Spring 2013

Immersed finite element method for interface problems with algebraic multigrid solver

Wenqiang Feng

Department:

IMMERSED FINITE ELEMENT METHOD FOR INTERFACE PROBLEMS WITH ALGEBRAIC MULTIGRID SOLVER

by

WENQIANG FENG

A THESIS

Presented to the Faculty of the Graduate School of

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN APPLIED MATHEMATICS

2013

Approved by

Dr. Xiaoming He, Advisor
Dr. John Singler
Dr. Yanzhi Zhang

ABSTRACT

This thesis is to discuss the bilinear and 2D linear immersed finite element (IFE) solutions generated from the algebraic multigrid solver for both stationary and moving interface problems. In contrast to the body-fitting mesh restriction of the traditional finite element methods or finite difference methods for interface problems, a number of numerical methods based on structured meshes independent of the interface have been developed. When these methods are applied to the real world applications, we often need to solve the corresponding large scale linear systems many times, which demands efficient solvers. The algebraic multigrid (AMG) method is a natural choice since it is independent of the geometry, which may be very complicated in interface problems. However, for those methods based on finite difference formulation and a structured mesh independent of the interface, the stiffness matrix of the linear system is usually not symmetric positive-definite, which demands extra efforts to design efficient multigrid methods. On the other hand, the stiffness matrix arising from the IFE methods are naturally symmetric positive-definite. Hence the IFE-AMG algorithm is proposed to solve the linear systems of the bilinear and 2D linear IFE methods for both stationary and moving interface problems after the IFE and multi-grid methods are reviewed respectively. The numerical examples demonstrate the features of the proposed algorithm, including the optimal convergence in both Ł^2 and semi-H^1 norms of the IFE-AMG solutions, the high efficiency with proper choice of the components and parameters of AMG, the influence of the tolerance and the smoother type of AMG on the convergence of the IFE solutions for the interface problems, and the relationship between the cost and the moving interface location.

ACKNOWLEDGEMENTS

Foremost, I would like to thank my parents, without whom none of this would have been possible.

Thanks to my advisor Dr. Xiaoming He for the continuous support of my Master study and research, for his patience, motivation, enthusiasm.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Yanzhi Zhang, Dr. John Singler, for their encouragement, insightful comments.

Thanks to the computers, without which none of the computations would have been feasible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

In this thesis, we first consider the following second order elliptic interface problem:

$$\begin{cases} -\nabla \cdot \big(\beta \nabla u\big) = f(X), & X \in \Omega, \\ u(X) = g(X), & X \in \partial\Omega, \end{cases} \tag{1.1}$$

together with the jump conditions on the interface Γ:

$$[u]\,|_{\Gamma} = 0, \tag{1.2}$$

$$\left[\beta \frac{\partial u}{\partial \mathbf{n}}\right]|_{\Gamma} = 0. \tag{1.3}$$

Here, see Figure 1.1, without loss of generality, we consider the case in which $\Omega \subset I\!R^2$ is an open rectangular domain, and the interface curve Γ is defined by a smooth function which separates Ω into two sub-domains Ω^-, Ω^+ such that $\overline{\Omega} = \overline{\Omega^-} \cup \overline{\Omega^+} \cup \Gamma$, and the coefficient $\beta(X)$ is a positive piecewise constant function defined by

$$\beta(X) = \begin{cases} \beta^-, & X \in \Omega^-, \\ \beta^+, & X \in \Omega^+. \end{cases}$$

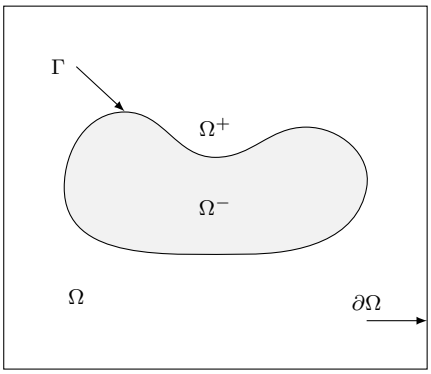

Figure 1.1. The sketch of domain Ω with the interface Γ.

We will also consider the following parabolic moving interface problem:

$$\begin{cases} u_t - \nabla \cdot (\beta \nabla u) = f(t, X), & X \in \Omega,\ t \in (0, T_{end}], \\ u(t, X) = g(t, X), & X \in \partial\Omega,\ t \in (0, T_{end}], \\ u(0, X) = u_0(X), & X \in \overline{\Omega}, \end{cases} \tag{1.4}$$

with the jump condition on a moving interface $\Gamma(t)$:

$$[u]|_{\Gamma(t)} = 0, \tag{1.5}$$

$$\left[\beta \frac{\partial u}{\partial \mathbf{n}}\right]\Big|_{\Gamma(t)} = 0. \tag{1.6}$$

Without loss of generality, we consider the case in which the interface curve $\Gamma(t)$ is defined by a smooth function $\Gamma : [0, T_{end}] \to \Omega$. At any time $t \in [0, T_{end}]$, the interface $\Gamma(t)$ separates Ω into two sub-domains $\Omega^+(t)$ and $\Omega^-(t)$ such that $\Omega = \Omega^+(t) \cup \Omega^-(t) \cup \Gamma(t)$, see Figure 1.2 for an illustration. The coefficient function $\beta(t, X)$ is discontinuous across the interface $\Gamma(t)$. For simplicity, we assume $\beta(t, X)$ is a piece-wise constant function as follows:

$$\beta(t, X) = \begin{cases} \beta^-, & X \in \Omega^-(t), \\ \beta^+, & X \in \Omega^+(t). \end{cases}$$

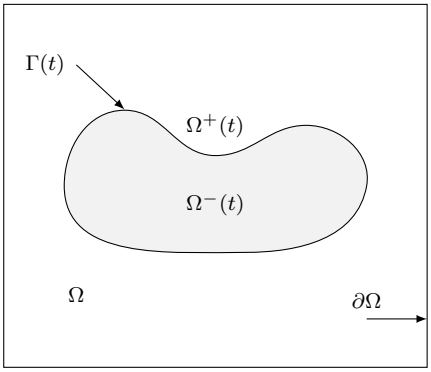

Figure 1.2. A sketch of the domain for the moving interface problem.

The stationary interface problems (1.1)-(1.3) and the moving interface problem (1.4)-(1.6) are involved in many applications of engineering and sciences, such as the field injection problem [25, 81], flow problem [5, 15], electromagnetic problems[6, 10, 42], shape/toplogy optimization problem [19, 9], and the Stefan problem [13, 61]. These interface problems can be solved by conventional finite difference or finite element methods with optimal convergence if a body-fitting mesh is utilized [7, 8, 11, 14, 37]. However, there are many applications, such as Particle-In-Cell method for plasma particle simulation [43, 59, 60, 74, 75] and moving interface problems [36], in which a structured mesh independent of the interface is preferred for solving the interface problems.

Therefore, many efforts have been attempted to develop such numerical methods for solving interface problems on structured meshes (Figure 1.3) independent of the interface even if their geometries are non-trivial. In the finite difference formulation, the immersed boundary method [28, 44, 54, 63, 64], immersed interface method [21, 22, 23, 49, 71, 80], matched interface and boundary method [24, 82, 83, 84, 85], cut-cell method [39, 40], and embedded boundary method [38, 41] have been developed.

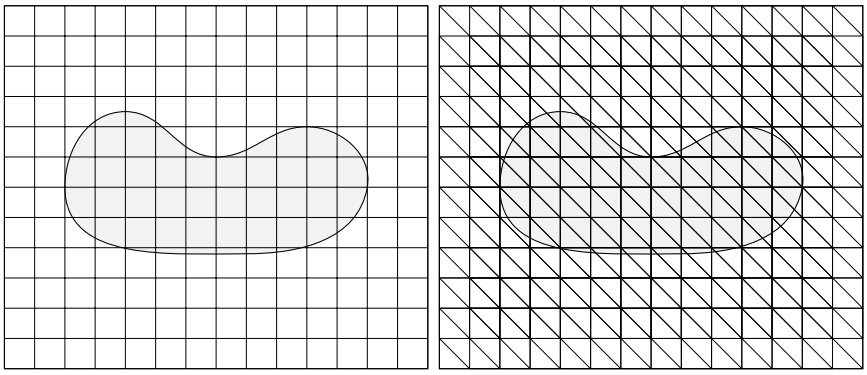

Figure 1.3. Rectangular and triangular Cartesian meshes independent of the interface.

In real world applications, we often need to solve large scale linear systems arising from these methods many times due to various realistic needs, such as the curse of the dimensionality, the high accuracy requirement, and moving interface. This demands very efficient solvers. The multigrid methods, which are well known for their efficiency and natural preconditioning feature, perform efficiently on Cartesian meshes which can be naturally provided by the aforementioned methods for interface problems. L. Adams and Z. Li [48] designed a geometric multigrid method for the immersed interface method of the second order elliptic interface problems. Furthermore, L. Adams and T. P. Chartier [46] developed a new restriction operator and the corresponding interpolation operator to guarantee that the coarse-grid matrices are M-matrices. R. D. Guy and B. Philip also applied a multigrid method for an implicit immersed boundary equations [29]. Furthermore, L. Adams and T. P. Chartier [47] also utilized a similar idea in [46] to design the corresponding algebraic multigrid method and compare it with the geometric one.

It is natural to consider the algebraic multigrid method [66, 70] since it is independent of the geometry, which may be very complicated in interface problems. However, extra efforts are usually needed in order to design efficient multigrid methods to solve the non-symmetric linear systems arising from those methods based on finite difference formulation and a structured mesh independent of the interface. On the other hand, the immersed finite element (IFE) methods [3, 4, 12, 17, 16, 18, 20, 26, 27, 31, 32, 33, 34, 35, 36, 42, 45, 50, 51, 52, 53, 55, 56, 57, 58, 69, 73, 78, 76, 79], which are developed under the general framework of finite elements and proposed by using piecewise local basis functions according to the interface jump conditions while their meshes do not have to be aligned with interfaces, naturally provide symmetric positive-definite matrices for the above interface problems. While minimizing the extra efforts to modify the traditional finite element packages, the IFE methods can also easily deal with complex interface with optimal accuracy order. Hence we believe that the combination of the features of the algebraic multigrid method (such as its efficiency, preconditioning capability and independence of the geometry) and the features of the IFE methods (such as their symmetric

positive-definite matrices, capability to handle the interface without using body-fitting meshes, and optimal convergence rates) can generate very efficient and competitive numerical methods for large-scale applications in which a structured mesh independent of the interface is preferred for solving the interface problems.

The rest of this thesis is organized as follows. In chapter 2, we recall the definitions of the bilinear and 2D linear IFE spaces. In chapter 3, we discuss the numerical scheme with bilinear IFE for the stationary interface problem and the numerical scheme with 2D linear IFE for the moving interface problem. In chapter 4, we recall the standard two-grid method and the multigrid. In chapter 5, we propose the IFE-AMG based on the standard multigrid techniques. In chapter 6, we provide the numerical experiments for both elliptic stationary interface problem and parabolic moving interface problem. Finally, we gave a summary of this thesis and the proposed further research in chapter 7.

2. THE BILINEAR AND 2D LINEAR IMMERSED FINITE ELEMENTS

In this section, we briefly recall the bilinear IFE space [31, 55] and the 2D linear IFE space [52, 53].

2.1. THE BILINEAR IMMERSED FINITE ELEMENTS

First, we consider a rectangular Cartesian mesh (see the left graph of Figure 1.3) independent of the interface. Let $\mathcal{T}_h$ denote the collection of all elements in a mesh with parameter h. When h is small enough, most of elements in $\mathcal{T}_h$ are non-interface elements not intersecting with the interface Γ. Only those elements in the vicinity of Γ have the possibility to be cut through by Γ and become the so-called interface elements. We will use $\mathcal{T}_{int}$ to denote the collection of all interface elements of $\mathcal{T}_h$.

On each non-interface element T, we let the local finite element space $S_h(T)$ be $S_h^{non}(T)$, which is spanned by the four standard bilinear nodal basis functions $\psi_i(x, y), i = 1, 2, 3, 4$ on a rectangular element. To describe the local IFE space on an interface element $T \in \mathcal{T}_{int}$, we assume that the vertices of T are $A_i, i = 1, 2, 3, 4$, with $A_i = (x_i, y_i)^T$. Without loss of generality, we assume that ∂T intersects with Γ at two points $D = (x_{_D}, y_{_D})^T$ and $E = (x_{_E}, y_{_E})^T$. When the mesh is fine enough, there are two types of rectangle interface elements with 12 cases (Figure2.2-Figure2.13). Type I are those for which the interface intersects with two of its adjacent edges; Type II are those for which the interface intersects with two of its opposite edges, see the sketch in Figure 2.1.

Since the line $\overline{DE}$ separates T into two subsets T^- and T^+, we naturally form a piecewise function by two bilinear polynomials defined in T^- and T^+, respectively. Then by using the interface conditions (1.2)-(1.3), the bilinear immersed functions are defined

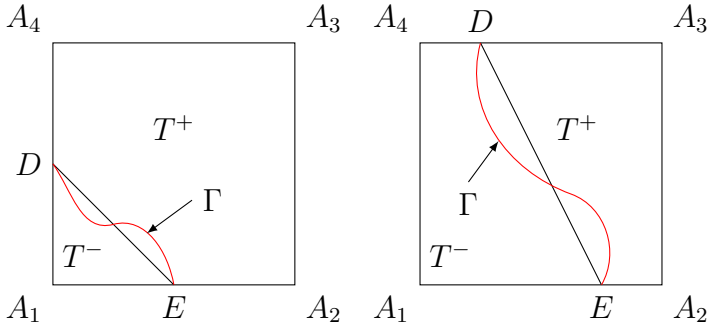

Figure 2.1. Two typical rectangular interface elements. The element on the left is of Type I while the one on the right is of Type II.

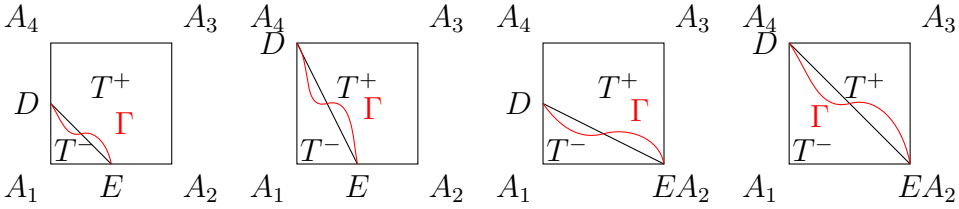

Figure 2.2. Case 1: $f(A_1) < 0$, $f(A_2) \geq 0$, $f(A_3) > 0$,$f(A_4) \geq 0$

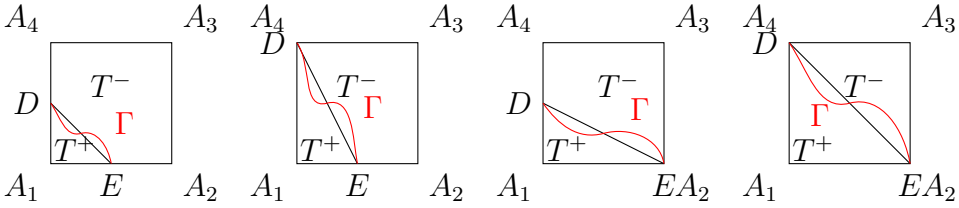

Figure 2.3. Case 2: $f(A_1) > 0$, $f(A_2) \leq 0$, $f(A_3) < 0$,$f(A_4) \leq 0$

as follows [31, 55]:

$$\psi(x,y) = \begin{cases} \psi^-(x,y) = a^- x + b^- y + c^- + d^- xy, & (x,y) \in T^-, \\ \psi^+(x,y) = a^+ x + b^+ y + c^+ + d^+ xy, & (x,y) \in T^+, \\ \psi^-(D) = \psi^+(D), \quad \psi^-(E) = \psi^+(E), d^- = d^+, & \\ \int_{\overline{DE}} \left(\beta^- \dfrac{\partial \psi^-}{\partial \mathbf{n}_{\overline{DE}}} - \beta^+ \dfrac{\partial \psi^+}{\partial \mathbf{n}_{\overline{DE}}} \right) ds = 0. & \end{cases} . \tag{2.7}$$

