N_{irr}.$$
(5.6)

Here  $\tilde{\phi}(x, y, z)$  and  $\phi(x, y, z)$  are the computed and exact electrostatic potential, respectively. The maximum values and summation are taken over all the irregular grid points near the molecular surface, whose number is given by  $N_{irr}$ . Given a sufficiently small grid spacing, a convergent solvation energy could be solved with all three methods. The necessary resolution required for a given accuracy of solvation energy, however, differs across the methods due to their different convergence properties. MIB and IIM have a convergence rate of 2, which is indicated by about fourfold decreasing in all three errors as the grid spacing is halved. Such a uniform convergence is attributed to the modified difference scheme near the interface. In contrast, PBEQ and APBS have similar convergence rates of around 0.3. As a result, a grid spacing of 0.5Å suffices for both MIB and IIM to yield an almost exact solvation energy whereas for PBEQ and APBS one has to use a mesh size smaller than 0.05Å to get a solvation energy of similar accuracy. The most significant difference occurs at the accuracy of the surface potential, which can be seen from the maximum absolute error, the maximum relative errors and the average relative errors in Table 5.3. For mesh size of 0.5Å, the surface potential of PBEQ and APBS deviate from the exact value by more than 84% at its maximum and by more than 23% on average. The errors in the surface potential calculated with PBEQ or APBS are consistently much larger than those of the interface method despite a slight decrease with the refinement of the mesh. The maximum error is around 50% even for a grid spacing of 0.05Å. With either interface method, however, the surface potential is accurate to about 8% at its maximum and only about 2% on average for a grid spacing of 0.5Å. If h = 0.2Å is chosen the maximum error is only about 2% with corresponding average error below 1%, which is almost unachievable by PBEQ and APBS due to their slow convergence.

The second system we studied is a sphere with a noncentered charge located at (a, 0, 0). As the charge approaches the surface, a stronger interaction between the charge and the surface is anticipated and a larger error may occur if the two interface conditions are not handled correctly. The electrostatic solvation energies for  $0 \le a \le 1.5$  were computed and collected in Table 5.4-5.5. In addition to the growth of the solvation energy these two tables also show that the computation errors, both for PBEQ and the two interface methods, increase as the charge approaches the surface. It is interesting to see that interface methods and PBEQ have different responses to the displacement of the charge. As the charge is moving from the center to (0.2, 0, 0)the relative errors, both the maximum and the average, increase about twofold for the MIB method whereas they remain in the same high level for PBEQ. A slight increase in the maximum relative error can also be seen for the IIM method. A larger error in the surface potential of both interface methods occurs as the displacement of charge increases but the average relative error is well below 5% except for a = 1.5. For PBEQ, although there is a slight decrease in the accuracy of surface potential, the maximum and average relative errors are always larger than 20% for the various displacements of charge. Furthermore, the average relative error in the interface methods can be efficiently reduced to less than 1% for all displacements by refining the mesh to a grid spacing of 0.2Å but the large error in the traditional finite difference method PBEQ decreases only slightly from 23% to 12%. Table 5.5 also shows that a very large error could occur on interface methods as the charge is very close to surface, such as a = 1.0, 1.2 or 1.5. Such a large error results from the mixing of the grid points for singular charge distribution and for the modified difference schemes near the interface. In particular, the solution on the grid points within the trilinear distribution of the singular charge is of the lowest accuracy. The inclusion of these grid points into the difference scheme at the irregular points, as in the case of a > 0.6 and h = 0.5, will degrade the accuracy of the solution near the interface. However, because the van der Walls radii of the partially charged atoms are usually larger than 1.0Å such a small distance between the charge and the surface will not occur in pratical biomolecular simulations.

A stronger interaction of charge and surface can be found by placing two noncentered charges in a single sphere. In the third system for our validation, one charge is placed at (a, 0, 0) and the other at (0, 0, a) where a varies from 0.2Å to 1.5Å. Except for the increase of the solvation energy, the results for both interface methods and PBEQ display a striking qualitative similarity with the case with a single noncentered charge, as presented in Table 5.6-5.7. The amplification of the charge-surface interaction results in a larger maximum relative error in the surface potential of interface methods, which could be 34% for a = 0.6Å. However, the average relative error does not grow with the increase of the number of charges, and is smaller than 3% for a < 1.0Å, similar to the case with single charge.

For all three systems, PBEQ provides a more negative (i.e. more favorable) free energy of solvation, and the deviation from the analytical value is around 5% for a mesh size of 0.5Å. Such a systematic percentage error amounts to about 3.87kcal/molfor a centered-charged single sphere and will lead to about a deviation of 17kcal/molfor sphere with two noncentered charges (a = 0.6).

These three numerical experiments also demonstrate some noticeable differences between the two interface methods. Compared with the MIB method, the smaller interface discretization stencil of IIM indicates that its interface discretization will less likely be influenced by the singular charge distribution. This speculation is validated by the smaller  $e_1$  and  $e_2$  of IIM in Tables 5.3 - 5.7 at a mesh size of 0.5Å if the displacement of the charge is smaller then 0.6Å. In particular, the maximum absolute error in the surface potential is 2.63kcal/mol/e for MIB and is about 1.98kcal/mol/efor IIM if the displacement of charge is zero. Substantial increases of the maximum absolute error and the maximum relative error are seen in the MIB method. Such increases for IIM are nevertheless much smaller.

This relative advantage of smaller discretization stencils in IIM will diminish when the charge is close to the interface. If that is the case, the grid points participating in the distribution of singular charges have to be involved in the discretization at irregular points and these two types of errors are always mixed together. A considerable increase in the maximum absolute error is seen in Table 5.3 and Table 5.6 when the displacement is increased from 0.6Å to 0.8Å or larger.

On the other hand, MIB has a more accurate approximation to the interface conditions and the resultant discretization at the irregular points as illustrated in Table 2.1 and Table 2.2. This explains its smaller average relative errors for all the tested cases in the above numerical experiments, despite its larger maximum absolute errors and large relative errors at small charge displacement. Furthermore, the errors in the solvation energies computed with IIM are much larger than those computed with MIB, especially at large charge displacement. For this reason we will only report the results of MIB in all the other numerical experiments and calculations for biomolecules.

For general molecules, however, the evaluation of the solutions of the variety of PB solvers remains difficult due to the lack of analytical solutions for reference. The slow convergence or stagnance of many Poisson-Boltzmann solvers, as found in practical simulations, prevents the achievement of a convergent solution through simple mesh refinement for self-checking. In this numerical experiment, we are going to investigate the convergence of PBEQ and MIB on the problems without analytical solutions. In

particular, we are interested in whether the numerical results provided by the MIB method can be used as an alternative reference to measure the accuracy of popular PB solvers.

Two unit positively charged spheres, of the same radius  $2\text{\AA}$ , are placed on the *x*-axis symmetrically with respect to the origin. The molecular surface of this model molecule is composed of two spherical caps at two ends and an inward tori surface in the middle, each of which is analytically given. The solvation energy is computed for systems with different sphere-sphere distances, which vary from  $2\text{\AA}$  to  $5\text{\AA}$ . The electrostatic potential computed from MIB at the finest mesh of  $h=0.05\text{\AA}$  is saved as reference to measure the accuracy of the surface potential computed with MIB and PBEQ at three coarse meshes of h = 0.5, 0.2, 0.1.

The data given in Table 5.8 indicate that the solvation energy is almost convergent for MIB with a grid spacing of 0.5Å and the further refinement of the mesh can only provide minor improvement. For PBEQ, a deviation of 4-6kcal/mol in solvation energy at the coarse mesh can be reduced by mesh refinement. Hence the solution of PBEQ essentially converges to that of MIB as far as the solvation energy is concerned. The convergence rate of PBEQ's solution to MIB's solution matches well with the rate observed in the last three systems with analytical solutions as reference. For all the charge distance considered, the maximum and average relative errors of the surface are about more than 80% and 20%, respectively for a grid spacing of 0.5Å, which are of the same magnitude as observed in the first three test systems, see Figure 5.4. The average absolute error in the surface potential at this resolution is about 3-4kcal/mol whereas the largest error could be 17-22kcal/mol.

#### 5.3 Application of interface method for biomolecules

Motivated by the superior accuracy and convergence property of MIB, we then move to the numerical experiments of MIB on biomolecules. We are going to investigate if traditional Poisson-Boltzmann solvers will converge to the solution of MIB at coarse mesh for large macromolecules and if we can achieve the savings in the computational



Figure 5.4: Maximum absolute error  $e_1$  (upper left), Maximum relative error  $e_2$ (upper right), average absolute errors(lower left) and average relative error  $e_3$  (lower right) in the surface potential of two positively charged spheres. The errors of MIB and PBEQ are plotted in black and blue, respectively. The average absolute error is the total of the absolute error at all irregular points divided by the number of irregular points. The absolute error is in kcal/mol/e and the relative error is the percentage. The x-axis is the distance between centers of two spheres. Circle: h = 0.5; Diamond: h = 0.2; Square: h = 0.1.

time and the memory due to the use of coarse mesh. Twenty four structures, mostly adopted from test set 1 of [19], were used in this study to examine the accuracy and robustness of MIB in handling different protein conformations. The missing hydrogen atoms in the original PDB files were added using the HBUILD function in CHARMM. The all-atom representations were generated and partial charges were assigned, both with the CHARMM22 force field. The coordinates and partial charges of the atoms prescribed as such were also supplied to APBS. It is noticed that hydrogen atoms in some of the amino groups are completely buried inside the nitrogen atom and MSMS can not handle such cases. These hydrogen atoms then have to be removed before being supplied to MSMS as they have no contribution to the molecular surface. The partial charges at these hydrogen atoms, however, have to be retained in the solution of the Poisson-Boltzmann equation as they do contribute to the electrostatic potential. The deviation of electrostatic solvation energies calculated by PBEQ with different mesh size from those calculated by MIB with a fixed mesh size of 0.5Å are plotted in Figure 5.5. For all 24 proteins, PBEQ delivers a more negative solvation energy at a mesh size of 0.5Å or 0.25Å. At a mesh size of 0.15Å, nevertheless, there are two proteins for which the PBEQ gives a slightly larger solvation energy. For protein 1ajj, the solvation energy solved from PBEQ is larger than that from MIB by 0.7kcal/mol or 0.06%. For protein 1uxc, this difference amounts to 1.5kcal/molor 0.13%. The overall monotonous decreasing of the deviation from the MIB with the improvement of resolution proves the reliability of MIB at low resolution as the reference for traditional Poisson-Boltzmann solvers. The saving of computational time and memory usage are critical for simulations of large molecules, for which it is usually impossible to use a resolution of 0.2Å or smaller to obtain a convergent solution. In particular, there is a strong correlation between this deviation and the total partial charge of the biological molecules. For protein 1a7m with 180 residues, for example, the total charge is about 677e and the deviation is about -103.5, -36.5and -12.5kcal/mol/e for h = 0.5, 0.25 and 0.15, respectively. A much larger error of solvation energy can be projected for molecules of large size or highly charged, and might contribute to the over stabilization of slat-bridges as found in many simulations [55, 95]. This correlation also explains the aforementioned overshoot for the small proteins 1ajj and 1uxc. For larger proteins with large total absolute charges, the solvation energies computed from PBEQ at a mesh size of 0.15Å is always smaller than those from MIB by more than 10kcal/mol.

The surface electrostatic potential of 1a63 generated with PBEQ is shown in Figure 5.6, along with the potential difference between the PBEQ solution and the MIB solution. There are scattered regions of negative potential but most of the potential surface is largely positive where the numerical error is also large, as seen in the right



Figure 5.5: Left y-axis: the deviation of electrostatic solvation energy calculated with PBEQ at h = 0.5 (circle), h = 0.25(diamond) and h = 0.2(square) from that calculated with MIB at h = 0.5. Right y-axis: the solid line plots the total absolute charge of the proteins. x-axis from left to right: 1ajj, 1vii, 2erl, 1bbl, 1cbn, 2pde, 1sh1, 1fca, 1ptq, 1bor, 1uxc, 1vjw, 1fxd, 1hpt, 1mbg, 1bpi, 1r69, 451c, 1neq, 1a2s, 1svr, 1frd, 1a63 and 1a7m.

chart. These errors in the surface potential, usually as large as -3kcal/mol/e to 3kcal/mol/e, may contribute to large discrepancies in calculating the binding affinity and reaction rate of protein-ligand association. More detailed work is required to elucidate and justify the role of this discrepancy.

		5	2													
	- - -								Er	rors in s	urface p	otential				
Ч		Þ	Q			MIB			IIM			PBEQ			APBS	
	MIB	IIM	PBEQ	APBS	e1	$e_2$	с <u>э</u>	$e_1$	$e_2$	e3	e1	e2	e3	e1	e2	e3
0.50	-81.97	-82.12	-85.78	-85.85	2.63	8.54	1.65	1.98	7.07	2.46	17.05	84.26	23.31	17.06	84.26	23.31
0.20	-81.98	-82.00	-82.84	-82.58	0.30	1.55	0.36	0.23	1.96	1.19	7.51	74.44	14.08	7.50	74.43	12.54
0.10	-81.98	-81.99	-82.49	-82.27	0.04	0.37	0.09	0.03	0.48	0.31	3.84	62.30	12.52	3.83	62.30	10.55
0.05	-81.98	-81.98	-82.20	-82.03	0.01	0.13	0.01	0.01	0.12	0.06	1.94	46.95	8.49	1.89	46.18	6.38

ntered unit charge.	
phere with a ce	
potential for a s	
in the surface	
ol and the error	
$\Delta G$ in kcal/m	mol.
Electrostatic solvation energy	olvation energy is -81.98 $kcal/n$
Table 5.3:	The exact s