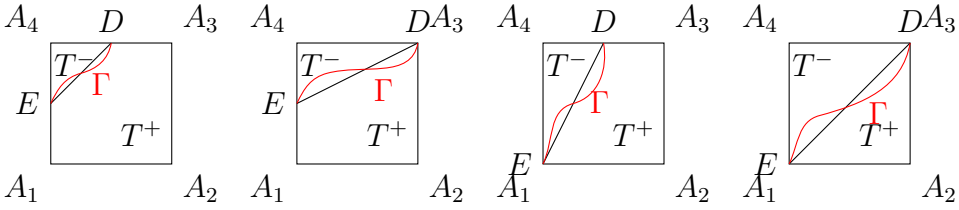

Figure 2.4. Case 3: $f(A_1) \geq 0$, $f(A_2) > 0$, $f(A_3) \geq 0$,$f(A_4) < 0$

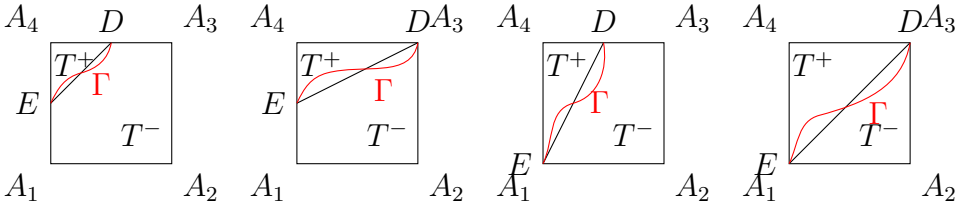

Figure 2.5. Case 4: $f(A_1) \leq 0$, $f(A_2) < 0$, $f(A_3) \leq 0$,$f(A_4) > 0$

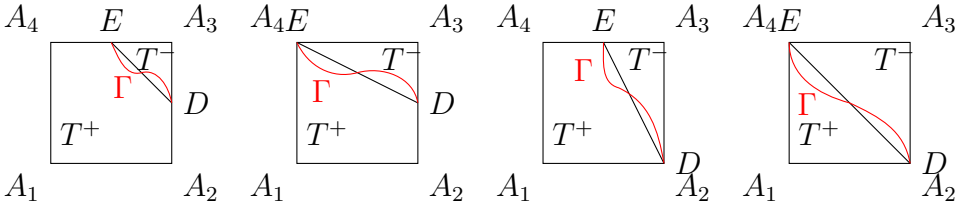

Figure 2.6. Case 5: $f(A_1) > 0$, $f(A_2) \geq 0$, $f(A_3) < 0$,$f(A_4) \geq 0$

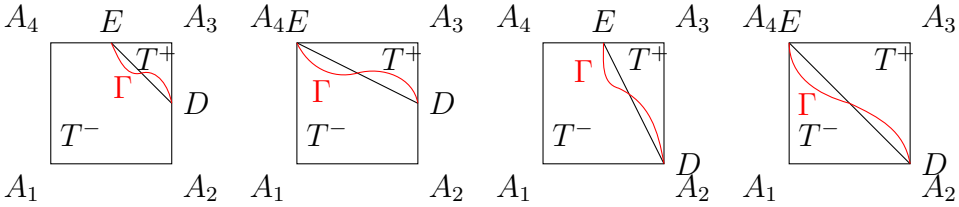

Figure 2.7. Case 6: $f(A_1) < 0$, $f(A_2) \leq 0$, $f(A_3) > 0$,$f(A_4) \leq 0$

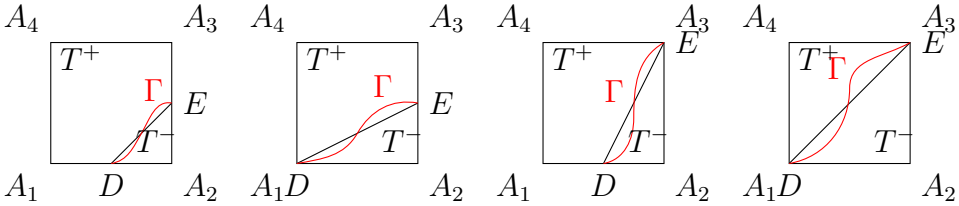

Figure 2.8. Case 7: $f(A_1) \geq 0$, $f(A_2) < 0$, $f(A_3) \geq 0$,$f(A_4) > 0$

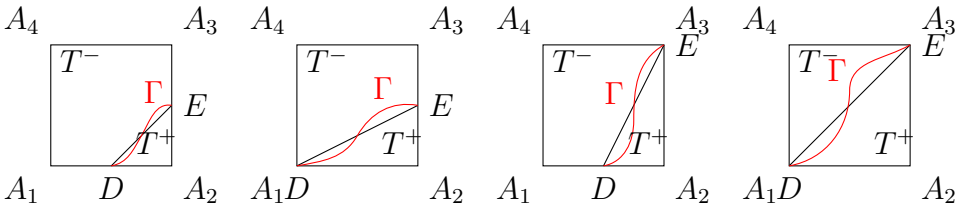

Figure 2.9. Case 8: $f(A_1) \leq 0$, $f(A_2) > 0$, $f(A_3) \leq 0$,$f(A_4) < 0$

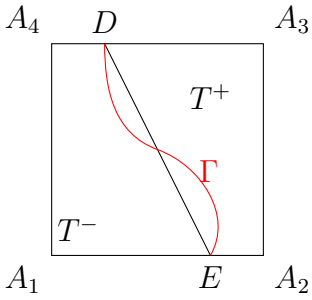

Figure 2.10. Case 9: $f(A_1) < 0$, $f(A_2) > 0$, $f(A_3) > 0$,$f(A_4) < 0$

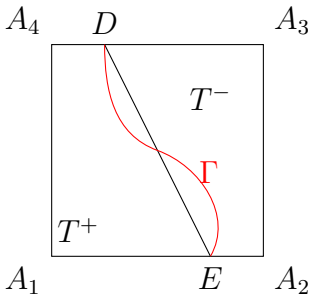

Figure 2.11. Case 10: $f(A_1) > 0$, $f(A_2) < 0$, $f(A_3) < 0$,$f(A_4) > 0$

Now let $\psi_i(X)$ be the bilinear IFE function described by (2.7) such that

$$\psi_i(x_j, y_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

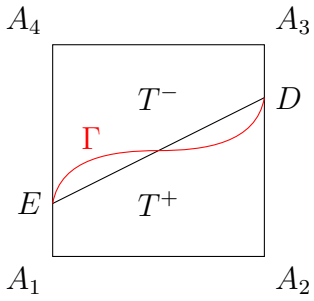

Figure 2.12. Case 11: $f(A_1) > 0$, $f(A_2) > 0$, $f(A_3) < 0$,$f(A_4) < 0$

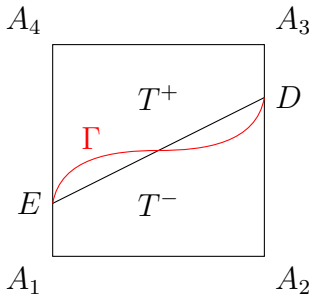

Figure 2.13. Case 12: $f(A_1) < 0$, $f(A_2) < 0$, $f(A_3) > 0$,$f(A_4) > 0$

for $1 \leq i, j \leq 4$, and we call them the bilinear IFE nodal basis functions on an interface element T. We then let $S_h^{int}(T) = span\{\psi_i,\, i = 1, 2, 3, 4\}$.

In summery, for each element $T \in \mathcal{T}_h$, we define

$$S_h(T) = \begin{cases} S_h^{non}(T), & \text{if } T \text{ is a non-interface element,} \\ \\ S_h^{int}(T), & \text{if } T \text{ is an interface element.} \end{cases}$$

Let $\mathcal{N}_h = \{X_i\}_{i=1}^N$ denote the set of nodes in $\mathcal{T}_h$, $\mathcal{N}_h^o = \mathcal{N}_h \cap \Omega$, $\mathcal{N}_h^b = \mathcal{N}_h \cap \partial\Omega$, $\mathcal{I}_h^o = \{i : X_i \in \mathcal{N}_h^o\}$, and $\mathcal{I}_h^b = \{i : X_i \in \mathcal{N}_h^b\}$. Define $\phi_i(X)$ $(i = 1, \cdots, N)$ to be a piecewise

bilinear function such that

$$\phi_i|_T \in S_h(T), \quad \forall T \in \mathcal{T}_h \text{ and } \phi_i(X_j) = \delta_{ij}, \quad \forall X_j \in \mathcal{N}_h.$$

Then the bilinear IFE space on the whole domain Ω is defined as

$$S_h^{IFE}(\Omega) = span\{\phi_i(X) : 1 \le i \le N\}.$$

We also define the subspace $S_{h,0}^{IFE}(\Omega) \subset S_h^{IFE}(\Omega)$ such that

$$S_{h,0}^{IFE}(\Omega) = span\{\phi_i(X) : i \in \mathcal{I}_h^o\}.$$

Remark 2.1. *Here $\phi_i(X)$ is a global bilinear IFE basis function if X_i is a node of any interface element. Otherwise, $\phi_i(X)$ is a standard global bilinear finite element basis function associated with the node X_i. Since IFE functions are discontinuous on the element edges cut by the interface, the immersed finite elements are nonconforming [31, 55].*

2.2. THE 2D LINEAR IMMERSED FINITE ELEMENTS

In the following we consider a triangular Cartesian mesh (see the right graph in Figure 1.3) independent of the interface. On each of the non-interface element T, we let the local finite element space $S_h(T)$ be $S_h^{non}(T)$ spanned by the three standard linear nodal basis functions $\phi_i(x, y),\ i = 1, 2, 3$ on T. For an interface element T with vertices $A_i = (x_i, y_i)^T,\ i = 1, 2, 3$, without loss of generality, we assume that ∂T intersects with Γ at two points $D = (x_{_D}, y_{_D})^T$ and $E = (x_{_E}, y_{_E})^T$. There is only one type of triangle interface elements, see the sketch in Figure 2.14.

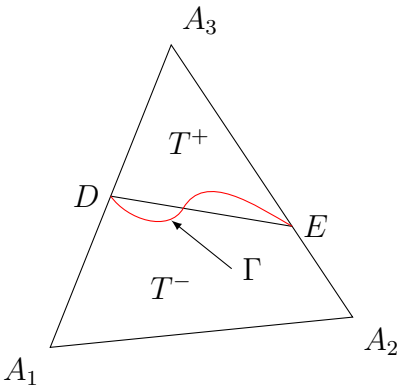

Figure 2.14. A typical triangular interface element.

Then by using the interface conditions (1.2)-(1.3), the 2D linear immersed finite element function are defined as follows [52, 53]:

$$\psi(x,y) = \begin{cases} \psi^-(x,y) = a^-x + b^-y + c^-, & (x,y) \in T^-, \\ \psi^+(x,y) = a^+x + b^+y + c^+, & (x,y) \in T^+, \\ \psi^-(D) = \psi^+(D), \quad \psi^-(E) = \psi^+(E), & \\ \beta^- \dfrac{\partial \psi^-}{\partial \mathbf{n}_{\overline{DE}}} - \beta^+ \dfrac{\partial \psi^+}{\partial \mathbf{n}_{\overline{DE}}} = 0, & \end{cases} \tag{2.8}$$

where $\mathbf{n}_{\overline{DE}}$ is the unit vector perpendicular to the line $\overline{DE}$. We let $\psi_i(X)$ be the linear IFE function described by (2.8) such that

$$\psi_i(x_j, y_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases}$$

for $1 \leq i, j \leq 3$, and we call them the 2D linear IFE nodal basis functions on an interface element T. We then let $S_h^{int}(T) = span\{\phi_i,\, i = 1, 2, 3\}$. Then we can use the same way as

in the bilinear IFE space to define $S_h(T)$, $\phi_i(X)$, $S_h^{IFE}(\Omega)$ and $S_{h,0}^{IFE}(\Omega)$ for the 2D linear IFE space.

2.3. THE EXISTENCE AND UNIQUENESS OF IFE BASIS FUNCTIONS

In this subsection, we discuss the existence and uniqueness for the 2D linear and bilinear IFE basis functions. As usual, we only need to dicuss the nodal linear IFE basis functions $\widehat{\phi}_i(\widehat{X})$, i = 1, 2, 3, on the reference element $\widehat{T}$ with vertices $\widehat{A}_i = (\widehat{x}_i, \widehat{y}_i)^T$ (Figure 2.15), i = 1, 2, 3:

$$\widehat{A}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \widehat{A}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \widehat{A}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.9}$$

and the intersection point $\widehat{D}$ and $\widehat{E}$:

$$\widehat{D} = \begin{pmatrix} 0 \\ \widehat{b} \end{pmatrix}, \widehat{E} = \begin{pmatrix} \widehat{a} \\ 1 - \widehat{a} \end{pmatrix}. \tag{2.10}$$

The interface element T is related to the corresponding reference element $\widehat{T}$ (Figure 2.15)

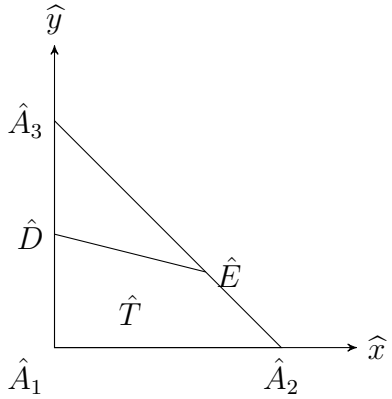

Figure 2.15. The reference interface element.

by the affine mapping (Figure 2.16):

$$X = F(\widehat{X}) = M\widehat{X} + B, \quad X = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \widehat{X} = \begin{pmatrix} \widehat{x} \\ \widehat{y} \end{pmatrix}, \tag{2.11}$$

where

$$M = \begin{pmatrix} x_{\tilde{A}_2} - x_{\tilde{A}_1} & x_{\tilde{A}_3} - x_{\tilde{A}_1} \\ y_{\tilde{A}_2} - y_{\tilde{A}_1} & y_{\tilde{A}_3} - y_{\tilde{A}_1} \end{pmatrix}, \quad B = \begin{pmatrix} x_{\tilde{A}_1} \\ y_{\tilde{A}_1} \end{pmatrix}. \tag{2.12}$$

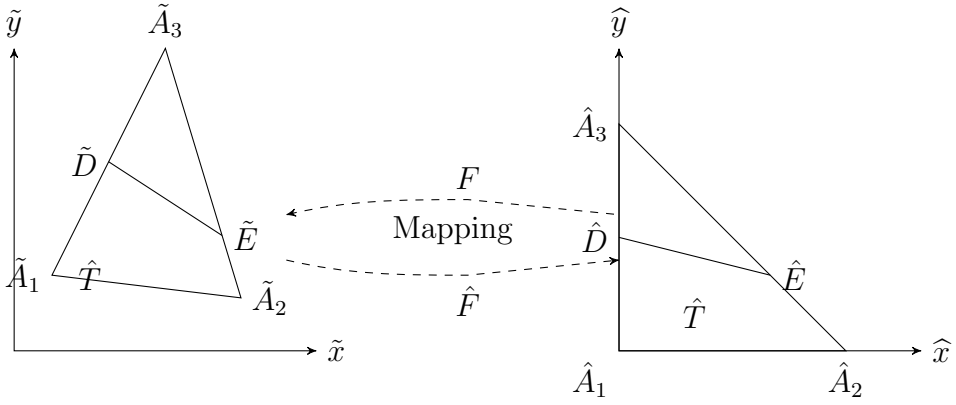

Figure 2.16. Affine mapping between the rotated local interface 2D linear element and the corresponding reference element.

Let $\widehat{\phi}_i(\widehat{X}), i = 1, 2, 3$ be the 2D linear IFE nodal basis on the reference element $\widehat{T}$ (Figure.2.15) such that[52, 53]

$$\phi_i(\widehat{x}, \widehat{y}) = \begin{cases} \widehat{a_i^-}\widehat{x} + \widehat{b_i^-}\widehat{y} + \widehat{c_i^-}, & \text{if} \quad (\widehat{x}, \widehat{y}) \in \widehat{T}^-, \\ \widehat{a_i^+}\widehat{x} + \widehat{b_i^+}\widehat{y} + \widehat{c_i^+}, & \text{if} \quad (\widehat{x}, \widehat{y}) \in \widehat{T}^+, \end{cases} \tag{2.13}$$

and

$$\begin{cases} \widehat{\phi}_i(\widehat{A}_j) = \delta_{ij} = \begin{cases} 1, & \text{if} \quad i = j, \\ 0, & \text{if} \quad i \neq j, \end{cases} \\ \widehat{\phi}_i^-(\widehat{D}) = \widehat{\phi}_i^+(\widehat{D}), \widehat{\phi}_i^-(E) = \widehat{\phi}_i^+(\widehat{E}), \\ \beta^- \dfrac{\partial \widehat{\phi}_i^-}{\partial \mathbf{n}_{\overline{\widehat{D}\widehat{E}}}} - \beta^+ \dfrac{\partial \widehat{\phi}_i^+}{\partial \mathbf{n}_{\overline{\widehat{D}\widehat{E}}}} = 0. \end{cases} \tag{2.14}$$

By using the same idea in [30], we can reproof the following existence and uniqueness result for 2D linear IFE basis functions:

Theorem 2.1. *[52, 53] Given the reference triangle $\triangle \widehat{A}_1\widehat{A}_2\widehat{A}_3$ as indicated in figure(2.14). The piecewise linear basis functions $\hat{\phi}_i(\hat{x}, \hat{y})$ are uniquely determined by (2.14).*

Proof. The C^0 function consists of piecewise linear polynomials which have six degrees of freedom. At the three vertices of the element, we specify the function values. The additional degrees of freedom are utilized to satisfy the approximation of the jump conditions. Therefore, we can get five linear equations as follows after substituting the coordinates of the three vertices of the original element and two intersection points into (2.14):

1. For the vertex $\hat{A}_1(0, 0)$: By substituting the coordinates of the vertex $\hat{A}_1(0, 0)$ into the basis function (2.13), we get

$$\hat{\phi}_i(\hat{A}_1) = 0a_1^- + 0b_1^- + c_1^- + 0a_1^+ + 0b_1^+ + 0c_1^+ = \delta_{i1} \tag{2.15}$$

2. For the vertex $\hat{A}_2(1, 0)$: By substituting the coordinates of the vertex $\hat{A}_2(1, 0)$ into the basis function (2.13), we get

$$\hat{\phi}_i(\hat{A}_2) = a_1^- + 0b_1^- + c_1^- + 0a_1^+ + 0b_1^+ + 0c_1^+ = \delta_{i2} \tag{2.16}$$

3. For the vertex $\hat{A}_3(0,1)$: By substituting the coordinates of the vertex $\hat{A}_3(0,1)$ into the basis function (2.13), we get

$$\hat{\phi}_i(\hat{A}_3) = 0a_1^- + 0b_1^- + 0c_1^- + 0a_1^+ + b_1^+ + c_1^+ = \delta_{i3} \tag{2.17}$$

4. For the interface intersection point $D(0,\hat{b})$: By substituting the coordinates of the vertex $\hat{D}(0,\hat{b})$ into the basis function (2.13), we get

$$\hat{\phi}_i(\hat{D}) = 0a_1^- + \hat{b}b_1^- + c_1^- - 0a_1^+ - \hat{b}b_1^+ - c_1^+ = 0 \tag{2.18}$$

5. For the interface intersection point $\hat{E}(\hat{a},1-\hat{a})$: By substituting the coordinates of the vertex $\hat{E}(\hat{a},1-\hat{a})$ into the basis function (2.13), we get

$$\hat{\phi}_i(\hat{E}) = \hat{a}a_1^- + (1-\hat{a})b_1^- + c_1^- - \hat{a}a_1^+ - (1-\hat{a})b_1^+ - c_1^+ = 0 \tag{2.19}$$

6. We can also get another linear function from the flux jump condition:

$$\widehat{a_i^-}\alpha - b_i^- = \rho(\widehat{a_i^+}\alpha - b_i^+), \tag{2.20}$$

where $\rho = \frac{\beta^+}{\beta^-}$ and the normal direction of the line $\overline{\widehat{D}\widehat{E}}$ is $(\alpha,-1)$ with $\alpha = (1-\widehat{a}-\widehat{b})/\widehat{a}$.

Thus, the linear system arising from (2.14) for the unknowns $\widehat{a_i^-}, \widehat{b_i^-}, \widehat{c_i^-}, \widehat{a_i^+}, \widehat{b_i^+}$, and $\widehat{c_i^+}$ is given as follows:

$$A \begin{pmatrix} \widehat{a_i^-} \\ \widehat{b_i^-} \\ \widehat{c_i^-} \\ \widehat{a_i^+} \\ \widehat{b_i^+} \\ \widehat{c_i^+} \end{pmatrix} = \mathbf{b}_i \tag{2.21}$$

where

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & \widehat{b} & 1 & 0 & -\widehat{b} & -1 \\ \widehat{a} & 1-\widehat{a} & 1 & -\widehat{a} & \widehat{a}-1 & -1 \\ \alpha & -1 & 0 & -\alpha\rho & \rho & 0 \end{pmatrix} \tag{2.22}$$

and

$$\mathbf{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.23}$$

Direct calculations give us

$$rank(A) = 6.$$

Thus from the theory of linear algebra, there is a unique solution for the linear system(2.21).