									Errors in	surface	potenti	lal		
$h$ $\Delta G$	$\nabla C$	$\Delta G$	5				NIB			IIM			PBEQ	
Exact MIB IIM	Exact MIB IIM	MIB IIM	IIM		PBEQ	eı	$e_2$	e3	e1	$e_2$	e3	e1	$e_2$	e3
0.5 -82.84 -82.8	-82.84 -82.8	-82.84 -82.8	-82.8	2	-86.84	7.33	17.39	2.26	6.00	10.62	2.58	21.63	84.88	24.73
0.2 -82.803 -82.80 -82.65	-82.803 -82.80 -82.65	-82.80 -82.65	-82.65		-83.70	0.45	1.76	0.37	0.35	1.82	0.85	9.97	76.38	15.21
0.1 -82.80 -82.64	-82.80 -82.64	-82.80 -82.64	-82.64		-83.32	0.06	0.43	0.09	0.05	0.45	0.22	5.15	65.19	12.98
0.5 -85.46 -85.02	-85.46 -85.02	-85.46 -85.02	-85.02		-89.83	6.13	19.94	2.19	4.00	9.90	2.55	27.63	86.45	24.51
0.2 -85.375 -85.37 -84.71	-85.375 -85.37 -84.71	-85.37 -84.71	-84.71		-86.80	0.71	2.03	0.38	0.56	2.34	0.84	13.31	78.13	15.01
0.1 -85.40 -84.70	-85.40 -84.70	-85.40 -84.70	-84.70		-86.09	0.10	0.74	0.27	0.09	0.59	0.21	6.99	67.91	12.77
0.5 -90.38 -89.38	-90.38 -89.38	-90.38 -89.38	-89.38		-95.20	7.17	31.96	2.35	6.55	13.64	2.87	33.86	86.23	24.23
0.2 -90.036 -90.04 -88.48	-90.036 -90.04 -88.48	-90.04 -88.48	-88.48		-91.18	1.17	2.39	0.40	0.93	3.13	0.83	17.91	79.68	14.67
0.1 -90.04 -88.46	-90.04 -88.46	-90.04 -88.46	-88.46		-90.70	0.16	0.64	0.18	0.12	0.78	0.21	9.46	70.50	12.40
0.5 -98.59 -97.00	-98.59 -97.00	-98.59 -97.00	-97.00		-103.89	10.02	45.72	2.85	14.37	17.42	3.56	40.53	86.78	23.86
0.2 -97.493 -97.50 -94.59	-97.493 -97.50 -94.59	-97.50 -94.59	-94.59		-98.91	2.01	2.85	0.42	1.62	4.13	0.83	24.26	81.03	14.15
0.1 -97.49 -94.55	-97.49 -94.55	-97.49 -94.55	-94.55		-98.30	0.32	2.26	0.70	0.23	0.11	0.22	13.30	72.95	11.88

**Table 5.4:** Electrostatic solvation energy  $\Delta G$  in kcal/mol and error in the surface potential for a sphere with a single noncentered charge.

argc.
sd ch
ntere
once
ıgle r
a sir
with
here
: a sp
al fo
tenti
cc bo
surfa
the
ror in
nd eri
tol a
al/n
in k
ν ΔG
nergy
ion e
olvat
atic s
trost
Elec
5.5:
able
H

									and the second se					
									Errors in	surface	potentia			
a	ч		Ā	5			MIB			IIM			PBEQ	
		Exact	MIB	IIM	PBEQ	e1	e2	e3	e1	$e_2$	e3	e1	$e_2$	e3
	0.5		-109.90	-107.49	-115.82	18.54	45.21	3.05	25.41	22.86	3.97	46.14	86.79	23.13
1.0	0.2	-109.121	-109.17	-104.31	-110.94	3.55	3.32	0.46	2.91	5.25	0.84	32.86	82.14	13.47
	0.1		-109.13	-104.21	-110.17	0.58	1.04	0.11	0.47	1.55	0.22	19.58	75.25	11.17
	0.5		-135.94	-135.12	-139.60	31.17	96.81	5.35	52.96	37.93	7.03	48.08	86.39	22.35
1.2	0.2	-127.766	-127.95	-120.35	-130.04	5.82	5.95	0.51	4.84	6.10	0.89	43.59	82.84	12.54
	0.1		-127.81	-120.12	-129.14	1.34	1.48	0.14	1.13	2.36	0.25	28.84	77.29	10.21
	0.5		-198.51	-201.52	-197.96	18.41	565.1	16.61	50.22	490.5	19.16	37.66	82.09	20.16
1.5	0.2	-186.554	-191.17	-176.47	-191.27	24.00	37.10	0.82	16.34	15.62	1.33	56.56	82.19	10.40
	0.1		-186.85	-172.74	-187.90	6.81	2.80	0.75	6.26	4.21	0.33	50.19	79.10	8.16

**Table 5.6:** Electrostatic solvation energy  $\Delta G$  in kcal/mol and the error in the surface potential for a sphere with two noncentered charges.

									Errors in	surface	potenti	ial		
a	ų		Ą	5			MIB			IIM			PBEQ	
		Exact	MIB	IIM	PBEQ	e1	$e_2$	e3	e1	e2	e3	e1	$e_2$	e3
	0.5		-329.55	-329.89	-345.07	10.20	12.05	1.83	7.90	8.22	2.48	41.46	84.81	24.67
0.2	0.2	-329.558	-329.51	-329.27	-333.06	0.78	1.73	0.36	09.0	1.76	0.85	18.05	75.79	15.25
	0.1		-329.54	-329.23	-331.60	0.10	0.42	0.11	0.07	0.45	0.22	9.14	64.16	13.02
	0.5		-334.64	-334.07	-350.85	11.62	13.60	1.75	8.35	8.87	2.42	49.43	85.34	24.56
0.4	0.2	-334.579	-334.53	-333.28	-338.25	1.01	1.87	0.36	0.77	2.08	0.85	21.59	76.98	15.14
	0.1		-334.57	-333.24	-336.72	0.11	0.64	0.21	0.09	0.50	0.22	10.94	65.85	12.90
	0.5		-343.94	-342.47	-360.82	12.67	21.72	2.02	7.29	11.39	2.88	57.34	85.61	24.43
0.6	0.2	-343.378	-343.37	-340.34	-347.35	1.38	2.08	0.36	1.06	2.71	0.84	25.95	78.11	14.94
	0.1		-343.38	-340.27	-345.68	0.18	0.51	0.09	0.14	0.65	0.22	13.34	67.75	12.71
	0.5		-358.87	-356.67	-376.24	10.35	32.56	2.47	14.74	14.98	3.64	64.08	85.52	24.13
0.8	0.2	-356.904	-356.93	-351.28	-361.35	2.12	2.48	0.38	1.67	3.61	0.86	31.63	79.24	14.63
	0.1		-356.95	-351.15	-359.47	0.30	0.66	0.09	0.23	0.93	0.23	16.87	69.91	12.38

								니	rrors in a	surtace l	potential			
a	Ч		Ā	55			MIB			MII			PBEQ	
		Exact	MIB	MII	PBEQ	$e_1$	$e_2$	e3	e1	$e_2$	e3	e1	e2	e3
	0.5		-379.09	-375.70	-397.36	19.06	34.59	3.01	26.07	17.86	4.47	67.92	85.22	23.62
1.0	0.2	-377.347	-377.49	-368.14	-382.47	3.58	3.02	0.44	2.89	4.70	0.92	39.19	80.33	14.15
	0.1		-377.38	-367.87	-380.30	0.58	0.94	0.10	0.23	0.93	0.23	22.32	72.44	11.89
	0.5		-426.16	-427.04	-437.68	31.71	79.56	6.14	54.69	32.17	8.61	66.99	84.68	23.07
1.2	0.2	-409.898	-410.30	-395.82	-415.66	5.80	4.83	0.53	4.78	5.60	1.03	48.68	81.33	13.42
	0.1		-409.98	-395.29	-413.34	1.34	1.40	0.13	1.13	2.19	0.30	31.00	75.12	11.15
	0.5		-540.35	-549.95	-541.39	19.47	113.65	11.03	51.97	87.62	15.52	50.09	80.17	21.12
1.5	0.2	-516.178	-525.41	-497.77	-526.10	23.96	33.78	0.00	16.38	14.12	1.66	59.60	81.07	11.54
	0.1		-516.80	-490.28	-519.27	6.81	2.76	0.31	6.27	4.06	0.39	51.47	77.78	9.30