□

The existence and uniqueness of the bilinear IFE basis functions have been proved similarly in [30]. We would like to point out that the affine mapping between the local element and the reference element (Figure.2.18) for the bilinear IFE is different from that of the standard bilinear finite element. We need to do a rotation (Figure.2.17) before we applying the affine mapping(Figure.2.18) to the bilinear IFE.

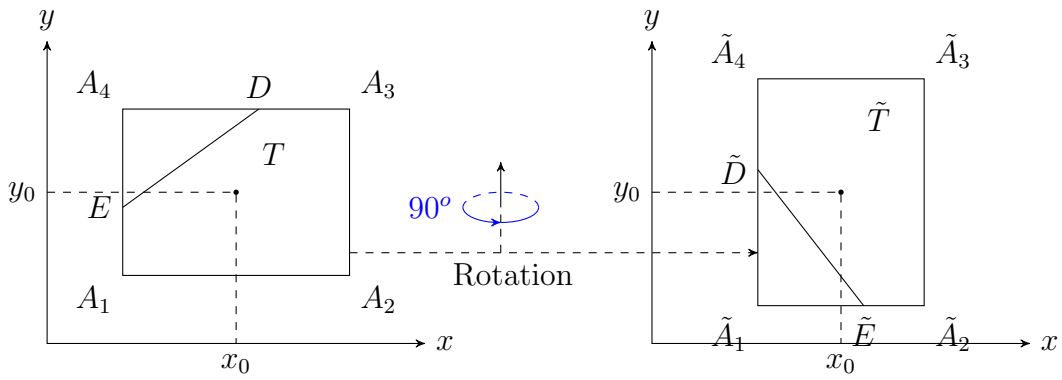

Figure 2.17. The rotation of the local interface element.

In the following subsection, the relationship between the interface coefficient β and the energy norm will be investigated.

2.4. UPPER BOUND OF THE IFE FUNCTIONS IN ENERGY NORM

It has been shown that the IFE basis functions and the IFE interpolation error have uniform upper bounds independent of the interface and the jump coefficients in L^2 and H^1 norms[30]. In this section, we will show that this is not true for the following energy

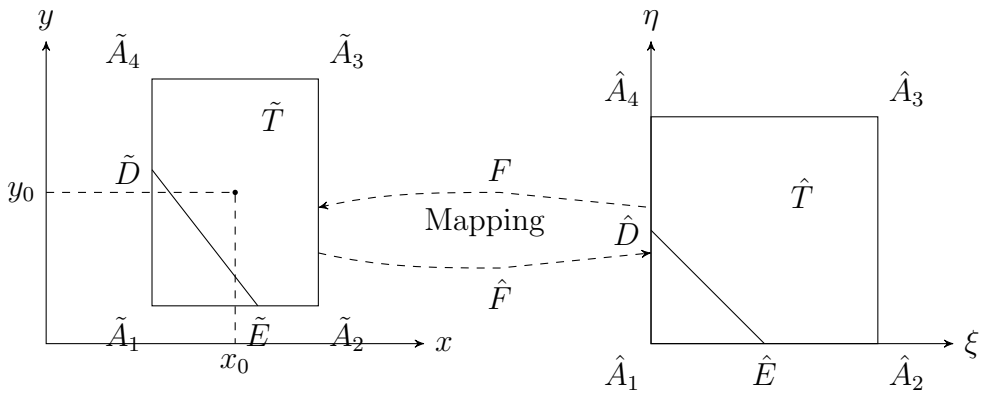

Figure 2.18. Affine mapping between the rotated local interface bilinear element and the corresponding reference element.

norms

$$||u - u_I||^2_{0,\beta,T} = \int_T \beta(u - u_I)^2 dxdy,$$
$$|u - u_I|_{1,\beta,T} = ||\nabla(u - u_I)||_{0,\beta,T}.$$

Since the traditional analysis framework of the geometric algebraic method does need the uniform upper bounds independent of the jump coefficients in the energy norm, it is not clear if the geometric algebraic method can be applied to the immersed finite elements without any extra efforts.

Lemma 2.1. *The constant C in the IFE interpolation error estimates*

$$||u - u_I||_{0,\beta,T} \leq Ch^2|u|_{2,\beta,T}$$
$$|u - u_I|_{1,\beta,T} \leq Ch|u|_{2,\beta,T}$$

in the energy norms may depend on the jump coefficients.

Proof. In order to prove this lemma, we only need to provide one example. By solving the linear system (2.21) for the specific intersection points $D(0, 1/2)$, $E(3/4, 1/4)$ (Figure.2.19-Figure.2.20), and jump coefficients $\beta^+ = \beta_1$, $\beta^- = \beta_2$, (without loss generality, we assume

$\beta_1 > \beta_2$), we can get the specific basis functions $\hat{\phi}_1(\widehat{x},\widehat{y})$, $\hat{\phi}_2(\widehat{x},\widehat{y})$, $\hat{\phi}_3(\widehat{x},\widehat{y})$ as follows:

$$\hat{\phi}_1(\widehat{x},\widehat{y}) = \begin{cases} -\widehat{x} + \dfrac{3\beta_2 - 23\beta_1}{11\beta_1 + 9\beta_2}\widehat{y} + 1, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^-, \\ -\dfrac{3\beta_1 + 17\beta_2}{11\beta_1 + 9\beta_2}\widehat{x} + \dfrac{\beta_1 - 21\beta_2}{11\beta_1 + 9\beta_2}\widehat{y} + \dfrac{21\beta_2 - \beta_1}{11\beta_1 + 9\beta_2}, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^+. \end{cases} \tag{2.24}$$

$$\hat{\phi}_2(\widehat{x},\widehat{y}) = \begin{cases} \widehat{x} + \dfrac{3\beta_1 - 3\beta_2}{11\beta_1 + 9\beta_2}\widehat{y}, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^-, \\ \dfrac{9\beta_1 + 11\beta_2}{11\beta_1 + 9\beta_2}\widehat{x} + \dfrac{3\beta_2 - 3\beta_1}{11\beta_1 + 9\beta_2}\widehat{y} + \dfrac{3\beta_1 - 3\beta_2}{11\beta_1 + 9\beta_2}, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^+. \end{cases} \tag{2.25}$$

$$\hat{\phi}_3(\widehat{x},\widehat{y}) = \begin{cases} \dfrac{20\beta_1}{11\beta_1 + 9\beta_2}\widehat{y}, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^-, \\ \dfrac{6\beta_2 - 6\beta_1}{11\beta_1 + 9\beta_2}\widehat{x} + \dfrac{2\beta_1 + 18\beta_2}{11\beta_1 + 9\beta_2}\widehat{y} + \dfrac{9\beta_1 - 9\beta_2}{11\beta_1 + 9\beta_2}, & \text{if} \quad (\widehat{x},\widehat{y}) \in \widehat{T}^+. \end{cases} \tag{2.26}$$

Consider the function

$$u = \begin{cases} u_{\widehat{T}^+} = \frac{(\frac{1}{3}\hat{x} + \hat{y} - 1/2)^2}{\beta_1}, \\ u_{\widehat{T}^-} = \frac{(\frac{1}{3}\hat{x} + \hat{y} - 1/2)^2}{\beta_2}. \end{cases} \tag{2.27}$$

So

$$u(\hat{A}_1) = \frac{1}{4\beta_2},\ u(\hat{A}_2) = \frac{1}{36\beta_2},\ u(\hat{A}_3) = \frac{1}{4\beta_1}, \tag{2.28}$$

and the interpolation function will be

$$\begin{aligned} u_I &= u(\hat{A}_1)\hat{\phi}_1 + u(\hat{A}_2)\hat{\phi}_2 + u(\hat{A}_3)\hat{\phi}_3 \\ &= \frac{-(18\beta_1^2\hat{x} - 267\beta_1\beta_2 - 54\beta_2^2\tilde{x} - 6\beta_1^2\tilde{y} - 162\beta_2^2\hat{y} + 6\beta_1^2 + 81\beta_2^2 + 196\beta_1\beta_2\hat{x} + 168\beta_1\beta_2\hat{y})}{(36\beta_1\beta_2(11\beta_1 + 9\beta_2))} \\ &= \frac{(-18\beta_1^2 + 54\beta_2^2 - 196\beta_1\beta_2)\hat{x} + (6\beta_1^2 + 162\beta_2^2 - 168\beta_1\beta_2)\hat{y} + 267\beta_1\beta_2 - 6\beta_1^2 - 81\beta_2^2}{36\beta_1\beta_2(11\beta_1 + 9\beta_2)}. \end{aligned}$$

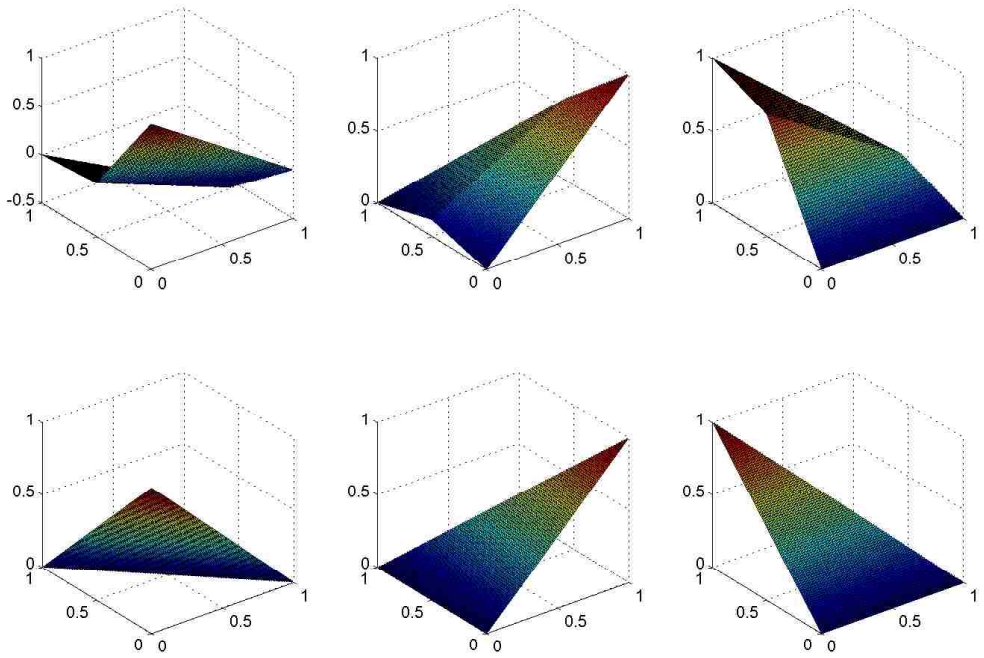

Figure 2.19. The IFE basis and FE basis. The first row are the IFE basis and The second row are the Standard FE basis.

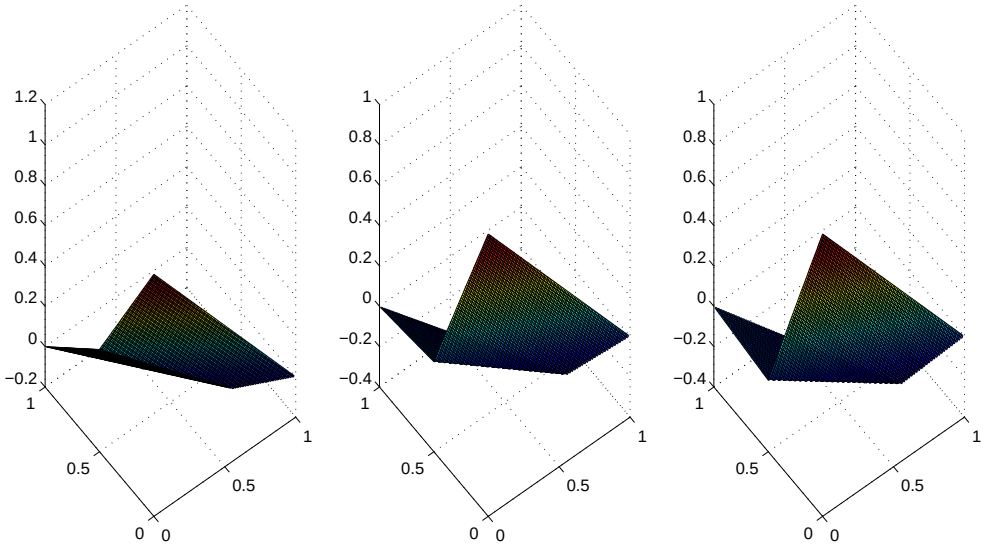

Figure 2.20. The figure of ϕ_1. Left: $\beta_1 = 2,\ \beta_2 = 1$; Middle: $\beta_1 = 10,\ \beta_2 = 1$; Right: $\beta_1 = 10000,\ \beta_2 = 1$.

Therefore, the corresponding interpolation error in the energy norm on $\hat{T}^+$ is

$$\begin{aligned}
&||u-u_I||^2_{0,\beta,\hat{T}^+} \\
&= \int_{\hat{T}^+} \beta_1 (u-u_I)^2 d\hat{x}d\hat{y} \\
&= \int_{1/2}^{1} \left(\int_0^{1-\hat{y}} \beta_1 (u-u_I)^2 d\hat{x} \right) d\hat{y} + \int_{1/4}^{1/2} \left(\int_{3/2-3\hat{y}}^{1-\hat{y}} \beta_1 (u-u_I)^2 d\hat{x} \right) d\hat{y} \\
&= \frac{-(405\beta_1^4 - 6777\beta_1^3\beta_2 + 2234\beta_1^2\beta_2^2 - 837\beta_1\beta_2^3 - 18225\beta_2^4)}{124416\beta_1\beta_2^2(11\beta_1+9\beta_2)^2}
\end{aligned} \tag{2.29}$$

Since $\beta_1 > \beta_2$ and the degree of the β_1 in the numerator is greater than the degree in denominator, so $||u-u_I||^2_{0,\beta,\hat{T}^+}$ cannot have an uniform upper bound independent of β_1. Therefore, the constant C in the upper bound of the interpolation error in the energy norm $||u-u_I||^2_{0,\beta,\hat{T}}$ may depend on the coefficient jump. Similar conclusion can be proved for $\|u-u_I\|_{1,\beta,\hat{T}}$ in the same way. □

3. NUMERICAL SCHEMES FOR THE INTERFACE PROBLEMS

In this section we will first describe how to use the IFE function spaces to formulate the linear system for the elliptic interface problems. And then we will apply the same idea to a Crank-Nicolson-type IFE method for the parabolic moving interface problem.

3.1. NUMERICAL SCHEME FOR ELLIPTIC INTERFACE PROBLEM

To formulate the linear system arising from the IFE method of the elliptic interface problem, we will first briefly recall the weak formulation and the IFE formulation [35, 53, 55]. Multiply the differential equation (1.1) by any $v \in H_0^1(\Omega)$ and integrate it over $\Omega^s (s = +, -)$ to have

$$-\int_{\Omega^s} \nabla \cdot \big(\beta^s \nabla u\big) v \; dxdy \;\; = \;\; \int_{\Omega^s} fv \; dxdy, \forall \; v \in H_0^1(\Omega).$$

Then a straightforward application of the Green's formula leads to

$$\int_{\Omega^s} \beta^s \nabla u \cdot \nabla v \; dxdy - \int_{\partial\Omega^s} \beta^s \frac{\partial u}{\partial \mathbf{n}} v \; ds \;\; = \;\; \int_{\Omega^s} fv \; dxdy, \;\; s = +, -, \forall \; v \in H_0^1(\Omega).$$

Summing the above equation over s and applying the flux jump condition (1.3) on the interface, we obtain the weak formulation for the elliptic interface problem: find $u \in H^1(\Omega)$ such that

$$\int_{\Omega} \beta \nabla u \cdot \nabla v \; dxdy \;\; = \;\; \int_{\Omega} fv \; dxdy, \forall \; v \in H_0^1(\Omega).$$

Then the IFE formulation is to find $u_h \in S_h^{IFE}(\Omega)$ such that

$$\sum_{T \in \mathcal{T}_h} \int_T \beta \nabla u_h \cdot \nabla v_h \; dxdy \;\; = \;\; \sum_{T \in \mathcal{T}_h} \int_T f v_h \; dxdy, \forall \; v_h \in S_{h,0}^{IFE}(\Omega). \tag{3.30}$$

Based on the construction of the IFE space $S_h^{IFE}(\Omega)$ and the given Dirichlet boundary condition, the approximate solution to the elliptic interface problem (1.1)-(1.3) is taken in the following form:

$$u_h(X) = \sum_{k\in\mathcal{I}_h^o} u_k\phi_k(X) + \sum_{s\in\mathcal{I}_h^b} g(X_s)\phi_s(X). \tag{3.31}$$

Plugging (3.31) into (3.30) and substituting $\phi_l \in S_{h,0}^{IFE}(\Omega)$ for v_h, the IFE formulation becomes: find the coefficients u_k ($k \in \mathcal{I}_h^o$) such that

$$\begin{aligned} &\sum_{k\in\mathcal{I}_h^o}\left(\sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_l\cdot\nabla\phi_k dX\right)u_k \\ =\ &\sum_{T\in\mathcal{T}_h}\int_T \phi_l f dX - \sum_{s\in\mathcal{I}_h^b}\left(\sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_l\cdot\nabla\phi_s dX\right)g(X_s),\ \forall l\in\mathcal{I}_h^o. \end{aligned}$$

Assume that the set $\mathcal{I}_h^o$ has n elements. Define k_i to be the i^{th} element in $\mathcal{I}_h^o$. Then

$$\begin{aligned} &\sum_{1\le j\le n}\left(\sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_{k_i}\cdot\nabla\phi_{k_j} dX\right)u_{k_j} \\ =\ &\sum_{T\in\mathcal{T}_h}\int_T \phi_{k_i} f dX - \sum_{s\in\mathcal{I}_h^b}\left(\sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_{k_i}\cdot\nabla\phi_s dX\right)g(X_s),\ 1\le i\le n. \end{aligned}$$

Rewriting the above system in matrix formulation yields

$$A_h\vec{u}_h = \vec{b}_h, \tag{3.32}$$

where

- $A_h = (a_{ij})_{n\times n}$ is the stiffness matrix with

$$a_{ij} = \sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_{k_i}\cdot\nabla\phi_{k_j} dX.$$

- $\vec{b}_h = (b_i)_{n\times 1}$ is the source vector with

$$b_i = \sum_{T\in\mathcal{T}_h}\int_T \phi_{k_i} f dX - \sum_{s\in\mathcal{I}_h^b}\left(\sum_{T\in\mathcal{T}_h}\int_T \beta\nabla\phi_{k_i}\cdot\nabla\phi_s dX\right) g(X_s).$$

- $\vec{u}_h = (u_{k_i})_{n\times 1}$ is the unknown vector.

Remark 3.1. *It is straightforward to see that the stiffness matrix A_h arising from the IFE method is symmetric positive definite, which is critical to the algebraic multigrid method. The optimal convergence rates are also expected for the IFE solutions u_h, which are second order in L^2 norm and first order in H^1 semi-norm for the linear and bilinear IFEs. The numerical experiments in the next section will verify this expectation.*

3.2. NUMERICAL SCHEME FOR MOVING INTERFACE PROBLEM

Now we discuss the parabolic moving interface problem (1.4)-(1.6), for which we will utilize a Crank-Nicolson-type IFE method [36] together with the above IFE-AMG algorithm. The matrix formed at each time iteration step will be different from the one from elliptic equation, but still symmetric positive definite.

At any time t, we define $\mathcal{N}_h^{i,t}$ to be the set of nodes of all interface elements at the time t and let $\mathcal{N}_h^{n,t} = \mathcal{N}_h/\mathcal{N}_h^{i,t}$ denote the rest of the nodes. Let $\phi_j^t(X)$ denote the global bilinear or linear IFE basis function, which has been discussed in Section 2, associated with the node $X_j \in \mathcal{N}_h^{i,t}$ at the time t while $\phi_j^t(X)$ is a standard global linear finite element basis function for $X_j \in \mathcal{N}_h^{n,t}$. Then we look for an IFE approximate solution to the parabolic interface problem (1.4) - (1.6) in the following form:

$$u_h(t, X) = \sum_{X_j\in\mathcal{N}_h} u_j(t)\phi_j^t(X). \tag{3.33}$$

From the above definitions, we know that if $X_j \in \mathcal{N}_h^{i,t}$, then $\phi_j^t(X)$ depends on the interface location, hence depends on the time t. On the other hand, $\phi_j^t(X)$ is independent

of the time t for $X_j \in \mathcal{N}_h^{n,t}$. Therefore,

$$\frac{\partial u_h(t,X)}{\partial t} = \sum_{X_j \in \mathcal{N}_h} \frac{\partial u_j(t)}{\partial t}\phi_j^t(X) + \sum_{X_j \in \mathcal{N}_h^{i,t}} u_j(t)\frac{\partial \phi_j^t(X)}{\partial t}. \tag{3.34}$$

Based on the following standard weak form at a given time t:

$$\int_\Omega v\frac{\partial u}{\partial t}\mathrm{d}X + \int_\Omega \nabla v \cdot (\beta \nabla u)\mathrm{d}X = \int_\Omega v f \mathrm{d}X, \quad \forall\ v \in H_0^1(\Omega),$$

which is equivalent to

$$\sum_{T\in\mathcal{T}_h}\int_T v\frac{\partial u}{\partial t}\mathrm{d}X + \sum_{T\in\mathcal{T}_h}\int_T \nabla v \cdot (\beta \nabla u)\mathrm{d}X = \int_\Omega v f \mathrm{d}X, \quad \forall\ v \in H_0^1(\Omega),$$

the following system can be obtained from the IFE semi-discretization [36]:

$$M_h(t)\mathbf{u}'(t) + K_h(t)\mathbf{u}(t) + A_h(t)\mathbf{u}(t) = \mathbf{f}(t), \tag{3.35}$$

where

- $M_h(t) = (m_{ij}(t))$ is the mass matrix with $m_{ij} = \int_\Omega \phi_i^t \phi_j^t \mathrm{d}X$.
- $K_h(t) = (k_{ij}(t))$ with $k_{ij} = \int_\Omega \phi_i^t \frac{\partial \phi_j^t}{\partial t}\mathrm{d}X$.
- $A_h(t) = (a_{ij}(t))$ is the stiffness matrix with $a_{ij} = \int_\Omega \nabla\phi_i^t \cdot \left(\beta\nabla\phi_j^t\right)\mathrm{d}X$.
- $\mathbf{f}(t) = (f_i(t))$ is the source vector with $f_i(t) = \int_\Omega \phi_i^t f \mathrm{d}X$.
- $\mathbf{u}(t)$ is the vector whose entries are $u_j(t)$, i.e. $\mathbf{u}(t) = (u_j(t))$.

For the time discretization, without loss of generality, we use a uniform partition $0 = t_0 < t_1 < \cdots < t_N = T$ in time, where $t_n = n\tau$ with $\tau = T_{end}/N$. Then we look for approximations such that

$$u_h^n(X) = \sum_{X_j \in \mathcal{N}_h} u_j^n \phi_j^{t_n}(X) \approx u_h(t_n, X).$$

In effect, we look for $\vec{u}^n = (u_j^n) \approx \vec{u}(t_n)$, for $n = 1, 2, \cdots, N$. Then applying the idea of Crank-Nicolson type discretization to (3.35) leads to the following algorithm [36]:

$$\begin{aligned}&\left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2}K_h^{n+\frac{1}{2},n+\frac{1}{2}}\right)\vec{u}^{n+1}\\ &\quad = \left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} - \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}} - \frac{\tau}{2}K_h^{n+\frac{1}{2},n+\frac{1}{2}}\right)\vec{u}^n + \tau\vec{f}^{n+\frac{1}{2},n+\frac{1}{2}}.\end{aligned}$$

where

- $M_h^{n_v,n_u} = \left(m_{ij}^{n_v,n_u}\right)$ is the mass matrix, where $m_{ij}^{n_v,n_u} = \int_\Omega \phi_i^{t_{n_v}}\phi_j^{t_{n_u}}\mathrm{d}X$.
- $A_h^{n_\beta,n_v,n_u} = \left(a_{ij}^{n_\beta,n_v,n_u}\right)$ is the stiffness matrix, where $a_{ij}^{n_\beta,n_v,n_u} = \int_\Omega \nabla\phi_i^{t_{n_v}}\cdot(\beta^{t_{n_\beta}}\nabla\phi_j^{t_{n_u}})\mathrm{d}X$.
- $K_h^{n_v,n_u} = \left(k_{ij}^{n_v,n_u}\right)$, where $k_{ij}^{n_v,n_u} = \int_\Omega \phi_i^{t_{n_v}}\left(\frac{\partial}{\partial t}\phi_j^{t_{n_u}}\right)\mathrm{d}X$.
- $\mathbf{f}^{n_v,n_f} = \left(f_i^{n_v,n_f}\right)$ is right hand side vector, where $f_i^{n_v,n_f} = \int_\Omega \phi_i^{t_{n_v}} f^{t_{n_f}}\mathrm{d}X$.

Here n_v, n_u, n_β, and n_f denote the time levels for the test function v, trial function u, coefficient function β, source function f, respectively.

The matrix $M_h^{n+\frac{1}{2},n+\frac{1}{2}}+\frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}+\frac{\tau}{2}K_h^{n+\frac{1}{2},n+\frac{1}{2}}$ is not symmetric since $K_h^{n+\frac{1}{2},n+\frac{1}{2}}$ is not symmetric. However, a simplified algorithm has been proposed based on Theorem 3.1 in [36] and numerically illustrated to be optimally convergent in [36]:

$$\left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}\right)\vec{u}^{n+1} = \left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} - \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}\right)\vec{u}^n + \tau\vec{f}^{n+\frac{1}{2},n+\frac{1}{2}}.$$

In this algorithm, the matrix $M_h^{n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}$ is symmetric positive definite matrix, which is critical to the AMG method. Then the IFE-AMG algorithm proposed in the following can be utilized to solve the linear system at each time iteration step with

$$\begin{aligned}A_h^1 &= M_h^{n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}},\\ \vec{b}_h^1 &= \left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} - \frac{\tau}{2}A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}\right)\vec{u}^n + \tau\vec{f}^{n+\frac{1}{2},n+\frac{1}{2}}.\end{aligned}$$

4. TWO GRID AND MULTI-GRID METHODS

The multigird method can be considered as the recursion of the two-grid method. Therefore, in this section, we will first review some basic principles of two-grid method and then briefly introduce the multi-grid method. Suppose at the finest grid a mesh-size of h is used and the resulting problem we are trying to solve is

$$A_h \vec{u}_h = \vec{b}_h. \tag{4.36}$$

Let $\vec{u}_h$ and $\tilde{\vec{u}}_h$ to be the approximation solution and the exact solution respectively. Then error in $\tilde{\vec{u}}_h^h$ is

$$e_h = \vec{u}_h - \tilde{\vec{u}}_h. \tag{4.37}$$

and the residual is

$$r_h = \vec{b}_h - A_h \vec{u}_h. \tag{4.38}$$

Since A_h is linear, by the definition of the error (4.37) and the residual (4.38), the error satisfies

$$A_h e_h = r_h.$$

It is known that the high frequencies of the error can be reduced in a few iterations, but low frequencies are reduced very slowly(Figure.4.1-Figure.4.5). Therefore, the extremely effective multigrid idea is to change the low frequencies to a coarse grid, on which the "smooth becomes rough" and the low frequencies act like higher frequencies. The classical iterative methods, such as Jacobi or Gauss-Seidel[68],[67], can be used to pro-

duce smooth errors on each grid level. In the following we recall some common smoother operators.

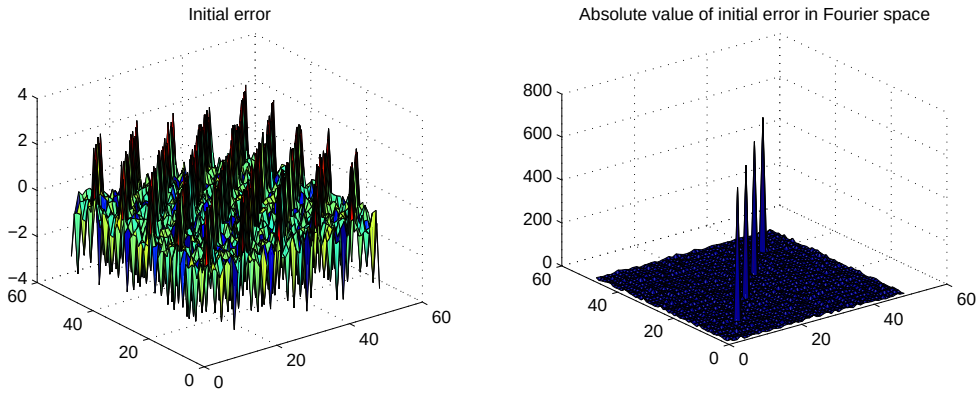

Figure 4.1. The initial error and the absolute value of initial error in Fourier space.

4.1. THE SMOOTHING OPERATOR

First, we start with the following decomposition

$$A = D - L - U, \tag{4.39}$$

where D is the diagonal of A, $-L$ and $-U$ are the strict lower part and the strict upper part, respectively, as illustrated in (Figure.4.6). In the following, $\xi_i^{(k)}$ denotes the i-th component of the iterate $\vec{u}_k$ and β_i is the i-th component of the right hand side $\vec{b}$.

4.1.1. Jacobi Iteration Method. The Jacobi iteration determines the i-th component of the next approximation so as to annihilate the i-th component of the residual vector. Based on

$$(b - Ax_{k+1})_i = 0,$$

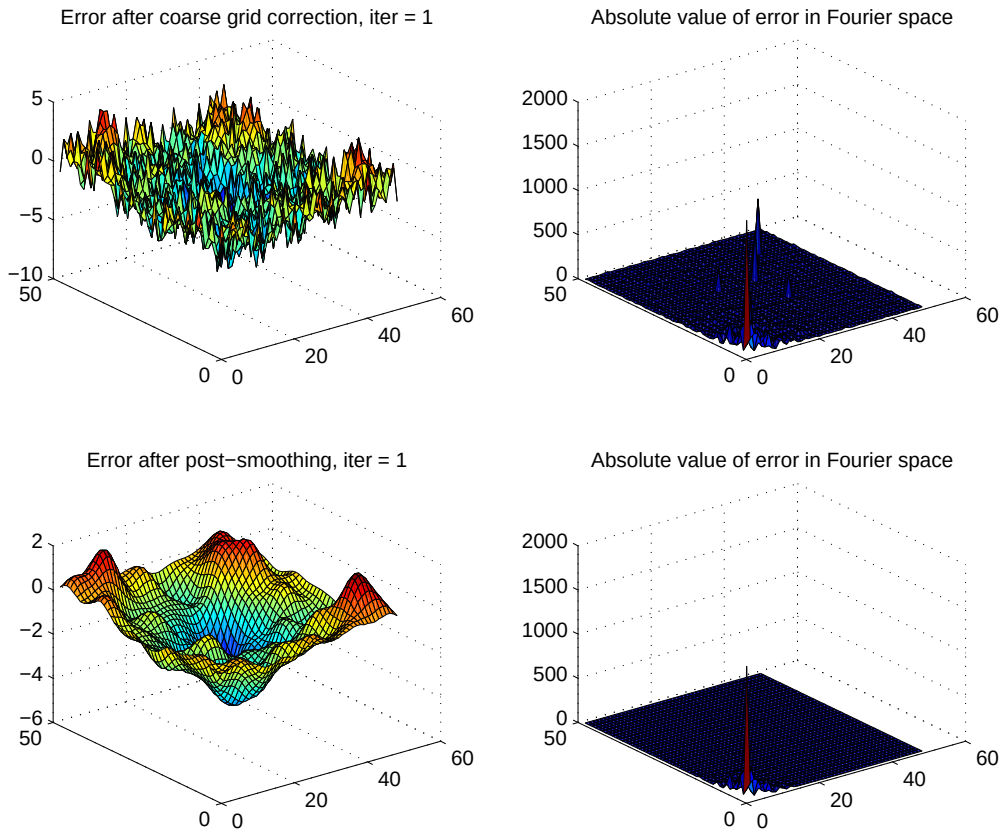

Figure 4.2. The error after 1_{st} coarse grid correction and post-smoothing with the absolute value of error in Fourier space.

in which $(*)_i$ represents the i-th component of the column vector $*$, we have

$$a_{ii}\xi_i^{(k+1)} = -\sum_{j=1,j\neq i}^{n} a_{ij}\xi_i^{(k)} + \beta_i$$

or

$$\xi_i^{(k+1)} = \frac{1}{a_{ii}}\left(\beta_i - \sum_{j=1,j\neq i}^{n} a_{ij}\xi_i^{(k)}\right), \quad i = 1,\ldots,n. \tag{4.40}$$

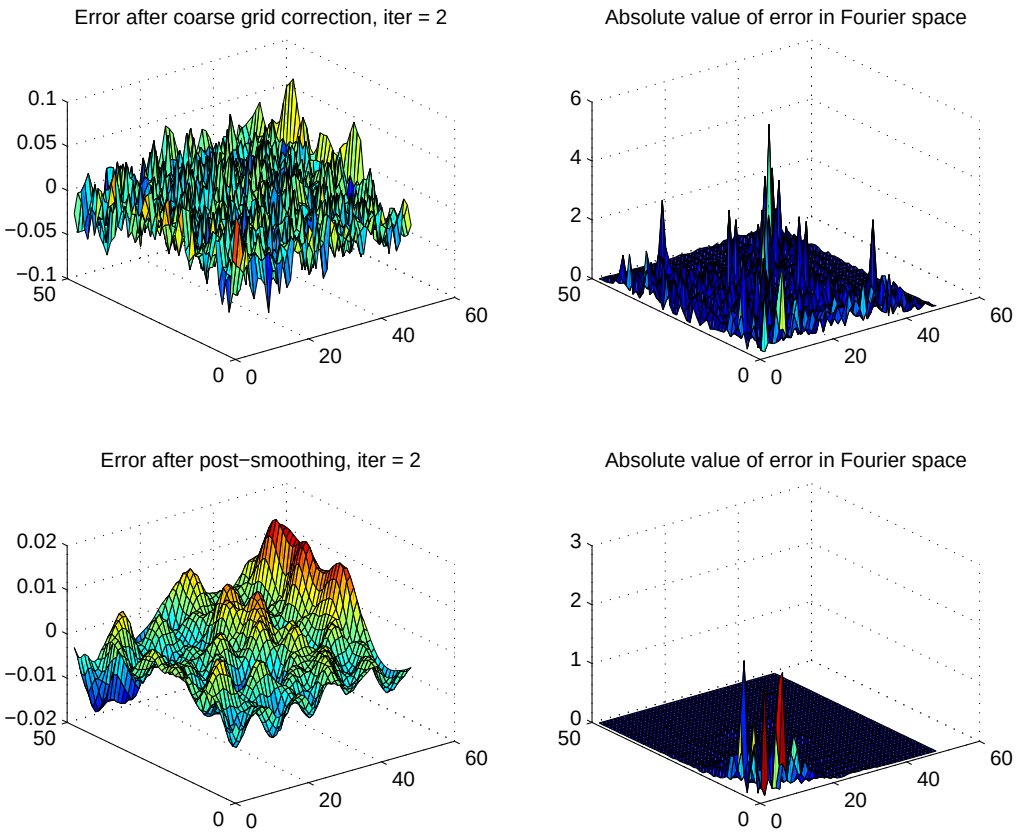

Figure 4.3. The error after 2_{nd} coarse grid correction and post-smoothing with the absolute value of error in Fourier space.