2
5
ω
1
0
<u> </u>
0
~
2
Ψ.
5
8
7
-
S.
2
g
0
ā
0
- 5
2
÷
-
.=
≥
0
<u> </u>
6
- E
~
1
5
പ
F
.0
بتب
C.
به
g
Ċ
ته
0
ā
0
- 2
<u>چ</u>
цт,
5
50
•.
0
-
E.
E C
5
5
~~
E
сIJ
c cri
ne eri
the eri
the eri
d the eri
nd the en
und the eri
and the en
l and the en
ol and the en
vol and the en
mol and the en
/mol and the en
<i>l/mol</i> and the en
ul/mol and the eri
cal/mol and the en
scal/mol and the en
keal/mol and the en
n keal/mol and the en
in kcal/mol and the en
' in keal/mol and the eri
3 in keal/mol and the er
G in kcal/mol and the eri
$\Delta G$ in $kcal/mol$ and the err
$\Delta G$ in kcal/mol and the err
y $\Delta G$ in $keal/mol$ and the eri
$\operatorname{ty} \Delta G$ in $keal/mol$ and the eri
rgy $\Delta G$ in $kcal/mol$ and the eri
ergy $\Delta G$ in $keal/mol$ and the er
nergy $\Delta G$ in $keal/mol$ and the eri
mergy $\Delta G$ in $kcal/mol$ and the eri
energy $\Delta G$ in $kcal/mol$ and the eri
1 energy $\Delta G$ in $kcal/mol$ and the eri
on energy $\Delta G$ in $keal/mol$ and the eri
ion energy $\Delta G$ in $keal/mol$ and the eri
tion energy $\Delta G$ in $keal/mol$ and the eri
ation energy $\Delta G$ in $keal/mol$ and the eri
vation energy $\Delta G$ in $kcal/mol$ and the eri
lvation energy $\Delta G$ in $keal/mol$ and the eri
olvation energy $\Delta G$ in $kcal/mol$ and the eri
solvation energy $\Delta G$ in $keal/mol$ and the eri
: solvation energy $\Delta G$ in $kcal/mol$ and the eri
ic solvation energy $\Delta G$ in $kcal/mol$ and the eri
tic solvation energy $\Delta G$ in $kcal/mol$ and the eri
atic solvation energy $\Delta G$ in $keal/mol$ and the eri
tatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
static solvation energy $\Delta G$ in $keal/mol$ and the eri
ostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
rostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
trostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
ctrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
ectrostatic solvation energy $\Delta G$ in $keal/mol$ and the eri
Nectrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
<b>7</b> : Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
.7: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
<b>5.7:</b> Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
<b>5.7:</b> Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
e 5.7: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
<b>Je 5.7:</b> Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
ble 5.7: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the er
tible 5.7: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the eri
able 5.7: Electrostatic solvation energy $\Delta G$ in $kcal/mol$ and the error

**Table 5.8:** Electrostatic solvation energy in kcal/mol of two positively charged spheres at four different mesh sizes h = 0.5, 0.2, 0.1 and 0.05. The distance between the centers of two spheres D vary from 2Å to 5Å.

<u></u>	MIB					PBEQ					
D	0.5	0.2	0.1	0.05		0.5	0.2	0.1	0.05		
2.0	-276.39	-276.78	-276.83	-276.84		-283.12	-279.66	-277.70	-277.10		
<b>2.5</b>	-266.68	-267.15	-267.22	-267.24		-272.28	-269.05	-268.15	-267.50		
3.0	-257.69	-258.00	-258.06	-258.07		-266.02	-260.34	-259.10	-258.60		
<b>3.5</b>	-249.28	-249.39	-249.43	-249.44		-252.72	-250.82	-250.33	-249.44		
4.0	-241.28	-241.45	-241.49	-241.50		-247.92	-243.39	-242.40	-241.90		
4.5	-234.39	-234.38	-234.38	-234.38		-238.76	-235.73	-235.22	-234.80		
5.0	-228.10	-228.18	-228.16	-228.17		-235.70	-230.04	-229.10	-228.60		

able 5.9: Sum	mary of test p	roteins ar	nd the co	mputed	electrost	atic solv	ation ene	srgies with	h MIB an	id PBEC	Q at diffe	rent mesl	ı sizes.
PD	3 ID	lajj	1bbl	lbor	1bpi	lcbn	lfca	1 frd	1fxd	1hpt	1mbg	lneq	1ptq
Number	of atoms	519	576	832	898	648	729	1478	824	858	903	1187	795
MTD/0 E Å V	$\Delta G$	-1139.5	-986.6	-853.4	-1302.5	-304.4	-1202.5	-2852.4	-3351.1	-811.2	-1353.5	-1728.8	-871.8
	(CPU Time)	(42)	(59)	(67)	(81)	(49)	(45)	(102)	(51)	(59)	(99)	(147)	(26)
	$\Delta G$	-1170.7	-1028.4	-890.7	-1356.6	-327.6	-1244.3	-2941.9	-3302.8	-867.0	-1398.3	-1800.7	-915.0
r DEQ(U.3A)	(CPU Time)	(9)	(2)	(6)	(11)	(2)	(12)	(18)	(12)	(10)	(10)	(19)	(6)
DDEO(0 95 Å	$\nabla C$	-1145.7	-1000.5	-865.7	-1318.9	-310.8	-1220.4	-2884.0	-3323.5	-830.9	-1362.5	-1752.0	-881.4
r DEQ(0.23A	<sup>)</sup> (CPU Time)	(26)	(69)	(87)	(66)	(65)	(134)	(155)	(135)	(28)	(06)	(182)	(20)
	$\nabla G$	-1138.8	-993.0	-857.0	-1307.0	-307	-1213	-2867.0	-3312.0	-820.0	-1355.0	-1738.0	-877.0
r DEQ(U.13A	(CPU Time)	(397)	(517)	(580)	(695)	(466)	(1009)	(1003)	(040)	(588)	(594)	(1273)	(556)

sizes
mesh
rent
diffe
2 at
PBE(
and
AIB
h N
s wit
ergie
n en
vatio
c sol
stati
ectro
d el
pute
com
l the
s anc
otein
st pro
of tes
ary (
mmr
 ເ
0.0
le
<b>Lab</b>
L 7

	1a7m	2809	-2156.5	(378)	-2260.1	(55)	-2193.0	(462)	-2169.0	(2608)
	1a63	2065	-2372.0	(269)	-2496.1	(41)	-2416.0	(367)	-2394.0	(2092)
•	1a2s	1272	-1911.6	(155)	-1968.5	(21)	-1933.0	(138)	-1920.0	(943)
	451c	1216	-1026.3	(88)	-1084.2	(14)	-1045.0	(108)	-1033.0	(735)
	2pde	667	-820.6	(43)	-849.3	(2)	-832.2	( 64)	-824.0	(450)
5	2erl	573	-949.6	(34)	-980.5	(9)	-959.1	(49)	-953.0	(365)
	lvjw	828	-1243.9	(52)	-1291.5	(12)	-1256.9	(123)	-1247.0	(865)
	lvii	596	-901.5	(42)	-936.8	(9)	-914.1	(53)	-905.0	(383)
	luxc	809	-1145.5	(02)	-1192.6	(6)	-1154.1	(73)	-1144.0	(521)
-	lsvr	1435	-1689.1	(135)	-1784.9	(18)	-1740.0	(163)	-1712.0	(1115)
	1sh1	702	-753.4	(48)	-803.2	( <u>-</u> )	-772.2	(58)	-764.0	(433)
	1r69	266	-1088.1	(84)	-1142.7	(11)	-1103.3	(83)	-1093.0	(1210)
6	IJ	of atoms	$\Delta G$	(CPU Time)	$\Delta G$	(CPU Time)	$\Delta G$	(CPU Time)	$\Delta G$	(CPU Time)
	PDB	Number o	NTD/O E Å V	(NGU) dIIM	DECIDEAD	LDEQ(U.3A)	DEC/0 or \$	LBEQ(U.Z3A)		LDEQ(U.10A)

Table 5.10: Summary of test proteins and the computed electrostatic solvation energies with MIB and PBEQ at differen	nt mesh sizes.
Cable 5.10:         Summary of test proteins and the computed electrostatic solvation energies with MIB and PBEQ	at differen
Table 5.10: Summary of test proteins and the computed electrostatic solvation energies with MIB	and PBEQ
Table 5.10: Summary of test proteins and the computed electrostatic solvation energies with	MIB
Table 5.10:         Summary of test proteins and the computed electrostatic solvation energie	s with
Table 5.10:         Summary of test proteins and the computed electrostatic solvation	ı energie
Table 5.10:         Summary of test proteins and the computed electrostatic	solvation
Table 5.10:         Summary of test proteins and the computed electros	static s
Table 5.10: Summary of test proteins and the computed	electros
Table 5.10: Summary of test proteins and the e	computed
Table 5.10: Summary of test proteins and	d the e
Table 5.10:         Summary of test	proteins an
Table 5.10: Summary	of test
<b>Fable 5.10:</b>	Summary
	<b>Table 5.10:</b>



Figure 5.6: The map of electrostatic potential on the surface of protein 1a63. Upper: potential computed with MIB; Lower: Difference of potentials computed with PBEQ and with MIB.

### Chapter 6

# Thesis Contribution and Future Research Plan

#### 6.1 Thesis Contribution

#### 6.1.1 The Matched Interface and Boundary Method

A new approach for solving elliptic interface problems, the matched interface and boundary (MIB) method, is generalized from its primitive 1-D formulation to higher dimensions in this dissertation. The MIB method, after generalization, is capable to solve 2-D or 3-D elliptic problems with discontinuous diffusion coefficients or singular sources at irregular internal interfaces which are usually not aligned with the mesh lines. The essential ingredient of the MIB method is the matching of the lowest order interface conditions simultaneously and exactly at select points on the interface with fictitious values. These fictitious values represent the extension of the solutions in its subdomain to other subdomains. For an interface that is not aligned with the mesh lines, appropriate coordinate transformations are defined such that the interface conditions in the normal direction can be handled in the Cartesian grid. These interface conditions are re-organized such that there is a primary direction in the final set of interface conditions. All the fictitious values are solved along its primary direction but can be used for the discretization of partial derivatives with respect to any coordinate. The matching of interface conditions provides essential algebraic relations for the expansion of these fictitious values in terms of real solution values and given jumps. Once the fictitious values are solved, in other words, once each subdomain is extended to a sufficiently large margin, the standard central finite difference scheme can be applied on the extended smooth domain to obtain desired high convergence and high accuracy. The same set of interface conditions can be iteratively used to solve for more fictitious values to support high order difference schemes. Among a variety of numerical methods for elliptic interface problems such as immersed interface method (IIM), immersed boundary method and ghost fluid method, MIB is the first numerical method that can be systmatically generalized to arbitrary high order for a general interface. The fourth-order and a sixth-order MIB schemes are successfully formulated in this work.

An interpolation formulation is also proposed in this dissertation whereby the interface conditions are matched via explicitly defined polynomials instead of through fictitious values. The comparison between these two implementations describes the similarities and relations among different interface methods based on Taylor expansion(IIM), Lagrange interpolation(MIB) and explicit polynomials(interpolation formulation).

#### 6.1.2 Interface Methods for the Poisson-Boltzmann equation

The application of interface methods can be found in a variety of disciplines such as fluid dynamics, material science, electromagnetics and molecular biology. However, this dissertation describes the first application of interface methods to the numerical solution of the Poisson-Boltzmann equation to conserve the continuity of the electrostatic potential and its flux at the dielectric interface. It is noticed that in most finite difference Poisson-Boltzmann solvers the molecular surface is only used to determine the accessibility of grid points but the exact location of the molecular surface is not considered. Therefore it was impossible for these methods to maintain the continuity of the potential flux at the molecular interface. In contrast, the analytical molecular surface generated by MSMS is incorporated into MIB for the discretization of the equation near the molecular surface. The accuracy of the solution, measured in either electrostatic potential field or the derived quantity such as electrostatic solvation energy, is significantly improved due to the consideration of the discontinuous potential gradient at the molecular surface. The accurate surface potential, solved with interface methods, makes it possible to quantitatively investigate the electrostatic complementary of the molecular surfaces involved in many bimolecular interactions such as electrostatically directed ligand association. The convergent electrostatic potential also provides an accurate calibration for the Born radii in the Generalized Born methods.

#### 6.2 Future research plan

### 6.2.1 Theoretical analysis and fast algebraic solvers for the MIB method

The matched interface and boundary method discussed in this dissertation is proved to be very promising for the interface problems, as illustrated by our extensive numerical experiments. Nevertheless, there are many directions for extending this thesis work.

Firstly, we expect to develop a solid analysis of the convergence of the MIB method. The approximation of the interface conditions in the current MIB method implies that the local truncation errors at the irregular points are one order lower than that at the regular points. Although it is commonly believed and actually demonstrated in this thesis that the convergence rate of the global error is not affected, we want to find out the rigorous conditions under which this claim is true.

Secondly, there are several numerical issues associated with the MIB method. The coefficient matrix of the linear system for the MIB method is always not symmetric and might not be diagonally dominant. Techniques to solve such systems efficiently are important for the applications of MIB method. The convergence might become a major problem with the increasing of the percentage of numbers of irregular points.

It is expected that some of the state-of-the-art techniques such as the multi-grid method, GMRES, QMR, *etc*, could by customized for the fast solution of the linear system derived from our MIB method.

Thirdly, the MIB method formulated in this thesis is based on the finite difference on a Cartesian grid. However, the essential idea of MIB, the matching of the interface conditions with fictitious values, might be extended to other formulations like variational principles or finite element methods on adaptive and/or composite meshes to attain more flexibility in handling the complex boundary or internal interface.