The above equation (4.40) can be used to rewrite the Jacobi iteration in the vector form as

$$x^{k+1} = D^{-1}(L+U)x^k + D^{-1}b. \tag{4.41}$$

4.1.2. Gauss-Seidel Iteration Method. Similarly, the Gauss-Seidel iteration corrects the i-th component of the current approximate solution, again to annihilate the i-th component of the residual. However, the approximate solution of the Gauss-Seidel iteration is updated immediately after the new component is determined. The newly computed components ξ_i^k can be changed within a working vector which is redefined at

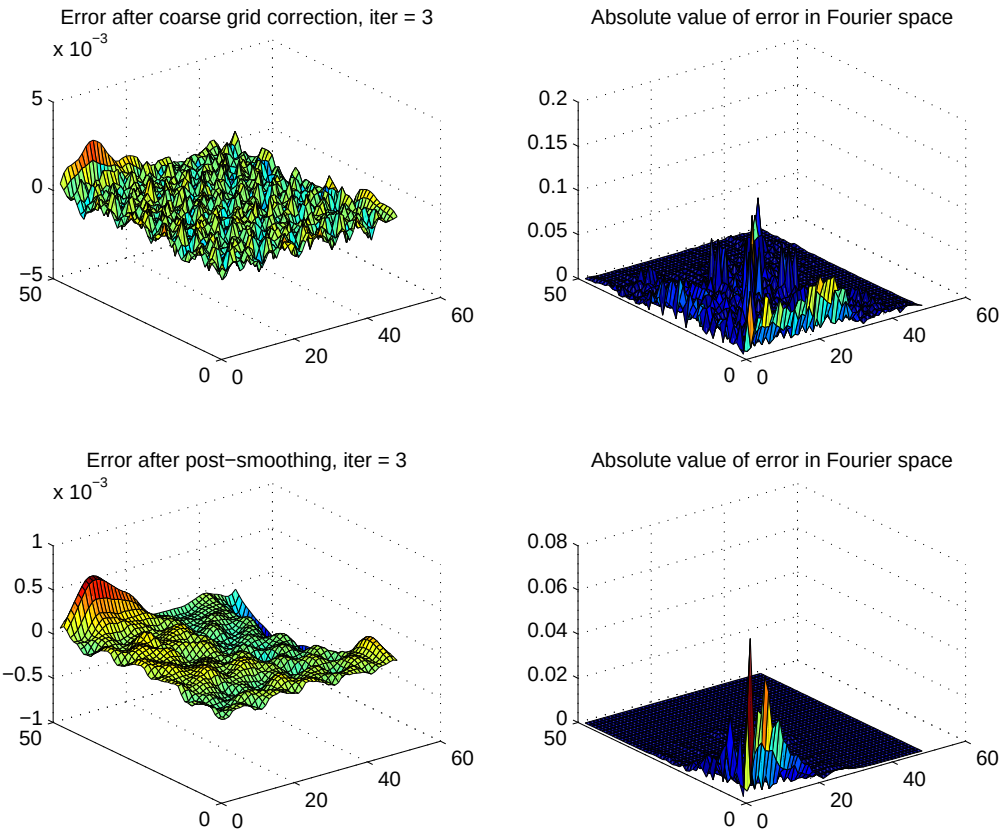

Figure 4.4. The error after 3_{rd} coarse grid correction and post-smoothing with the absolute value of error in Fourier space.

each relaxation step. Thus, the result at i-th step is

$$\beta_i - \sum_{j=1}^{i-1} a_{ij}\xi_j^{k+1} - a_{ii}\xi_i^{k+1} - \sum_{j=i+1}^{n} a_{ij}\xi_j^k = 0, \tag{4.42}$$

which leads to the iteration,

$$\xi_i^{k+1} = \frac{1}{a_{ii}}\left(\beta_i - \sum_{j=1}^{i-1} a_{ij}\xi_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}\xi_j^k\right), \tag{4.43}$$

The equation (4.42) can be rewritten as

$$b + Lx^{k+1} - Dx^{k+1} + Fx^k = 0,$$

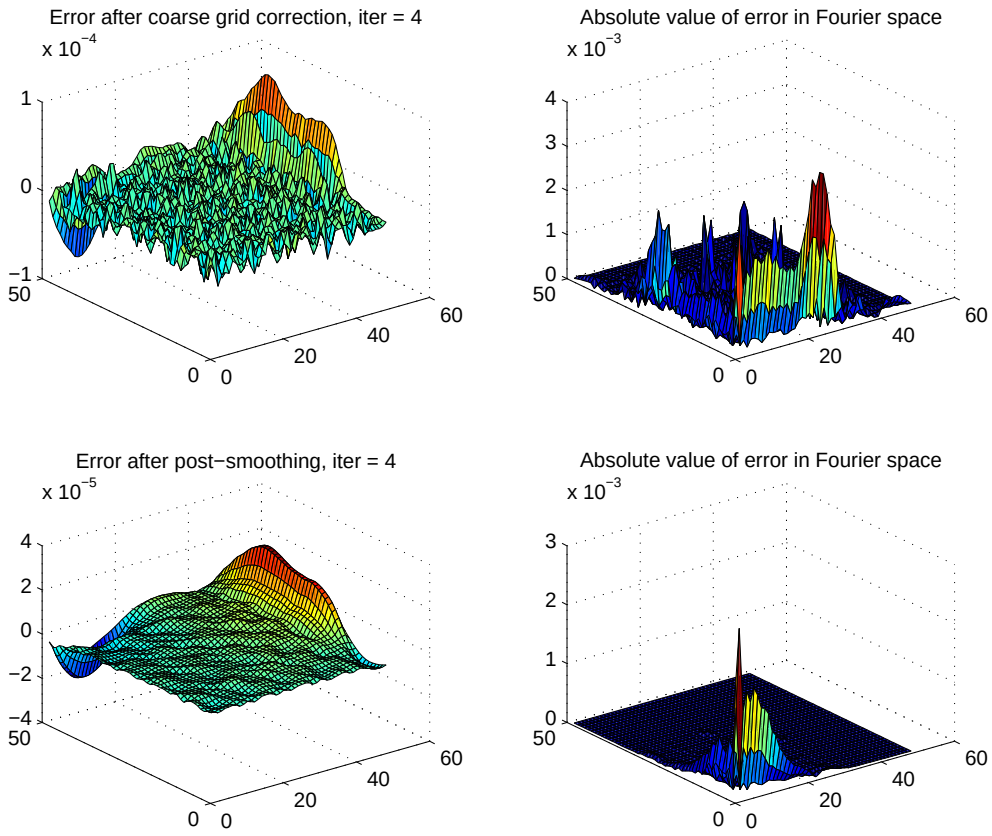

Figure 4.5. The error after 4_{th} coarse grid correction and post-smoothing with the absolute value of error in Fourier space.

So, the vector form of the Gauss-Seidel iteration is obtained as following

$$x^{k+1} = (D - L)^{-1}Ux^k + (D - L)^{-1}b.$$

4.1.3. Incomplete LU Factorizations. Based on the LU decomposition (Figure.4.7) (an upper triangular matrix in U and a "psychologically lower triangular matrix in L, i.e, A is decomposed to a product of a lower triangular and a permutation matrices), another simple different way is to perform an Incomplete LU factorization (Figure.4.8) of the original matrix A. This entails a decomposition of the form

$$A = LU - R \tag{4.44}$$

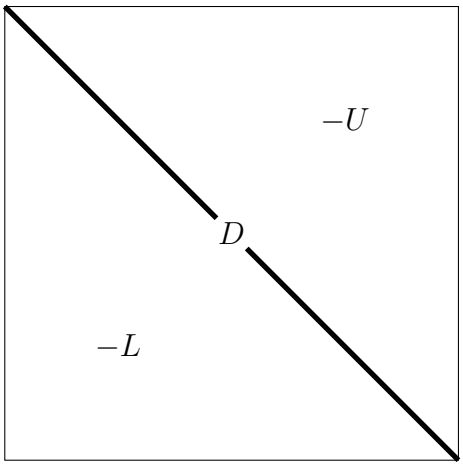

Figure 4.6. The initial decomposition of matrix A.

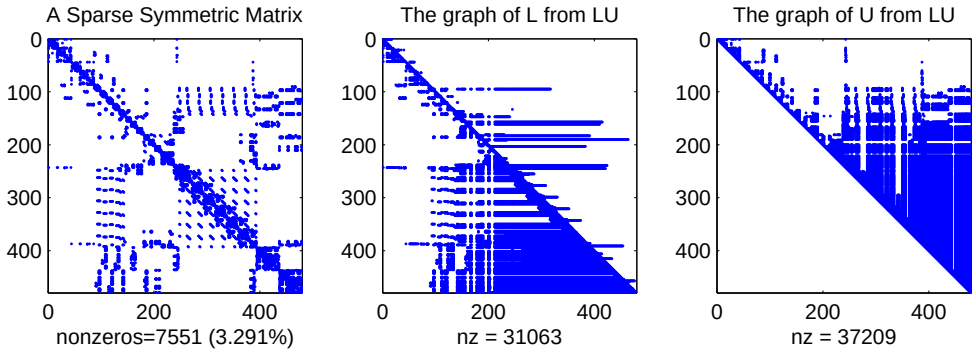

Figure 4.7. The graph of the matrix and the graph of the LU decomposition matrix.

where L and U have the same nonzero structure as the lower and upper parts of A respectively, and R is the residual or error of the factorization.

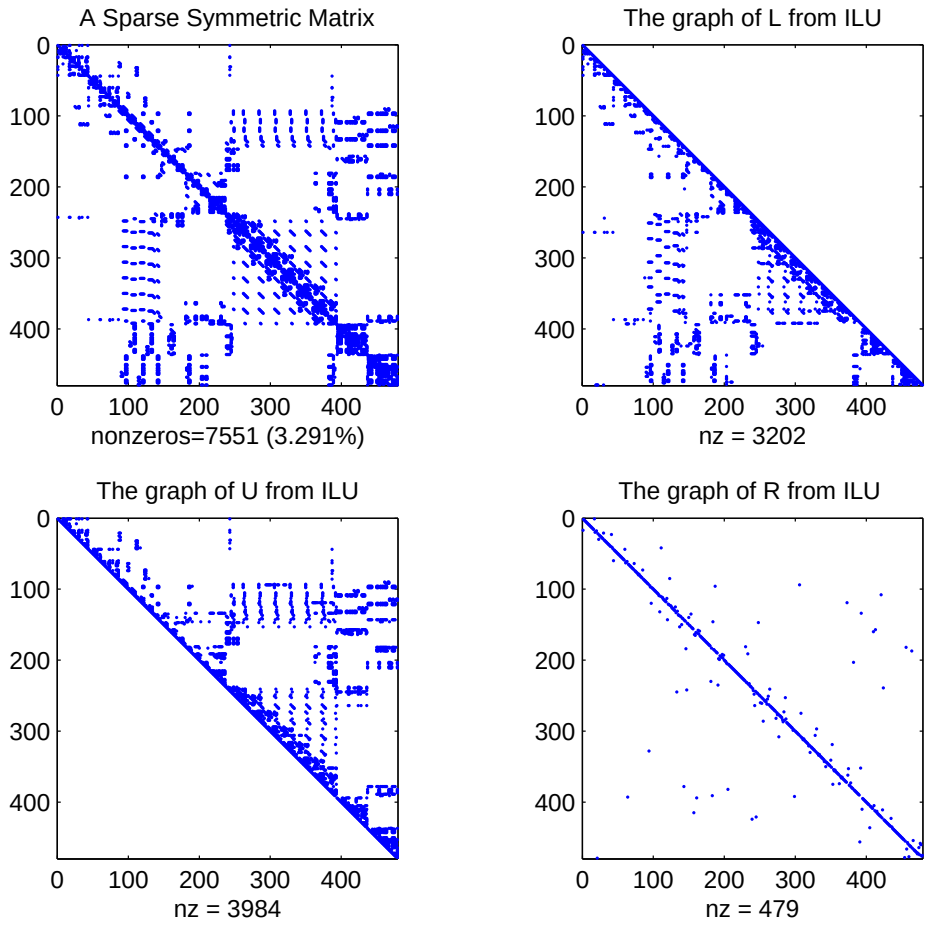

Figure 4.8. The graph of the matrix and the graph of the ILU decomposition matrix.

In general, the ILU factorizations require fewer iterations to converge, but the pre-processing cost for computing the factors is higher.

4.2. THE INTERPOLATION AND RESTRICTION IN 2-DIMENSIONS

Two-grid method requires going back and forth between fine grid Ω^h and coarse grid Ω^H, (in the following we will take $H = 2h$, but other choices are possible) as illustrated in Figure (4.9).

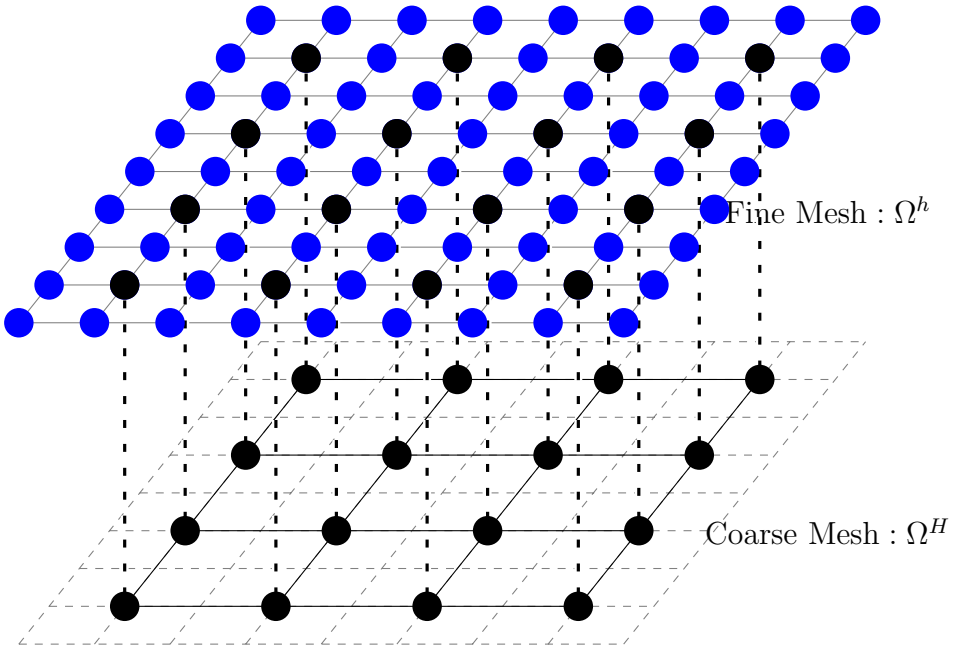

Figure 4.9. Mesh on the fine and coarse grid of the two-grid method.

4.2.1. The Interpolation Operation. The interpolation operation takes a vector from coarse grid Ω^H to fine grid Ω^h.

$$I_H^h : \Omega^H \to \Omega^h.$$

The simplest way to define a interpolation operator is through linear interpolation.

1. **1-D case**

   Let $x_0, x_1, \ldots, x_{n+1}$ be the nodes of 1-D partition, where the x_0, x_{n+1} are boundary points and the number of the internal nodes n is assumed to be odd. Given a vector $(r_i^H)_{i=0,\ldots,(n+1)/2}$, the vector $r^h = I_H^h r^H I_h^H$ of Ω_h is defined as follows

$$\begin{cases} r_{2j}^h = r_j^H \\ r_{2j+1}^h = (r_j^H + r_{j+1}^H)/2 \end{cases} \quad \text{for} \quad j = 0, \ldots, \frac{n+1}{2}. \tag{4.45}$$

The one-dimensional stencil can be denoted by

$$p = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{4.46}$$

2. **2-D case**

In 2-D, the linear interpolation can be defined in an extended manner from the 1-D case. The simplest way to define a prolongation operator is through the bilinear interpolation. Let $I^h_{H,x}$ and $I^h_{H,y}$ denote the interpolation in x and y direction respectively. So the interpolation in x direction can be written as

$$r^{h,x} = I^h_{H,x} r^H,$$

where,

$$\begin{cases} r^{h,x}_{2i,:} = r^H_{i,:} \\ r^{h,x}_{2i+1,:} = (r^H_{i,:} + r^H_{i+1,:})/2 \end{cases} \quad \text{for} \quad i = 0, \ldots, \frac{m+1}{2}. \tag{4.47}$$

Then by using the above semi-interpolated result $r^{h,x}$, we can get the interpolation value with respect to y variable:

$$r^h = I^h_{H,y} r^{h,x},$$

where

$$\begin{cases} r^h_{:,2j} = r^{H,x}_{:,j} \\ r^h_{:,2j+1} = (r^{H,x}_{:,j} + r^{H,x}_{:,j+1})/2 \end{cases} \quad \text{for} \quad j = 0, \ldots, \frac{n+1}{2}. \tag{4.48}$$

Therefore, the (4.47) and (4.48) give the following 2-D interpolation formulas from an element r^H in Ω^H to the corresponding element $r^h = I^h_H r^H I^H_h$ in Ω^h,

$$\begin{cases} r^h_{2i,2j} = r^H_{i,j} \\ r^h_{2i+1,2j} = (r^H_{i,j} + r^H_{i+1,j})/2 \\ r^h_{2i,2j+1} = (r^H_{i,j} + r^H_{i,j+1})/2 \\ r^h_{2i+1,2j+1} = (r^H_{i,j} + r^H_{i+1,j} + r^H_{i,j+1} + r^H_{i+1,j+1})/4 \end{cases} \quad \text{for} \quad \begin{cases} i = 0, \ldots, \dfrac{m+1}{2} \\ j = 0, \ldots, \dfrac{n+1}{2} \end{cases} \tag{4.49}$$

This derivation shows that the 2-D interpolation can be represented as the tensor product of the two one-dimensional interpolation, i.e,

$$I^h_H = I^h_{H,y} \otimes I^h_{H,x}, \tag{4.50}$$

and the stencil for 2-D linear interpolation is

$$p = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.51}$$

4.2.2. The Restriction Operation. The restriction operation is the reverse of interpolation, i.e

$$I^H_h = (I^h_H)^T. \tag{4.52}$$

4.3. TWO-GRID V-CYCLE

When a smoother is applied to a linear system at a fine level, the residual(Figure(4.1))

$$\vec{r}_h = \vec{b}_h - A_h \vec{u}_h$$

obtained at the end of the smoothing step will typically still be large (Figure4.2). However, it will have small components in the space associated with the high-frequency modes. If these components are removed by solving the above system (exactly) at the lower level, then a better approximation should result. Two-grid methods (Figure.4.10) are rarely practical because the coarse-mesh problem may still be too large to be solved exactly. However, they are useful from a theoretical point of view and provide a pathway to the more practical multi-grid method.

Algorithm 1 Two-Grid cycle

Input: Matrix $A_h, \vec{b}_h$, Initial value $\vec{u}_h^0$, Smooth parameters (ν_1, ν_2)
Output: Approximation solution $\vec{u}_h$.
Metode:

1: Pre-smooth: $\vec{u}_h$:=smooth$(A_h, \vec{u}_h, \vec{b}, \nu_1)$
2: Residual: $\vec{r}_h = \vec{b}_h - A_h\vec{u}_h$
3: Coarsening: $\vec{r}_H = I_h^H \vec{r}_h I_H^h, A_H = I_h^H A_h I_H^h$
4: Solve: $A_H \delta^H = \vec{r}_H$
5: Correction: $\vec{u}_h = \vec{u}_h + I_H^h \delta^H I_h^H$
6: Post-smooth: $\vec{u}_h$:=smooth$(A_h, \vec{u}_h, \vec{b}_h, \nu_2)$

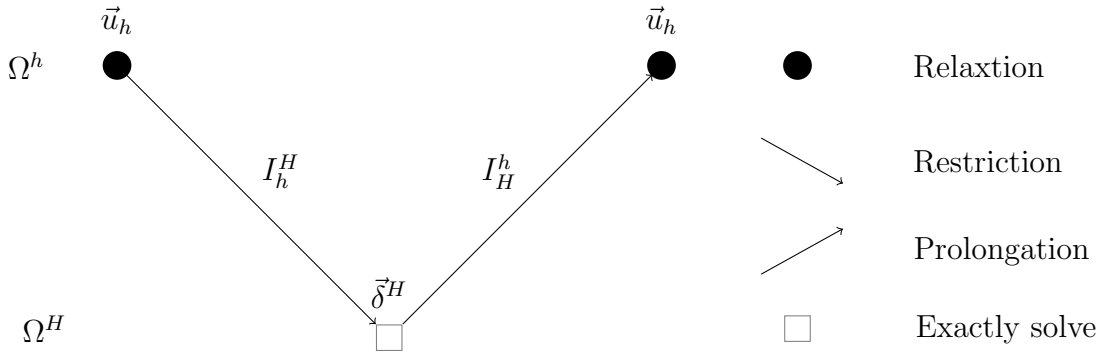

Figure 4.10. Two level method.

In the following subsection, the relationship between the two-gird and multigrid will be investigated.

4.4. FROM TWO-GRID TO MULTIGRID

The recursively defined multigrid cycle is as follows: apply the 2-grid cycle recursively until a coarse enough level is reached and then solve the corresponding system exactly (typically use a direct solver). We now introduce the general multigrid cycle which generalizes the V-cycle mentioned above. This gives the algorithms described below, called the V-cycle multigrid (Algorithm2) and W-cycle multigrid (Algorithm3). Once more, the implementation of the multigrid cycle is of a recursive nature.

4.4.1. V-cycle of Multigrid. A V-cycle multigrid method is obtained when the coarse problem is solved approximately with 1 iteration of the two-grid scheme on that level, and so on, until the coarsest level on which an exact solver is performed as illustrated in Figure 4.11 and Algorithm 2.

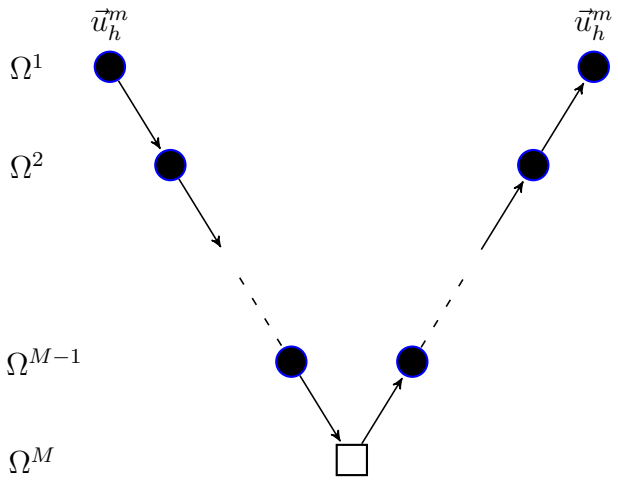

Figure 4.11. V-Cycle of the Multigrid method.

Algorithm 2 The V-cycle Multigrid Algorithm: $\vec{u}_h^m = V - cycle(A_h^m, \vec{u}_0^m, \vec{b}_h^m, \nu_1, \nu_2)$

Input: Matrix $A_h, \vec{u}_h$, Initial value $\vec{u}_0^m$, Smooth parameters (ν_1, ν_2)
Output: Approximation solution $\vec{u}_h^m$.
Metode:

1: Pre-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_0^m, \vec{b}_h^m, \nu_1)$
2: Residual: $\vec{r}^m = \vec{b}_h^m - A_h^m \vec{u}_h^m$
3: Coarsening: $\vec{r}^{m+1} = I_m^{m+1} \vec{r}^m I_{m+1}^m, A_h^{m+1} = I_m^{m+1} A_h^m I_{m+1}^m$
4: **if** (m==M) **then**
5: Solve: $A_h^{m+1} \delta^{m+1} = \vec{r}^{m+1}$
6: **else**
7: Recursion: $\delta^{m+1} = V - cycle(A_h^{m+1}, 0, \vec{r}^{m+1}, \nu_1, \nu_2)$
8: **end if**
9: Correction: $\vec{u}_h^m = \vec{u}_h^m + I_{m+1}^m \delta^{m+1} I_m^{m+1}$
10: Post-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, F^m, \nu_2)$

4.4.2. W-cycle of Multigrid. The W-cycle based on two stationary iterations at each level as illustrated in Figure 4.12 and Algorithm 3.

Algorithm 3 The General Multigrid Algorithm:$\vec{u}_h^m = MG(A_h^m, \vec{u}_0^m, \vec{b}_h^m, \nu_1, \nu_2, \gamma)$

Input: Matrix $A_h, \vec{b}_h$, Initial value $\vec{u}_0^m$, Smooth parameters (ν_1, ν_2),iteration (γ)
Output: Approximation solution $\vec{u}_h^m$.
Metode:

1: pre-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}^m, \nu_1)$
2: Residual: $\vec{r}^m = \vec{b}_h^m - A_h^m \vec{u}_h^m$
3: Coarsening: $\vec{r}^{m+1} = I_m^{m+1} \vec{r}^m I_{m+1}^m, A^{m+1} = I_m^{m+1} A_h^m I_{m+1}^m$
4: **if** m==M **then**
5: Solve: $A_h^{m+1} \delta^{m+1} = \vec{r}^{m+1}$
6: **else**
7: Recursion: $\delta^{m+1} = MG(A_h^{m+1}, 0, \vec{r}^{m+1}, \nu_1, \nu_2, \gamma)$
8: **end if**
9: Correction: $\vec{u}_h^m = \vec{u}_h^m + I_{m+1}^m \delta^{m+1} I_m^{m+1}$
10: Post-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}^m, \nu_2)$

The new parameter,γ, determines how many times MG is iterated. The case $\gamma = 1$ yields the V-cycle multigrid(Figure(4.11)) . The case $\gamma = 2$ is known as the W-cycle multigrid as illustrate in Figure (4.12).

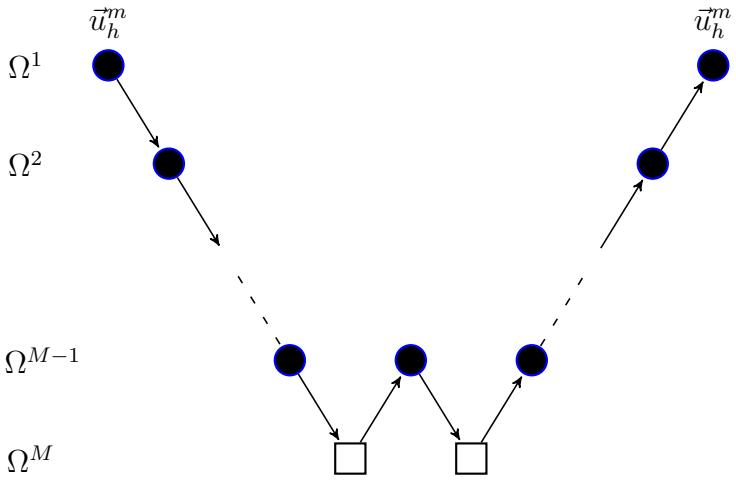

Figure 4.12. W-Cycle of the Multigrid method.

5. THE IFE-AMG ALGORITHM

In the following we will introduce the AMG method [65, 66, 72, 77] that is appropriate for solving the linear system (3.32) which arises from the IFE method. Let $A_h^1 = A_h$, $\vec{u}_h^1 = \vec{u}_h$, $\vec{b}_h^1 = \vec{b}_h$. Then in one V-cycle a sequence of linear systems

$$A_h^m \vec{u}_h^m = \vec{b}_h^m,\ m = 1, \cdots, M,$$

can be generated from different grid levels. Here $A_h^m = \left(a_{ij}^m\right)_{n_m \times n_m}$, $\vec{b}_h = \left(b_i^m\right)_{n_m \times 1}$, $\vec{u}_h = \left(u_i^m\right)_{n_m \times 1}$, and $n = n_1 > n_2 > \cdots > n_m$. Now we discuss two main phases of the AMG algorithm: setup phase and solution phase [65].

5.1. SETUP PHASE OF AMG

5.1.1. Construction of the Coarser Grid. In the setup phase, let Ω^m denote the set of unknowns $u_i^m (1 \le i \le n_m)$ of the m^{th} grid level. And the coarser grid Ω^{m+1} is chosen as a subset of Ω^m, which is denoted as C^m in the m^{th} grid level. The remaining subset $\Omega^m - C^m$ will be denoted by F^m. A point u_i^m is said to be strongly connected to u_j^m, if

$$|a_{ij}^m| \ge \eta \cdot \max_{i \ne j} |a_{ij}^m|, \quad 0 < \eta \le 1. \tag{5.53}$$

Let S_i^m denote the set of all strongly connection points of u_i^m and let the coarse interpolatory set $C_i^m = C^m \cap S_i^m$. In general, C^m and F^m are chosen by the following criteria:

(C1) For each $u_i^m \in F^m$, each point $u_j^m \in S_i^m$ should be either in C_i^m itself or should strongly connected to at least one point in C_i^m;

(C2) C^m should be maximal subset of all points with the property that no two C-points

are strongly connected to each other.

Define the set of points which are strongly connected to u_i^m to be

$$S_i^{m,T} \equiv \{u_j^m : \ u_i^m \in S_j^m\}. \tag{5.54}$$

For a set P, let $|P|$ denote the number of the elements in P. Then Algorithm 4 is proposed by Ruge and Stüben in [66, 70] can be used to chose the coarse grid $\Omega^{m+1} = C^m$ and F^m.

Algorithm 4 The construction of coarse grid

Input: Ω^m.

Output: C^m and F^m.

Method:

$C^m \leftarrow \emptyset, F^m \leftarrow \emptyset, \vec{u}_h^m \leftarrow \Omega^m$ and $\lambda_k^m = |S_k^{m,T}|$ $(1 \le k \le n_m)$
for $(1 \le i \le n_m)$ **do**
  if $(\vec{u}_h^m \neq \emptyset)$ **then**
    Pick the $u_i^m \in \vec{u}_h^m$ such that $\lambda_i^m = \max_{1 \le k \le n_m} \lambda_k^m$, and set $C^m = C^m \cup \{u_i^m\}, \vec{u}_h^m = \vec{u}_h^m - \{u_i^m\}$
    for (all $u_j^m \in S_i^{m,T} \cap \vec{u}_h^m$) **do**
      Set $F^m = F^m \cup \{j\}$ and $\vec{u}_h^m = \vec{u}_h^m - \{j\}$
      for (all $u_l^m \in S_j^m \cap \vec{u}_h^m$) **do**
        set $\lambda_l^m = \lambda_l^m + 1$
      end for
    end for
    for (all $u_j^m \in S_i^m \cap \vec{u}_h^m$) **do**
      set $\lambda_j^m = \lambda_j^m - 1$
    end for
  else
    Stop.
  end if
end for

5.1.2. The Interpolation and Restriction Operators. Once the coarse grid Ω^{m+1} is chosen, the interpolation operators I_{m+1}^m, restriction operators I_m^{m+1} and the coarse grid equation can be constructed as follows. Let $N_i^m = \{u_j^m \in \Omega^m : j \neq i, a_{ij}^m \neq 0\}$

denote the neighborhood of a point $u_i^m \in \Omega^m$, and $D_i^m = N_i^m - C_i^m$. Then the set of the fine grid neighborhood points which are strong connected to u_i^m will be $D_i^{m,s} = D_i^m \cap S_i^m$, and the rest set of the neighborhood points which are weakly connected (non-strong connected) to i will be $D_i^{m,w} = D_i^m - D_i^{m,s}$. Each $u_i^m \in C^m$ can be directly interpolated from the corresponding variable in Ω^{m+1} with unity weight. Each $u_i^m \in F^m$ can be interpolated as a weighted summation of the points in the coarse interpolatory set C_i^m. Assume $u_i^m \in C^m$ is corresponding to $u_{k_i}^{m+1} \in \Omega^{m+1}$. Ruge and Stüben proposed the corresponding interpolation formula [66]:

$$I_{m+1}^m \{u_k^{m+1}\}_{k=1}^{n_{m+1}} = \begin{cases} u_{k_i}^{m+1} & \forall u_i^m \in C^m \\ \sum\limits_{\{j:u_j^m \in C_i^m\}} w_{ij}^m u_{k_j}^{m+1} & \forall u_i^m \in F^m \end{cases} \tag{5.55}$$

where

$$w_{ij}^m = -\frac{1}{a_{ii}^m + \sum\limits_{\{r:u_r^m \in D_i^{m,w}\}} a_{ir}^m} \left[a_{ij}^m + \sum_{\{r:u_r^m \in D_i^{m,s}\}} a_{ir}^m a_{rj}^m \Big/ \sum_{\{l:u_l^m \in C_i^m\}} a_{il}^m \right] \tag{5.56}$$

The Galerkin type method in [66] is a simple approach to define the restriction operator I_m^{m+1}

$$I_m^{m+1} = (I_{m+1}^m)^T \tag{5.57}$$

and

$$A_h^{m+1} = I_m^{m+1} A_h^m I_{m+1}^m, \quad \vec{b}_h^{m+1} = I_m^{m+1} \vec{b}_h^m I_{m+1}^m.$$

5.2. SOLUTION PHASE OF AMG

In the solution phase, the smoothing operator needs to be chosen with proper parameters ν_1 and ν_2, which are the number of the pre-smoothing and post-smoothing steps. In the next chapter, we will investigate the influence of the type of the operator (Gauss-Seidel

and incomplete LU) and these two parameters. Furthermore, we will consider V-cycle only with the maximum number of levels M in this article. Another critical component of AMG is the stopping tolerance, which may have significant effect on the accuracy. Our study in the next chapter shows that the tolerance needs to be small enough for the chosen mesh size. Once all the above components are specified, the recursively defined IFE-AMG algorithm (Algorithm 5) with V-cycle can be proposed in the usual framework as follows [66].

Remark 5.1. *For the moving interface case, the IFE-AMG algorithm proposed above can be utilized to solve the linear system at each time iteration step with*

$$\begin{aligned} A_h &= M_h^{n+\frac{1}{2},n+\frac{1}{2}} + \frac{\tau}{2} A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}}, \\ \vec{b}_h &= \left(M_h^{n+\frac{1}{2},n+\frac{1}{2}} - \frac{\tau}{2} A_h^{n+\frac{1}{2},n+\frac{1}{2},n+\frac{1}{2}} \right) \vec{u}^n + \tau \vec{f}^{n+\frac{1}{2},n+\frac{1}{2}}. \end{aligned}$$

Algorithm 5 The IFE-AMG algorithm of the elliptic interface problem

Input: Model parameters and AMG parameters

Output: IFE-AMG approximation solution $\vec{u}_h$.

Method:

Assemble the matrix from the IFE formulation: $A_h = (a_{ij})_{n\times n}$ with $a_{ij} = \sum_{T\in\mathcal{T}_h} \int_T \beta \nabla\phi_{k_i} \cdot \nabla\phi_{k_j} dX$ where $\phi_{k_i}, \phi_{k_j} \in S_{h,0}^{IFE}(\Omega)$

Assemble the vector from the IFE formulation: $\vec{b}_h = (f_i)_{n\times 1}$ with $f_i = \sum_{T\in\mathcal{T}_h} \int_T \phi_{k_i} f dX - \sum_{s\in\mathcal{I}_h^b} \left(\sum_{T\in\mathcal{T}_h} \int_T \beta \nabla\phi_{k_i} \cdot \nabla\phi_s dX \right) g(X_s)$ where $\phi_{k_i} \in S_{h,0}^{IFE}(\Omega)$ and $\phi_s \in S_h^{IFE}(\Omega)$

relative residual= 1, $\vec{u}_h = 0$

while relative residual>tolerance **do**

    $m = 1$, $A_h^1 = A_h$, $\vec{b}_h^1 = \vec{b}_h$, $\vec{u}_h^1 = \vec{u}_h$

    Call algorithm $MG(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \Omega^m, \nu_1, \nu_2, m)$ as follows:

        Call Algorithm 4 with Ω^m to obtain the C^m and F^m

        Set $\Omega^{m+1} = C^m$

        Define I_{m+1}^m, $I_m^{m+1} = (I_{m+1}^m)^T$

        Pre-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \nu_1)$

        Residual: $\vec{r}_h^m = \vec{b}_h^m - A_h^m \vec{u}_h^m$

        Coarsening: $\vec{r}_h^{m+1} = I_m^{m+1} \vec{r}_h^m I_{m+1}^m$, $A_h^{m+1} = I_m^{m+1} A_h^m I_{m+1}^m$, $\vec{b}_h^{m+1} = I_m^{m+1} \vec{b}_h^m I_{m+1}^m$

        If m==M

            Solve: $A_h^{m+1} \delta^{m+1} = \vec{r}_h^{m+1}$

        Else

            Recursion: $\delta^{m+1} = MG(A_h^{m+1}, 0, \vec{r}_h^{m+1}, \Omega^{m+1}, \nu_1, \nu_2, m+1)$

        EndIf

        Correction: $\vec{u}_h^m = \vec{u}_h^m + I_{m+1}^m \delta^{m+1} I_m^{m+1}$

        Post-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \nu_2)$

    END of MG

    $\vec{u}_h = \vec{u}_h^1$

    relative residual $= \left\| \vec{b}_h - A_h \vec{u}_h \right\| / \left\| \vec{b}_h \right\|$

end while

6. NUMERICAL EXPERIMENTS

In this section, we present numerical examples to illustrate the features of bilinear and linear immersed finite element methods with algebraic multigrid solvers for both the stationary and moving interface problems. We set the initial vector u^0 to be 0 and the strongly connection threshold $\eta = 0.25$. We denote number of V-cycles by V's, the size of the coarsest mesh by N_c, and the stopping tolerance on residual by *tol*. The Gauss-Seidel (GS) and incomplete LU (ILU) iterations are compared as the pre-soothing and post-smoothing operations.

6.1. EXPERIMENTS FOR STEADY INTERFACE PROBLEM

We consider the steady interface problem defined by (1.1)-(1.3) on the typical rectangular domain $\Omega = [-1, 1]\times[-1, 1]$. The interface curve Γ is a circle with radius $r_0 = \pi/6.28$ that separates Ω into two sub-domains Ω^- and Ω^+ with $\Omega^- = \{(x, y) \mid x^2 + y^2 \leq r_0^2\}$. The coefficient function is

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

where $\beta^- = 1$ and $\beta^+ = 10$ are chosen in this example. The boundary condition function $g(x, y)$ and the source term $f(x, y)$ are chosen such that the following function u is the exact solution.

$$u(x, y) = \begin{cases} \dfrac{r^\alpha}{\beta^-}, & \text{if } r \leq r_0, \\ \dfrac{r^\alpha}{\beta^+} + \left(\dfrac{1}{\beta^-} - \dfrac{1}{\beta^+}\right) r_0^\alpha, & \text{otherwise,} \end{cases} \tag{6.58}$$

with $\alpha = 5$, $r = \sqrt{x^2 + y^2}$. We use the bilinear immersed finite elements in this numerical experiment.