Finally, there are a large number of interface problems in which the interface is moving. It is therefore worthwhile to combine our MIB with the level set or similar marching methods for the moving interfaces to provide an accurate solutions for this type of problems.

### 6.2.2 Investigation of biological significance of accurate surface potential

In this thesis, the MIB method was used to solve for the convergent and accurate electrostatic potential from the Poisson-Boltzmann equation. The significance of the MIB method, however, is mostly substantiated by the accurate electrostatic solvation energies and the implication of an accurate surface potential in the analysis of the protein/ligand binding and the steering of ligands toward the protein. It is these applications that would signify the applications of the interface methods for the Poisson-Boltzmann equation. It is necessary to quantify the surface potential in analysis of the binding or association. The physical model and mathematical tools for such quantitative analysis, however, have not yet been well established. Most studies were based on the binding free energy or association entropy and molar partition coefficient[5]. Only a small number of works were based on the similarity analysis [82] on the electrostatic potential of different enzymes and the scoring functions using geometric-electrostatic correlation functions [26]. We hope that the accurate surface potential solved with the MIB method can be helpful in the accurate identification and prediction of binding site and binding affinity with these new analytical tools and protocols.

# **APPENDICES**

. . .

## APPENDIX A

# Kirkwood Solution to Non-Centered Charged Sphere

We consider a sphere of radius b in a solvent of dielectric constant  $\epsilon_w$ . The sphere has dielectric constant  $\epsilon_p$  and M discrete point charges  $q_1, q_2, \ldots, q_M$  inside. The sphere-solvent complex is schematically show below.

Following the general solution due to Kirkwood [39], one can choose a spherical coordinate system with origin at the center of the sphere. The positions of M charges are given by  $(r_k, \theta_k, \varphi_k), k = 1, ..., M$ . It should be noted that here  $\theta \in [0, \pi]$  is the polar angle from z-axis and  $\varphi \in [0, 2\pi)$  is the azimuthal angle in the x - y plane from the x-axis. This notational convention is different from that we used in the derivation of the interface scheme. It is adopted here to maintain the consistence with the original notation used by Kirkwood and the convention normally used in physics.

The electrostatic potential inside the sphere  $\phi_p$  satisfies the Poisson equation and is given by a spherical harmonic series

$$\phi_p = \sum_{k=1}^{M} \frac{q_k}{\epsilon_p |r - r_k|} + \sum_{n=0}^{\infty} \sum_{m=-n}^{n} B_{nm} r^n P_n^m(\cos \theta) e^{im\varphi}$$
(A.1)

$$= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{E_{nm}}{\epsilon_p r^{n+1}} P_n^m(\cos\theta) e^{im\varphi} + \sum_{n=0}^{\infty} \sum_{m=-n}^{n} B_{nm} r^n P_n^m(\cos\theta) e^{im\varphi} (A.2)$$

where  $|r - r_k|$  is the distance of the charge  $q_k$  from the point  $(r, \theta, \varphi)$  where the

electrostatic potential is to be measured. The first spherical harmonic represents the potential induced by the point charges  $q_k$  and the second is the dielectrical contributions of the solvent outside of the sphere.  $P_n^m(\cos\theta)$  are the associated Legendre functions of the first kind and i is the imaginary unit. Coefficients  $E_{nm}$  are determined by the distribution of the point charges, as

$$E_{nm} = \frac{(n - |m|)!}{(n + |m|)!} \sum_{k=1}^{M} q_k r_k^n P_n^m(\cos \theta_k) e^{-im\varphi_k}$$
(A.3)

Assume the ionic concentration is zero in the solvent, the electrostatic potential outside of the sphere  $\phi_w$  shall satisfy the Laplace equation and have a spherical harmonic expansion

$$\phi_w = \sum_{n=0}^{\infty} \sum_{m=-n}^n \left( \frac{C_{nm}}{r^{n+1}} + G_{nm} r^n \right) P_n^m(\cos \theta) e^{im\varphi}.$$
(A.4)

All the coefficients,  $B_{nm}$ ,  $C_{nm}$  and  $G_{nm}$ , have to be fixed by fitting the appropriate boundary conditions.

As potential  $\phi_w$  must be vanishing as r approaches to  $\infty$ ,  $G_{nm} = 0$  follows. The remaining coefficients,  $B_{nm}$  and  $C_{nm}$ , will be determined by fitting the continuity conditions on the surface of the sphere, i.e.,

$$\phi_w = \phi_p, \qquad \epsilon_w \frac{\partial \phi_w}{\partial r} = \epsilon_p \frac{\partial \phi_p}{\partial r},$$
 (A.5)

which provide two equations for every harmonic component of the potential:

$$\frac{E_{nm}}{\epsilon_p} + b^{2n+1} B_{nm} = C_{nm}, \tag{A.6}$$

$$(n+1)E_{nm} - n\epsilon_p b^{2n+1}B_{nm} = (n+1)\epsilon_w C_{nm}.$$
 (A.7)

Solution of these two equations gives the expression of  $B_{nm}$  and  $C_{nm}$ :

$$B_{nm} = -\frac{(\epsilon_w/\epsilon_p - 1)(n+1)E_{nm}}{[n\epsilon_p + (n+1)\epsilon_w]b^{2n+1}},$$
(A.8)

$$C_{nm} = \frac{(2n+1)E_{nm}}{n\epsilon_p + (n+1)\epsilon_w}.$$
(A.9)

The final real-valued solution to electrostatic potential can be obtained by considering only the real part of equations (A.2) and (A.4), although the coefficients  $E_{nm}, C_{nm}, B_{nm}$  determined as above are complex functions.

## References

- [1] L. Adams and T.P. Chartier. New geometric immersed interface multigrid solvers. SIAM Journal on Scientific Computing, 25:1516–1533, 2004.
- [2] B. H. Ameur, M. Burger, and B. Hackl. Level set methods for geometric inverse problems in linear elasticity. *Inverse Problems*, 20:673–696, 2004.
- [3] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5:207–213, 1970.
- [4] N.A. Baker. Biomolecular applications of Poisson-Boltzmann methods. John Wiley and Sons, 2005.
- [5] N. Ben-Tal, B. Honig, C. Miller, and S. McLaughlin. Electrostatic binding of proteins to membranes. theoretical predictions and experimental results with charybdotoxin and phospholipid vesicles. *Biophysical Journal*, 73(4):1717–1727, 1997.
- [6] P.A. Berthelsen. A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. *Journal of Computational Physics*, 197:364–386, 2004.
- [7] G. Biros, L.X. Ying, and D. Zorin. A fast solver for the stokes equations with distributed forces in complex geometries. *Journal of Computational Physics*, 193:317–348, 2004.
- [8] A. H. Boschitsch, M.O. Fenley, and H.X. Zhou. Fast boundary element method for the linear poisson-boltzmann equation. *Journal of physical Chemistry B*, 106(10):2741-2754, 2002.
- [9] A. H. Boschitsch and O. Fenley. Hybrid boundary element and finite difference method for solving the nonlinear poisson-boltzmann equation. Journal of Computational Chemistry, 25(7):935-955, 2004.
- [10] J. Bramble and J. King. A finite element method for interface problems in domains with smooth boundaries and interfaces. Advances in Computational Mathematics, 6:109-138, 1996.
- [11] Z. Chen and J. Zou. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numerische Mathematik*, 79:175-202, 1998.
- [12] M.L. Connolly. Molecular surfaces: A review. 1996.
- [13] S.Z. Deng, K. Ito, and Z.L. Li. Three-dimensional elliptic solvers for interface problems and applications. *Journal of Computational Physics*, 184:215–243, 2003.
- [14] M.A. Dumett and J.P. Keener. An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions. SIAM Journal on Scientific Computing, 25:348-367, 2003.
- [15] J. Dzubiella, J.M.J. Swanson, and J.A. McCammon. Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Physical review letters*, 96(8):087802+, 2006.
- [16] M. Eigen and E. Wicke. The thermodynamics of electrolytes at higher concertration. Journal of Chemical Physics, 58(9):702-714, 1954.
- [17] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersedboundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161:30-60, 2000.
- [18] R.P. Fedkiw, T. Aslam, B. Merriman, and S. S. Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.
- [19] M. Feig and C.L. Brooks III. Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Current Opinions in Structural Biology*, 14:217–224, 2004.
- [20] M. Feig, A. Onufriev, M.S. Lee, W. Im, D.A. Case, and C.L. Brooks III. Performance comparison of generalized born and poisson methods in the calculation of electrostatic solvation energies for protein stuctures. *Journal of Computational Chemistry*, 25:265–284, 2004.
- [21] J.Q. Feng. Electrostatic interaction between two charges dielectric spheres in contact. *Physical Review E*, 62(2):2891-2897, 2000.
- [22] B. Fornberg. Calculation of weights in finite difference formulas. SIAM Review, 40:685-691, 1998.
- [23] F. Gibou and R.P. Fedkiw. A forth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. Journal of Computational Physics, 202:577–601, 2005.
- [24] M.K. Gilson and B. Honig. Calculation of the total electrostatic energy of a macromolecular system: Solvation energies, binding energies and conformational analysis. *Proteins*, 4:7–18, 1988.
- [25] M.K. Gilson, K.A. Sharp, and B.H. Honig. Calculating the electrostatic potential of molecules in solution: method and error assessment. *Journal of Computational chemistry*, 9(4):327–335, 1987.
- [26] S.C. Gupta. The classic Stefen problem. JAI Press, 2003.
- [27] A. Heifetz, E. Katchalski-Katzir, and M. Eisenstein. Electrostatics in proteinprotein docking. *Protein Science*, 11(3), 2002.

- [28] M. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the poisson-boltzmann equation i. algorithms and examples. JOurnal of computational chemistry, 21(15):1319-1342, 2000.
- [29] M.J. Holst. The Poisson-Boltzmann Equation: Analysis and Multilevel Numerical Solution. PhD thesis, University of Illinois, Urbana-Champaign, 1993.
- [30] S. Hou and X.-D. Liu. A numerical method for solving variable coefficient elliptic equation with interfaces. *Journal of Computational Physics*, 202:411-445, 2005.
- [31] T.Y. Hou, Z.L. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the hele-shaw flow. *Journal of Computational Physics*, 134:236-252, 1997.
- [32] H. Huang and Z.L. Li. Convergence analysis of the immersed interface method. IMA Journal of Numerical Analysis, 19:583-608, 1999.
- [33] J.K. Hunter, Z.L. Li, and H. Zhao. Reactive autophobic spreading of drops. Journal of Computational Physics, 183:335-366, 2002.
- [34] G. Iaccarino and R. Verzicco. Immersed boundary technique for turbulent flow simulations. *Applied Mechnics Reviews*, pages 331-347, 2003.
- [35] O.G. Jenkins and K.L.C. Hunt. Nonlocal dielectric functions on the nanoscale: Screened forces from unscreened potentialsa. *Journal of Chemical Physics*, 119(16):8250-8256, 2003.
- [36] S. Jin and X.L. Wang. Robust numerical simulation of porosity evolution in chemical vapor infiltration ii. two-dimensional anisotropic fronts. *Journal of Computational Physics*, 179:557–577, 2002.
- [37] H. Johansen and P. Colella. A cartesian grid embedding boundary method for poisson's equation on irregular domains. *Journal of Computational Physics*, 147:60-85, 1998.
- [38] L. Joly, C. Ybert, E. Trizac, and L. Bocquet. Hydrodynamics within the electric double layer on slipping surfaces. *Physical review letters*, 93:257805, 2004.
- [39] J.D. Kandilarov. Immersed interface method for a reaction-diffusion equation with a moving own concentrated source, volume 2542, pages 506-513. Springer, 2003.
- [40] J. G. Kirkwood. Theory of solution of molecules containing widely separated charges with special application to zwitterions. *Journal of Chemical Physics*, 7:351–361, 1934.
- [41] L. L. Adams and Z.L. Li. The immersed interface/multigrid methods for interface problems. *SIAM Journal on Scientific Computing*, 24:463–479, 2002.
- [42] L. Lee and R.J. LeVeque. An immersed interface method for incompressible navier-stokes equations. SIAM Journal on Scientific Computing, 25:832–856, 2003.

- [43] R.J. LeVeque and Z.L. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM Journal on Numerical Analysis, 31:1019–1044, 1994.
- [44] Z.L. Li. A fast iterative algorithm for elliptic interface problems. SIAM Journal on Numerical Analysis, pages 230–254, 1998.
- [45] Z.L. Li. The immersed interface method using a finite element formulation. Applied Numerical Mathematics, 27(3):253-267, 1998.
- [46] Z.L. Li. An overview of the immersed interface method and its applications. Taiwanese Journal of Mathematics, 7:1-49, 2003.
- [47] Z.L. Li and K. Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. SIAM Journal on Scientific Computing, 23:339– 361, 2001.
- [48] Z.L. Li and K. Ito. The Immersed Interface Method Numerical Solutions of PDEs Involving Interfaces and Irregular Domains. SIAM, 2006.
- [49] Z.L. Li and S.R. Lubkin. Numerical analysis of interfacial two-dimensional stokes flow with discontinuous viscosity and variable surface tension. *International Journal for Numerical Methods in Fluids*, 37:525–540, 2001.
- [50] Z.L. Li, W.-C. Wang, I.-L. Chern, and M.-C. Lai. New formulations for interface problems in polar coordinates. *SIAM Journal on Scientific Computing*, 25:224–245, 2003.
- [51] S. Lim and C.S. Peskin. Simulations of the whirling instability by the immersed boundary method. SIAM Journal on Scientific Computing, 25:2066–2083, 2004.
- [52] X.D. Liu, Fedkiw R.P., and M. Kang. A boundary condition capturing method for poisson's equation on irregular domains. *Journal of Computational Physics*, 160:151–178, 2000.
- [53] B. Lombard and J. J. Piraux. How to incorporate the spring-mass conditions in finite-difference schemes. SIAM Journal on Scientific Computing, 24:1379–1407, 2003.
- [54] B.Z. Lu, D.Q. Zhang, and J.A. McCammon. Computation of electrostatic forces between solvated molecules determined by the poisson-boltzmann equation using a boundary element method. *Journal of Chemical Physics*, 122(21):214102+, 2005.
- [55] G. Luo, S. Malkova, J. Yoon, G. Schultz, B. Lin, M. Meron, L. Benjamin, P. Vanýsek, and M. L. Schlossman. Ion distributions near a liquid-liquid interface. *Science*, 311(5758):216-218, 2006.
- [56] R. Luo, H. David, L.and Hung, J. Devaney, and M.K. Gilson. Strength of solventexposed salt-bridges. *Journal of Physical Chemistry: B*, 103:727–736, 1999.
- [57] G.S. Manning. Limiting laws and counterion condensation in polyelectrolyte solutions v. further development of the chemical model. *Biophysical Chemistry*, 9(1):65–60, 1978.