6.1.1. The Experiments for the Optimal Convergence Rate. The errors of the IFE-AMG solutions with Gauss-Seidel smoother and various step size are given in Table 6.1 and Table 6.2 . Using linear regression, we can also see that the errors in those table obey the following results:

- Linear regression for Table 6.1

$$\|u - u_h\|_{L^2} \approx 0.4354 h^{2.0104},$$
$$|u - u_h|_{H^1} \approx 0.9115 h^{0.9895}.$$

- Linear regression for Table 6.2

$$\|u - u_h\|_{L^2} \approx 0.4381 h^{2.0124},$$
$$|u - u_h|_{H^1} \approx 0.9115 h^{0.9895}.$$

Table 6.1. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with GS smoother, $tol = 10^{-8}$, and $(\nu_1, \nu_2) = (1, 1)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50028×10^{-4}	9
1/32	26^2	4.10020×10^{-4}	2.94836×10^{-2}	4.85354×10^{-4}	22
1/64	97^2	1.01550×10^{-4}	1.48173×10^{-2}	3.25858×10^{-4}	22
1/128	347^2	2.53035×10^{-5}	7.52028×10^{-3}	1.59749×10^{-4}	45

Table 6.2. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with GS smoother, $tol = 10^{-8}$, and $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50035×10^{-4}	7
1/32	26^2	4.09991×10^{-4}	2.94836×10^{-2}	4.85435×10^{-4}	19
1/64	97^2	1.01487×10^{-4}	1.48173×10^{-2}	3.25996×10^{-4}	19
1/128	347^2	2.51954×10^{-5}	7.52028×10^{-3}	1.60087×10^{-4}	39

The errors of the IFE-AMG solutions with incomplete LU smoother and various step size are given in Table 6.3 and Table 6.4. Using linear regression, we can also see that the errors in those table obey the following results:

- Linear regression for Table 6.3

$$\|u - u_h\|_{L^2} \approx 0.4432 h^{2.0158},$$
$$|u - u_h|_{H^1} \approx 0.9115 h^{0.9895}.$$

- Linear regression for Table 6.4

$$\|u - u_h\|_{L^2} \approx 0.4383 h^{2.0126},$$
$$|u - u_h|_{H^1} \approx 0.9115 h^{0.9895}.$$

Table 6.3. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with ILU smoother, $tol = 10^{-8}$, and $(\nu_1, \nu_2) = (1, 1)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50021×10^{-4}	2
1/32	26^2	4.10047×10^{-4}	2.94836×10^{-2}	4.85274×10^{-4}	3
1/64	97^2	1.01640×10^{-4}	1.48173×10^{-2}	3.25641×10^{-4}	4
1/128	347^2	2.49837×10^{-5}	7.52027×10^{-3}	1.60422×10^{-4}	7

Table 6.4. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with ILU smoother, $tol = 10^{-8}$, and $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50022×10^{-4}	1
1/32	26^2	4.10048×10^{-4}	2.94836×10^{-2}	4.85274×10^{-4}	2
1/64	97^2	1.01303×10^{-4}	1.48173×10^{-2}	3.26107×10^{-4}	2
1/128	347^2	2.51978×10^{-5}	7.52028×10^{-3}	1.59894×10^{-4}	5

These linear regressions indicate that the bilinear IFE-AMG solutions with Gauss-Seidel or incomplete LU smoothers can converge in the optimal rates, which are second order in L^2 norm and first order in H^1 semi-norm.

6.1.2. The Experiments for the Influence of the Smoother on V-cycle. The smoother usually has significant impact on the efficiency and accuracy of the solution. From Table 6.2 and Table 6.4, it can be also easily observed that the incomplete LU smoother significantly reduces the number of V-cycles, which dramatically improve the

efficiency of the IFE-AMG method. Furthermore, from Table 6.5, we can also see that the increase of the number of smoothing steps may decrease the the number of V-cycles while it increases the cost in smoothing phase. Hence the number of smoothing steps needs to be chosen properly in order to balance the total cost.

Table 6.5. Number of V-cycles of the bilinear IFE-AMG solution for the elliptic interface problem with $tol = 10^{-8}$, $(\nu_1, \nu_2) = (2, 2)$.

	GS smoother			ILU smoother		
h	$\#s = 1$	$\#s = 2$	$\#s = 3$	$\#s = 1$	$\#s = 2$	$\#s = 3$
1/16	9	7	6	2	1	1
1/32	22	19	17	3	2	1
1/64	22	19	18	4	2	2
1/128	45	39	36	7	5	4

6.1.3. The Experiments for the Influence of the Tolerance on the Convergence. In the following we will investigate the influence of the tolerance on the convergence of the IFE solutions for the interface problems. From Table 6.2 and Table 6.4, we can see that the bilinear IFE-AMG solutions with both Gauss-Seidel and incomplete LU smoothers converge in the optimal rates when $tol = 10^{-8}$. However, from Table 6.6-6.9, we can see that when $tol = 10^{-5}$, $tol = 10^{-6}$ the bilinear IFE-AMG solutions do not perform optimally any more. This indicates that the tolerance needs to be small enough for the chosen mesh size in order to keep the optimal convergence.

Table 6.6. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with GS smoother and $tol = 10^{-5}$, $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.64796×10^{-3}	5.88162×10^{-2}	9.58376×10^{-4}	4
1/32	26^2	3.77347×10^{-4}	2.94853×10^{-2}	6.18271×10^{-4}	9
1/64	97^2	4.15913×10^{-4}	1.48661×10^{-2}	7.18952×10^{-4}	8
1/128	347^2	6.82136×10^{-4}	7.71096×10^{-3}	8.25904×10^{-4}	15

Table 6.7. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with ILU smoother and $tol = 10^{-5}$, $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50022×10^{-4}	1
1/32	26^2	4.08628×10^{-4}	2.94836×10^{-2}	4.87086×10^{-4}	1
1/64	97^2	3.87171×10^{-4}	1.48596×10^{-2}	5.46570×10^{-4}	1
1/128	347^2	9.83024×10^{-4}	7.90898×10^{-3}	1.17598×10^{-3}	2

Table 6.8. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with GS smoother and $tol = 10^{-6}$, $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65322×10^{-3}	5.88161×10^{-2}	9.51021×10^{-4}	5
1/32	26^2	4.03984×10^{-4}	2.94836×10^{-2}	5.03028×10^{-4}	12
1/64	97^2	9.35785×10^{-5}	1.48173×10^{-2}	3.56913×10^{-4}	12
1/128	347^2	6.39989×10^{-5}	7.52157×10^{-3}	2.28196×10^{-4}	36

Table 6.9. Errors of the bilinear IFE-AMG solution for the elliptic interface problem with ILU smoother and $tol = 10^{-6}$, $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	5^2	1.65383×10^{-3}	5.88161×10^{-2}	9.50022×10^{-4}	1
1/32	26^2	4.08628×10^{-4}	2.94836×10^{-2}	4.87086×10^{-4}	1
1/64	97^2	1.01303×10^{-4}	1.48173×10^{-2}	3.26107×10^{-4}	2
1/128	347^2	7.94820×10^{-5}	7.52023×10^{-3}	2.26107×10^{-4}	3

In the last experiments, the results show that the stop tolerance will affect the convergence rate of the IFE solutions for the interface problems.

6.2. EXPERIMENTS FOR THE MOVING INTERFACE PROBLEM

We consider the moving interface problem defined by (1.4)-(1.6) on $\Omega \times [0, T_{end}]$, where $\Omega = (-1, 1) \times (-1, 1)$ and $T_{end} = 1$. The interface $\Gamma(t)$ is a moving circle centered at origin with radius $r(t)$ which separates Ω into two sub-domains $\Omega^-(t) = \{(x, y) \in \Omega : x^2 + y^2 < r(t)^2\}$, and $\Omega^+(t) = \{(x, y) \in \Omega : x^2 + y^2 > r(t)^2\}$. Let $\beta^- = 1$ and $\beta^+ = 10$.

The exact solution is chosen as:

$$u(t,x,y) = \begin{cases} \frac{r^\alpha}{\beta^-}\cos(t), & r(t) \in \Omega^-(t), \\ \frac{r^\alpha}{\beta^+}\cos(t) + \left(\frac{1}{\beta^-} - \frac{1}{\beta^+}\right) r(t)^\alpha \cos(t), & r(t) \in \Omega^+(t). \end{cases} \tag{6.59}$$

In all the numerical examples presented below, the radius change is governed by $r(t) = r_0\left(\frac{\sin(t)+3}{4}\right)$ with $r_0 = \frac{\pi}{6.28}$, and we use triangular Cartesian meshes $\mathcal{T}_h$ which are formed by partitioning Ω with $N_s \times N_s$ rectangles of size $h = 2/N_s$ and then cutting each rectangle into two triangles alone one of its diagonal line. For time discretization, we denote its step size by τ and define $t_n = n\tau$, with $n = 1, 2, \cdots, N$. We use the linear immersed finite elements in this numerical experiment.

The errors of the IFE-AMG solutions with Gauss-Seidel smoother and various step size are given in Table 6.10. Using linear regression, we can also see that the errors in this table obey

$$\|u - u_h\|_{L^2} \approx 0.6882h^{1.9381},$$
$$|u - u_h|_{H^1} \approx 0.6709h^{0.9234}.$$

The errors of the IFE-AMG solutions with incomplete LU smoother and various step size are given in Table 6.11. Using linear regression, we can also see that the errors in this table obey

$$\|u - u_h\|_{L^2} \approx 0.6883h^{1.9382},$$
$$|u - u_h|_{H^1} \approx 0.6709h^{0.9234}.$$

Table 6.10. Errors of the bilinear IFE-AMG solution for the moving interface problem with GS smoother, $tol = 10^{-8}$, and $(\nu_1, \nu_2) = (2, 2)$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	7^2	3.2659×10^{-3}	5.2764×10^{-2}	1.5801×10^{-3}	6
1/32	15^2	8.1519×10^{-4}	2.6920×10^{-2}	9.2506×10^{-4}	7
1/64	77^2	2.1175×10^{-4}	1.4116×10^{-2}	4.5706×10^{-4}	14
1/128	263^2	5.8132×10^{-5}	7.7486×10^{-3}	2.5078×10^{-4}	18

Table 6.11. Errors of the bilinear IFE-AMG solution for the moving interface problem with ILU smoother and $tol = 10^{-8}$.

h	N_c	$\|u - u_h\|_{L^2}$	$\|u - u_h\|_{H^1}$	$\|u - u_h\|_{l^\infty}$	V's
1/16	7^2	3.2659×10^{-3}	5.2764×10^{-2}	1.5801×10^{-3}	1
1/32	15^2	8.1519×10^{-4}	2.6920×10^{-2}	9.2506×10^{-4}	1
1/64	77^2	2.1175×10^{-4}	1.4116×10^{-2}	4.5711×10^{-4}	1
1/128	263^2	5.8122×10^{-5}	7.7486×10^{-3}	2.5078×10^{-4}	2

These linear regressions indicate that the bilinear IFE-AMG solutions with Gauss-Seidel or incomplete LU smoothers can converge in the optimal rates, which are second order in L^2 norm and first order in H^1 semi-norm. From Table 6.10 and Table 6.11, it can be also easily observed that the incomplete LU smoother significantly reduces the number of V-cycles, which dramatically improve the efficiency of the IFE-AMG method.

It is also clearly showed in Fig 6.1 that the algebraic multigrid solvers are stable and efficient for the linear systems arising from the IFE methods since the residual error

quickly decreases to a small magnitude.

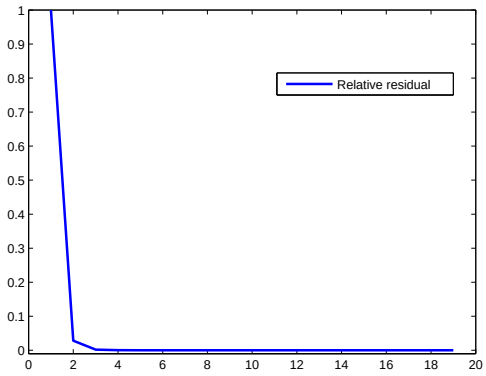

Figure 6.1. Residual after each iteration of V-cycle at $t = 1$ when GS smoother is used.

Furthermore, Tables 6.12 and 6.13 provide the number of V-cycles of the linear IFE-AMG solution at different time steps when the interface locations are different. We can observe that the number of V-cycles for Gauss-Seidel smoother depend on the moving interface locations but not very severely. The incomplete LU smoother can reduce the dependence of the number of V-cycles on the moving interface since it needs much less number of V-cycles than the Gauss-Seidel smoother.

Table 6.12. Number of V-cycles of the linear IFE-AMG solution at different time steps for the moving interface problem with GS smoother and $tol = 10^{-8}$.

h	$t = \triangle t$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
1/32	8	8	13	8	7
1/64	8	8	12	8	14
1/128	9	13	19	15	18

Table 6.13. Number of V-cycles of the linear IFE-AMG solution at different time steps for the moving interface problem with ILU smoother and $tol = 10^{-8}$.

h	$t = \triangle t$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
1/32	1	1	1	1	1
1/64	1	1	1	1	1
1/128	1	2	2	2	2

7. CONCLUSIONS AND FUTURE WORK

7.1. CONCLUSIONS

In this thesis, we discussed the bilinear and linear immersed finite element (IFE) solutions from the algebraic multigrid solver for both stationary and moving interface problems. The feature of the symmetric positive-definite matrices from the IFE methods naturally matches the corresponding need of the algebraic multigrid solver in order to guarantee its efficiency. Furthermore, the combination of the other features of the algebraic multigrid method and the IFE methods, such as the preconditioning property, independence of geometry, optimal convergence rates, flexibility to handle the interface on structured meshes instead of body-fitting meshes, can dramatically improve the efficiency of the proposed methods when the IFE-AMG method is applied to real world applications. The numerical experiments are performed to demonstrate these features as well as the influence of the tolerance and the smoother on the efficiency and convergence.

7.2. FUTURE WORK: IFE-AGMG SOLVER

We will investigate the immersed finite elements with aggregation-based algebraic multigrid method (IFE-AGMG) in the future. The AGMG[62] and the AMG is the algorithm for coarsening.In this section, we will recall the aggregation-based coarsening algorithm. More Detailed descriptions may ne found in[2],[62] and [1].

Coarsening by aggregation works differently from the classical AMG. It needs to define *aggregates* G_i, which are some disjoint subsets of the set of variables. The number of coarse variables n_c is the number of such subsets, and $A_c = P^T AP$, where P is given by

$$P_{ij} = \begin{cases} 1 & \text{if} \quad i \in G_j \\ 0 & \text{otherwise} \end{cases} \qquad (1 \leq i \leq n, 1 \leq j \leq n_c). \tag{7.60}$$

If $\cup_i G_i = [1, n]$, P is a Boolean matrix with exactly one nonzero entry per row. Details are given in Algorithm (6) and Algorithm (7). Similar to the classical AMG coarsening, the AGMG is also based on the strong negative couplings. One defines the set of nodes S_i to which i is strongly negatively coupled, using the Strong/Weak coupling threshold β:

$$S_i = \{j \neq i | a_{ij} < -\beta \max_{a_{ik}<0} |a_{ik}|\}. \tag{7.61}$$

Then, one picks up an unmarked node i at a time, giving the priority to nodes with minimal m_i, where m_i is the number of unmarked nodes that are strongly negatively coupled to i.

Algorithm 6 Pairwise aggregation

Input: Matrix $A = (a_{ij})$ with n rows, Strong/ Weak coupling threshold β, Logical parameter $CheckDD$

Output: Number of coarse variables n_c and subset (aggregates) G_i, $i = 1 \ldots, n_c$,(such that $G_i \cap G_j = \emptyset$ for $i \neq j$)

Metode:

if $(CheckDD)$ **then**
  $U = [1, n] \backslash \{i | a_{ii} > 5 \sum_{j \neq i} |a_{ij}|\}$
else
  $U = [1, n]$
end if
for (all i) **do**
  $S_i = \{j \in U \backslash \{i\} | a_{ij} < -\beta max_{a_{ik}<0} |a_{ik}|\}$, $m_i = |\{j | i \in S_j\}|$; $n_c = 0$
end for
while $U \neq \emptyset$ **do**
  Select $i \in U$ with minimal $m_i; n_c = n_c + 1$
  Select $j \in U$ such that $a_{ij} = min_{k \in U} a_{ik}$
  if $(j \in S_i)$ **then**
    $G_{n_c} = i, j$
  else
    $G_{n_c} = i$
  end if
  $U = U \backslash G_{n_c}$
  for $(k \in G_{n_c}, l \in S_k)$ **do**
    $_l = m_l - 1$
  end for
end while

Algorithm 7 Double pairwise aggregation

Input: Matrix $A = (a_{ij})$ with n rows, Strong/ Weak coupling threshold β, Logical parameter $CkDD$

Output: Number of coarse variables n_c and subset (aggregates) G_i, $i = 1 \ldots, n_c$,(such that $G_i \cap G_j = \emptyset$ for $i \neq j$)

Metode:

1: Apply Algorithm (6) to A with threshold β and $CheckDD = CkDD$
2: Output: n_{c_1}, and $G_i^{(1)}$, $i = 1, \ldots, n_{c_1}$
3: Compute the $n_{c_1} \times n_{c_1}$ auxiliary matrix $A_1 = (a_{ij}^{(1)})$ with
4: $a_{ij}^{(1)} = \sum_{k \in G_i^{(1)}} \sum_{l \in G_j^{(1)}} a_{kl}$
5: Apply Algorithm (6) to A_1 with threshold β and $CheckDD = false$
6: n_{c_1}, and $G_i^{(2)}$, $i = 1, \ldots, n_{c_1}$
7: For $i = 1, \ldots, n_c : G_i = \cup_{j \in G_i^{(2)}} G_j^{(1)}$

Once the five components $\Omega^m, I_m^{m+1}, I_{m+1}^m, A^m$ and G^m (the smoothing operator) are defined, the recursively defined IFE-AGMG cycle is as algorithm (8). We will investigate the properties of IFE-AGMG and then compare them with the IFE-AMG's in the future.

Algorithm 8 The Immersed Finite Element with Aggregation-based Algebraic Multigrid Algorithm

Input: Matrix A, F, Initial value U_h^0 , Smooth parameters (ν_1, ν_2)
Output: Approximation solution $\tilde{U}_h$.
Metode:

1: Assemble the matrix from the IFE formulation: $A_h = (a_{ij})_{n\times n}$ with $a_{ij} = \sum_{T\in\mathcal{T}_h} \int_T \beta \nabla\phi_{k_i} \cdot \nabla\phi_{k_j} dX$ where $\phi_{k_i}, \phi_{k_j} \in S_{h,0}^{IFE}(\Omega)$
2: Assemble the vector from the IFE formulation: $\vec{b}_h = (f_i)_{n\times 1}$ with $f_i = \sum_{T\in\mathcal{T}_h} \int_T \phi_{k_i} f dX - \sum_{s\in\mathcal{I}_h^b} \left(\sum_{T\in\mathcal{T}_h} \int_T \beta \nabla\phi_{k_i} \cdot \nabla\phi_s dX \right) g(X_s)$ where $\phi_{k_i} \in S_{h,0}^{IFE}(\Omega)$ and $\phi_s \in S_h^{IFE}(\Omega)$
3: relative residual= 1, $\vec{u}_h = 0$
4: **while** relative residual>tolerance **do**
5: $m = 1$, $A_h^1 = A_h$, $\vec{b}_h^1 = \vec{b}_h$, $\vec{u}_h^1 = \vec{u}_h$
6: Call algorithm $MG(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \Omega^m, \nu_1, \nu_2, m)$ as follows:
7: Call Algorithm 7 with Ω^m to obtain the C^m and F^m
8: Set $\Omega^{m+1} = C^m$
9: Define I_{m+1}^m, $I_m^{m+1} = (I_{m+1}^m)^T$
10: Pre-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \nu_1)$
11: Residual: $\vec{r}_h^m = \vec{b}_h^m - A_h^m \vec{u}_h^m$
12: Coarsening: $\vec{r}_h^{m+1} = I_m^{m+1} \vec{r}_h^m I_{m+1}^m$, $A_h^{m+1} = I_m^{m+1} A_h^m I_{m+1}^m$, $\vec{b}_h^{m+1} = I_m^{m+1} \vec{b}_h^m I_{m+1}^m$
13: If m==M
14: Solve: $A_h^{m+1} \delta^{m+1} = \vec{r}_h^{m+1}$
15: Else
16: Recursion: $\delta^{m+1} = MG(A_h^{m+1}, 0, \vec{r}_h^{m+1}, \Omega^{m+1}, \nu_1, \nu_2, m+1)$
17: EndIf
18: Correction: $\vec{u}_h^m = \vec{u}_h^m + I_{m+1}^m \delta^{m+1} I_m^{m+1}$
19: Post-smooth: $\vec{u}_h^m$:=smooth$(A_h^m, \vec{u}_h^m, \vec{b}_h^m, \nu_2)$
20: END of MG
21: $\vec{u}_h = \vec{u}_h^1$
22: relative residual $= \left\| \vec{b}_h - A_h \vec{u}_h \right\| / \left\| \vec{b}_h \right\|$
23: **end while**

BIBLIOGRAPHY

[1] Y. Notay A. Napov. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM J. Sci. Comput.*, 34:A2288–A2316, 2012.

[2] Y. Notay A. Napov. An algebraic multigrid method with guaranteed convergence rate. *SIAM J. Sci. Comput.*, 34:A1079–A1109, 2012.

[3] S. Adjerid and T. Lin. Higher-order immersed discontinuous Galerkin methods. *Int. J. Inf. Syst. Sci.*, 3(4):555–568, 2007.

[4] S. Adjerid and T. Lin. p-th degree immersed finite element for boundary value problems with discontinuous coefficients. *Appl. Numer. Math*, 59(6):1303–1321, 2009.

[5] A. Almgre, J. B. Bell, P. Collela, and T. Marthaler. A cartesian grid emthod for imcompressible Euler equations in complex geometries. *SIAM Journal on Scientific Computing*, 18:1289–1390, 1997.

[6] Y. Arakawa and M. Nakano. An efficient three-dimensional optics code for ion thruster research. In *AIAA-3196*, 1996.

[7] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5:207–213, 1970.

[8] I. Babuška and J. E. Osborn. Can a finite element method perform arbitrarily badly? *Math. Comp.*, 69(230):443–462, 2000.

[9] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in optimal design using a homogenization method. *Comp. Meth. Appl. Mech. Engng.*, 71:197–224, 1988.

[10] I. Boyd, D. VanGilde, and X. Liu. Monte carlo simulation of neutral xenon flows in electric propulsion devices. *J. Propulsion and Power*, 14(6):1009–1015, 1998.

[11] J. H. Bramble and J. T. King. A finite element method for interface problems in domains with smooth boundary and interfaces. *Adv. Comput. Math.*, 6:109–138, 1996.

[12] B. Camp, T. Lin, Y. Lin, and W. Sun. Quadratic immersed finite element spaces and their approximation capabilities. *Adv. Comput. Math.*, 24(1-4):81–112, 2006.

[13] J. R. Cannon, Jr. Douglas, Jim, and C. Denson Hill. A multi-boundary stefan problem and the disappearance of phases. *J. Math. Mech.*, 17:21–33, 1967.

[14] Z. Chen and J. Zou. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numer. Math.*, 79:175–202, 1998.

[15] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys*, 2:12, 1967.

[16] S. Chou, D. Y. Kwak, and K. T. Wee. Optimal convergence analysis of an immersed interface finite element method. *Adv. Comput. Math.*, 33(2):149–168, 2010.

[17] S. H. Chou. An immersed linear finite element method with interface flux capturing recovery. *Discrete Contin. Dyn. Syst. Ser. B*, 17(7):2343–2357, 2012.

[18] Y. Chu, Y. Cao, X.-M. He, and M. Luo. Asymptotic boundary conditions for two-dimensional electrostatic field problems with immersed finite elements. *Comput. Phys. Commun.*, 182(11):2331–2338, 2011.

[19] A. Dadone and B. Grossman. Design optimization of fluid dynamic problems using cartesian grids. In K. Srinivas S. Armfield, P. Morgan, editor, *Computational Fluid Dynamics 2002*, Verlin, 591-596. Springer.

[20] R. E. Ewing, Z. Li, T. Lin, and Y. Lin. The immersed finite volume element methods for the elliptic interface problems. Modelling '98 (prague). *Math. Comput. Simulation*, 50(1-4):63–76, 1999.

[21] X. Feng and Z. Li. Simplified immersed interface methods for elliptic interface problems with straight interfaces. *Numer. Methods Partial Differential Equations*, 28(1):188–203, 2012.

[22] X. Feng, Z. Li, and L. Wang. Analysis and numerical methods for some crack problems. *Int. J. Numer. Anal. Model. Ser. B*, 2(2-3):155–166, 2011.

[23] A. L. Fogelson and J. P. Keener. Immersed interface methods for Neumann and related problems in two and three dimensions. *SIAM J. Sci. Comput.*, 22:1630–1654, 2001.

[24] W. Geng and G. W. Wei. Multiscale molecular dynamics using the matched interface and boundary method. *J. Comput. Phys.*, 230(2):435–457, 2011.

[25] B. H. Gilding. Qualitative mathematical analysis of the Richards equation. *Transport Porous Med.*, 6(5-6):651–666, 1991.

[26] Y. Gong, B. Li, and Z. Li. Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions. *SIAM J. Numer. Anal.*, 46:472–495, 2008.

[27] Y. Gong and Z. Li. Immersed interface finite element methods for elasticity interface problems with non-homogeneous jump conditions. *Numer. Math. Theory Methods Appl.*, 3(1):23–39, 2010.

[28] B. E. Griffith and S. Lim. Simulating an elastic ring with bend and twist by an adaptive generalized immersed boundary method. *Commun. Comput. Phys.*, 12(2):433–461, 2012.

[29] R. D. Guy and B. Philip. A multigrid method for a model of the implicit immersed boundary equations. *Commun. Comput. Phys.*, 12(2):378–400, 2012.

[30] X.-M. He. Bilinear immersed finite elements for interface problems. *Ph.D. dissertation, Virginia Polytechnic Institute and State University*, 2009.

[31] X.-M. He, T. Lin, and Y. Lin. Approximation capability of a bilinear immersed finite element space. *Numer. Methods Partial Differential Equations*, 24(5):1265–1300, 2008.

[32] X.-M. He, T. Lin, and Y. Lin. A bilinear immersed finite volume element method for the diffusion equation with discontinuous coefficients. *Commun. Comput. Phys.*, 6(1):185–202, 2009.

[33] X.-M. He, T. Lin, and Y. Lin. Interior penalty discontinuous Galerkin methods with bilinear IFE for a second order elliptic equation with discontinuous coefficient, dedicated to Professor David Russells 70th birthday. *J. Syst. Sci. Complex.*, 23(3):467–483, 2010.

[34] X.-M. He, T. Lin, and Y. Lin. Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *Int. J. Numer. Anal. Model.*, 8(2):284–301, 2011.

[35] X.-M. He, T. Lin, and Y. Lin. The convergence of the bilinear and linear immersed finite element solutions to interface problems. *Numer. Methods Partial Differential Equations*, 28(1):312–330, 2012.

[36] X.-M. He, T. Lin, Y. Lin, and X. Zhang. Immersed finite element methods for parabolic equations with moving interface. *Numer. Methods Partial Differential Equations*, 29(2):619–646, 2013.

[37] B. Heinrich. *Finite difference methods on irregular networks*, volume 82 of *International Series of Numerical Mathematics*. Birkhäuser, Boston, 1987.

[38] D. W. Hewitt. The embedded curved boundary method for orthogonal simulation meshes. *J. Comput. Phys.*, 138:585–616, 1997.

[39] D. M. Ingram, D. M. Causon, and C. G. Mingham. Developments in Cartesian cut cell methods. *Math. Comput. Simulation*, 61(3-6):561–572, 2003.

[40] H. Ji, F.-S. Lien, and E. Yee. An efficient second-order accurate cut-cell method for solving the variable coeffcient Poisson equation with jump conditions on irregular domains. *Int. J. Numer. Methods Fluids*, 52:723–748, 2006.

[41] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 147:60–85, 1998.

[42] R. Kafafy, T. Lin, Y. Lin, and J. Wang. Three-dimensional immersed finite element methods for electric field simulation in composite materials. *Int. J. Numer. Meth. Engrg.*, 64(7):940–972, 2005.

[43] R. Kafafy, J. Wang, and T. Lin. A hybrid-grid immersed-finite-element particle-in-cell simulation model of ion optics plasma dynamics. *Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms*, 12:1–16, 2005.

[44] Y. Kim, Y. Seol, M. Lai, and C. S. Peskin. The immersed boundary method for two-dimensional foam with topological changes. *Commun. Comput. Phys.*, 12(2):479–493, 2012.

[45] D. Y. Kwak, K. T. Wee, and K. S. Chang. An analysis of a broken p_1-nonconforming finite element method for interface problems. *SIAM J. Numer. Anal.*, 48(6):2117–2134, 2010.

[46] T. P. Chartier L. Adams. New geometric immersed interface multigrid solvers. *SIAM J. Sci. Comput*, 25(5):1516–1533, 2004.

[47] T. P. Chartier L. Adams. A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems. *SIAM J. Sci. Comput*, 26(3):762–784, 2005.

[48] Z. Li L. Adams. The immersed interface/multigrid methods for interface problems. *SIAM J. Sci. Comput*, 24(2):463–479, 2002.

[49] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 34:1019–1044, 1994.

[50] Z. Li. The immersed interface method using a finite element formulation. *Appl. Numer. Math.*, 27(3):253–267, 1997.

[51] Z. Li and K. Ito. *The immersed interface method: Numerical solutions of PDEs involving interfaces and irregular domains*, volume 33 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[52] Z. Li, T. Lin, Y. Lin, and R. C. Rogers. An immersed finite element space and its approximation capability. *Numer. Methods Partial Differential Equations*, 20(3):338–367, 2004.

[53] Z. Li, T. Lin, and X. Wu. New Cartesian grid methods for interface problems using the finite element formulation. *Numer. Math.*, 96(1):61–98, 2003.

[54] Z. Li and P. Song. An adaptive mesh refinement strategy for immersed boundary/interface methods. *Commun. Comput. Phys.*, 12(2):515–527, 2012.

[55] T. Lin, Y. Lin, R. C. Rogers, and L. M. Ryan. A rectangular immersed finite element method for interface problems. In P. Minev and Y. Lin, editors, *Advances in Computation: Theory and Practice, Vol. 7*, pages 107–114. Nova Science Publishers, Inc., 2001.

[56] T. Lin, Y. Lin, and W. Sun. Error estimation of a class of quadratic immersed finite element methods for elliptic interface problems. *Discrete Contin. Dyn. Syst. Ser. B*, 7(4):807–823, 2007.

[57] T. Lin, Y. Lin, W. Sun, and Z. Wang. Immersed finite element methods for 4th order differential equations. *J. Comput. Appl. Math.*, 235(13):3953–3964, 2011.

[58] T. Lin and D. Sheen. The immersed finite element method for parabolic problems with the Laplace transformation in time discretization. *Int. J. Numer. Anal. Model.*, 10(2):298–313, 2013.

[59] T. Lin and J. Wang. An immersed finite element electric field solver for ion optics modeling. In *Proceedings of AIAA Joint Propulsion Conference, Indianapolis, IN, July, 2002*. AIAA, 2002-4263.

[60] T. Lin and J. Wang. The immersed finite element method for plasma particle simulation. In *Proceedings of AIAA Aerospace Sciences Meeting, Reno, NV, Jan., 2003.* AIAA, 2003-0842.

[61] Gunter H. Meyer. Multidimensional stefan problems. *SIAM J. Numer. Anal.*, 10:522–538, 1973.

[62] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.

[63] C. S. Peskin. Flow patterns around heart valves. *J. Comput. Phys.*, 10:252–271, 1972.

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

[65] H. Fu Q. Chang, Y. Wong. On the algebraic multigrid method. *Journal of Computational Physics*, 125:279–292, 1996.

[66] J. W. Ruge and K. Stüben. "Algebraic multigrid". In S. F. McCormick, editor, multigrid methods. *SIAM, Philadelphia*, 4:73–130, 1987.

[67] Yousef Saad. *Iteretive Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, 1996.

[68] Yousef Saad. *Iterative Methods for Sparse Linear Systems 2nd edition.* Society for Industrial and Applied Mathematics, 2003.

[69] S. A. Sauter and R. Warnke. Composite finite elements for elliptic boundary value problems with discontinuous coefficients. *Computing*, 77(1):29–55, 2006.

[70] K. Stüben. Algebraic multigrid (AMG): experiences and comparisons. *Applied Mathematics and Computation*, 13:419–451, 1983.

[71] Z. Tan, D. V. Le, K. M. Lim, and B. C. Khoo. An immersed interface method for the simulation of inextensible interfaces in viscous fluids. *Commun. Comput. Phys.*, 11(3):925–950, 2012.

[72] C. Oosterlee U. Trottenberg and A.schuller. *Multigrid*, volume 631. Academic Press, London, 2001.

[73] S. Vallaghè and T. Papadopoulo. A trilinear immersed finite element method for solving the electroencephalography forward problem. *SIAM J. Sci. Comput.*, 32(4):2379–2394, 2010.

[74] J. Wang, X.-M. He, and Y. Cao. Modeling spacecraft charging and charged dust particle interactions on lunar surface. *Proceedings of the 10th Spacecraft Charging Technology Conference, Biarritz, France*, 2007.

[75] J. Wang, X.-M. He, and Y. Cao. Modeling electrostatic levitation of dusts on lunar surface. *IEEE Trans. Plasma Sci.*, 36(5):2459–2466, 2008.

[76] K. Wang, H. Wang, and X. Yu. An immersed eulerian-lagrangian localized adjoint method for transient advection-diffusion equations with interfaces. *Int. J. Numer. Anal. Model.*, 9(1):29–42, 2012.

[77] Van Emden Henson William L. Briggs and Steve F. McCormick. *A Multigrid Tutorial, Second Edition*. Society for Industrial and Applied Mathematics, 2000.

[78] C. Wu, Z. Li, and M. Lai. Adaptive mesh refinement for elliptic interface problems using the non-conforming immersed finite element method. *Int. J. Numer. Anal. Model.*, 8(3):466–483, 2011.

[79] H. Xie, Z. Li, and Z. Qiao. A finite element method for elasticity interface problems with locally modified triangulations. *Int. J. Numer. Anal. Model.*, 8(2):189–200, 2011.

[80] S. Xu. An iterative two-fluid pressure solver based on the immersed interface method. *Commun. Comput. Phys.*, 12(2):528–543, 2012.

[81] M. Ye, R. Khaleel, and T. J. Yeh. Stochastic analysis of moisture plume dynamics of a field injection experiment. *Water Resour. Res.*, 41:W03013, 2005.

[82] S. Zhao. High order matched interface and boundary methods for the Helmholtz equation in media with arbitrarily curved interfaces. *J. Comput. Phys.*, 229(9):3155–3170, 2010.

[83] Y. C. Zhou, J. Liu, and D. L. Harry. A matched interface and boundary method for solving multi-flow Navier-Stokes equations with applications to geodynamics. *J. Comput. Phys.*, 231(1):223–242, 2012.

[84] Y. C. Zhou and G. W. Wei. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. *J. Comput. Phys*, 219(1):228–246, 2006.

[85] Y. C. Zhou, S. Zhao, M. Feig, and G. W. Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *J. Comput. Phys*, 213(1):1–30, 2006.

PUBLICATION

1. Wenqiang Feng, Xiaoming He, Yanping Lin, Xu Zhang, *Immersed Finite Element for Interface Problems with Algebraic Multigrid Solver*, Submitted.

2. Wenqiang Feng, Xiaoming He, Zhu Wang,Xu Zhang, *Non-iterative domain decomposition methods for a non-stationary Stokes-Darcy model with Beavers-Joseph*, Applied Mathematics and Computation,2:453-463,2012.

3. Wenqiang Feng, Shumin Li, Kelong Zheng, *A non-local bilateral filter for image denoising*, ICACIA(2010), pp. 253-257. (EI:11807001)

4. Kelong Zheng, Wenqiang Feng, Hanlin Chen, *An adaptive non-local means algorithm for image denoising via pixel region growing and merging*, CISP(2010), pp. 621-625.(EI:11676923)

VITA

Wenqiang Feng was born in a small village in Hubei province, China. He got the bachelor degree under the supervision of Prof. Kelong Zheng and Yiping Xu with thesis *The Application of Algebraic Multigrid Algorithms in Image Denoising* at Southwest University of Science and Technology in 2009. Then, he was recommended to Department of mathematics, University of Science and Technology of China. He got the master degree under the supervision of Prof. Shumin Li with thesis *The Application of the Non-Local Algorithm in Image Denoising* at Department of mathematics, University of science and technology of China in 2011. In May 2013, he received his Master Degree in Applied Mathematics from the Missouri University of science and Technology , Rolla, Missouri, USA.

As a student, besides taking the courses and participating in research projects, he has not given up any chance to improve himself. He also served as the president of the Chinese Scholar and Student Association at Missouri University of Science and Technology.