- [58] N.L. Max and E. D. Getzoff. Spherical harmonic molecular surfaces. IEEE Computer Grphics, 8(4):42-50, 1988.
- [59] A. Mayo. The fast solution of poisson's and the biharmonic equations on irregular regions. SIAM Journal on Numerical Analysis, 21:285–299, 1984.
- [60] A. Mckenney, L. Greengard, , and A Mayo. A fast poisson solver for complex geometries. Journal of Computational Physics, 118:348-355, 1995.
- [61] H. Ohshima, E. Mishonova, and E. Alexov. Electrostatic interaction between two charged spherical molecules. *Biophysical Chemistry*, 57(2):189–203, 1996.
- [62] W.H. Orttung. Direct solution of the poisson equation for biomolecules of arbitrary shape, polarizability density and charge distribution. Annals of the New York Academy of Sciences, 303:22-37, 1977.
- [63] C.S. Peskin. Numerical analysis of blood flow in heart. Journal of Computational Physics, 25:220–252, 1977.
- [64] C.S. Peskin. Lectures on mathematical aspects of physiology. Lectures in Applied Mathematics, 19:69–107, 1981.
- [65] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in Fortran: the art of scientific computing*. Cambridge University Press, 1999.
- [66] A.A. Rashin. Hydration phenomena, classical electrostatics, and the boundary element method. *Journal of physical chemistry*, 94(5):1725–1733, 1990.
- [67] V.A. Rokerts, H.C Freeman, A.J. Olson, J.A. Tainer, and E.D. Getzoff. Electrostatic orientiation of the electron-transfer complex between plastocyanin and cycochrome c<sup>\*</sup>. Journal of Biological Chemistry, 266:13431-13441, 1991.
- [68] M.F. Sanner, A.J. Olson, and J.C. Spehner. Fast and robust computation of molecular surface. In Proceedings of the 11th symposium on Computational geometry, pages C6-C7, 1995.
- [69] M. Schulz and G. Steinebach. Two-dimensional modelling of the river rhine. Journal of Computational and Applied Mathematics, 145:11-20, 2002.
- [70] J.A. Sethian. Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, 1999.
- [71] J.A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163, 2000.
- [72] K. Sharp, A. Jean-Charles, and B. Honig. Journal of Physical Chemistry, 96:3822-3828, 1992.
- [73] F.B. Sheinerman, R. Norel, and B. Honig. Electrostatic aspects of proteinprotein interactions. *Current Opinion in Structural Biology*, 10:153–159, 2000.

- [74] I.A. Shkel, O.V. Tsodikov, and M.T. Jr. Record. Asymptotic solution of the cylindrical nonlinear poisson-boltzmann equation at low salt concentration: analytical expressions for surface potential and preferential interaction coefficient. *Proceedings of the National Academy of Sciences of the United States of America*, 99(5):2597-2602, 2002.
- [75] T. Simonson. Dielectric constant of cytochrome c from simulation in a water droplet including all electrostatic interactions. Journal of American Chemisty Society, 120:4875-4876, 1998.
- [76] T. Simonson and D. Perahia. Proceedings of the National Academy of Sciences of the United States of America, 92:1082–1086, 1995.
- [77] X. Song. An inhomogeneous model of protein dielectric properties: intrinsic polarizabilities of amino acids. *Journal of Chemical Physics*, 116(21):9359–9363, 2002.
- [78] N.V. Sushkin and G.D.L. Phillies. Charged dielectric spheres in electrolyte solutions: Induced dipole and counterion exclusion effects. *Journal of Chemical Physics*, 103(11):4600-4612, 1995.
- [79] A.K. Tornberg and B. Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200:462–488, 2004.
- [80] L. Visscher and K. G. Dyall. Dirac-fock atomic electronic structure calculations using different nuclear charge distributions. *Atmoic data and nuclear data tables*, 67:207-224, 1997.
- [81] J.V. Voorde, J. Vierendeels, and E. Dick. Flow simulations in rotary volumetric pumps and compressors with the fictitious domain method. *Journal of Computational and Applied Mathematics*, 168:491–499, 2004.
- [82] R.C. Wade, R.R. Gabdoulline, S.K. Lüdemann, and V. Lounnas. Eletrostatic streering and ionic techering in enzyme-ligand binding: insights from simulations. *Proceedings of the National Academy of Sciences of the United States of America*, 95(11):5942-5949, 1998.
- [83] J. H. Walther and G. Morgenthal. An immersed interface method for the vortexin-cell algorithm. *Journal of Turbulence*, 3(39):1-9, 2002.
- [84] W.-C. Wang. A jump condition capturing finite difference scheme for elliptic interface problems. SIAM Journal on Scientific Computing, 25:1479–1496, 2004.
- [85] A. Warshel and A. Papazyan. Electrostatic effects in macromolecules: fundamental concepts and pratical modeling. Correct Opinion in structural biology, 8:211-217, 1998.
- [86] J. Warwicker and H.C. Watson. Calculation of the electric potential in the active site cleft due to a-helix dipoles. *Journal of Molecular Biology*, 157(4):671–679, 1982.

- [87] G.W. Wei, Y.H. Sun, Y.C. Zhou, and M. Feig. Molecular multiresolution surfaces. 2005. arXiv:math-ph/0511001.
- [88] J. Weiser, P.S. Shenkin, and W.C. Still. Optimization of gaussian surface calculations and extension to solvent-accessible surface area. *Journal of Computational chemistry*, 20(7):688-703, 1999.
- [89] A. Wiegmann and K.P. Bube. The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. SIAM Journal on Numerical Analysis, 35:177-200, 1998.
- [90] A. Wiegmann and K.P. Bube. The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions. *SIAM Journal* on Numerical Analysis, 37:827-862, 2000.
- [91] J.C. Xu. Error estimates of the finite element method for the 2nd order elliptic equation with discontinuous coefficients. *Journal of Xiangtan University*, 1, 1982.
- [92] Y.-H. Tseng Y.-H. and J.H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192:593-623, 2003.
- [93] R.J. Zauhar and R.S. Morgan. A new method for computing the macromolecular electric potential. *Journal of Molecular Biology*, 186(4):815–820, 1985.
- [94] S. Zhao, , and G.W. Wei. High order fdtd methods via derivative matching for maxwell's equations with material interfaces. *Journal of Computational Physics*, 200:60–103, 2004.
- [95] R. Zhou and B.J. Berne. Can a continuum solvent model reproduce the free energy landscape of a  $\beta$ -hairpin folding in water? Proceedings of the National Academy of Sciences of the United States of America, 99:12777-12782, 2002.
- [96] Y.C. Zhou, S. Zhao, M. Feig, and G.W. Wei. High order matched interface and boundary (mib) schemes for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213:1–30, 2006.