Immersed and Discontinuous Finite Element Methods

Nabil Chaabane

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Slimane Adjerid, Chair
Tao Lin, Co-chair
Serkan Gugercin
Yuriko Renardy

March 20, 2015
Blacksburg, Virginia

Keywords: Immersed finite element, discontinuous Galerkin, Stokes interface problem,
LDG, Q_1/Q_0

Immersed and Discontinuous Finite Element Methods

Nabil Chaabane

(ABSTRACT)

In this dissertation we prove the superconvergence of the minimal-dissipation local discontinuous Galerkin method for elliptic problems and construct optimal immersed finite element approximations and discontinuous immersed finite element methods for the Stokes interface problem.

In the first part we present an error analysis for the minimal dissipation local discontinuous Galerkin method applied to a model elliptic problem on Cartesian meshes when polynomials of degree at most k and an appropriate approximation of the boundary condition are used. This special approximation allows us to achieve $k+1$ order of convergence for both the potential and its gradient in the L^2 norm. Here we improve on existing estimates for the solution gradient by a factor $\sqrt{h}$.

In the second part we present discontinuous immersed finite element (IFE) methods for the Stokes interface problem on Cartesian meshes that does not require the mesh to be aligned with the interface. As such, we allow unfitted meshes that are cut by the interface. Thus, elements may contain more than one fluid. On these unfitted meshes we construct an immersed Q_1/Q_0 finite element approximation that depends on the location of the interface. We discuss the basic features of the proposed Q_1/Q_0 IFE basis functions such as the unisolvent property. We present several numerical examples to demonstrate that the proposed IFE approximations applied to solve interface Stokes problems maintain the optimal approximation capability of their standard counterpart applied to solve the homogeneous Stokes problem. Similarly, we also show that discontinuous Galerkin IFE solutions of the Stokes interface problem maintain the optimal convergence rates in both L^2 and broken H^1 norms. Furthermore, we extend our method to solve the axisymmetric Stokes interface problem with a moving interface and test the proposed method by solving several benchmark problems from the literature.

This work received support from NSF grants DMS-1016313 and DMS-0809262.

Dedication

To my parents.

Acknowledgments

I would like to take the opportunity to reflect on the past five years and thank all the friends and family who helped me and supported me along the way.

I would like to express my sincerest feelings of gratitude to both my advisors Dr. Slimane Adjerid and Dr. Tao Lin, who guided me through my Ph.D. Their constant advise have greatly helped me work through several challenging problems and their support have helped me overcome my failures. Their integrity and passion have inspired me to become a better scholar.

I would like to thank Dr. Pengtao Yue for his guidance through my last part of my dissertation. His insightful advises and comments helped me extend my work and approach more complex problems.

Members of my Ph.D. committee Dr. Yuriko Renardy and Dr. Serkan Gugercin deserve my sincerest thanks. I have learned a lot from each of them and I am grateful for their valuable advises and insightful comments on my research and dissertation.

Special thanks to Ms. Eileen Shugart for being my teaching mentor for the past years and sharing a lot of her valuable experience with me.

I wish to express my sincerest gratitude to my parents. Their constant support is an inspiration for me to become a better person and a better scholar.

Contents

1 Introduction **1**

1.1 Discontinuous Galerkin method 1

1.1.1 *A priori* error estimates for the DG method 3

1.1.2 Superconvergence error estimates 4

1.1.3 *A posteriori* error estimates for the DG method 5

1.2 The immersed finite element method 6

1.2.1 Illustration of the IFE method for a one-dimensional elliptic problem 6

1.2.2 History of the IFE method 8

1.3 The Stokes interface problem 9

1.3.1 Statement of the problem 9

1.3.2 Applications of the Stokes interface problem 10

1.3.3 Survey of numerical methods for solving the Stokes problem 12

1.3.4 Numerical methods for solving the Stokes interface problem 14

1.4 Motivations for studying immersed finite element methods 16

1.5 Outline of the dissertation 19

2 Local discontinuous Galerkin method **20**

2.1 Problem formulation and notations 20

2.2 The md-LDG method 22

2.3 Error analysis 24

2.4 Numerical results 35

3 Q_1/Q_0 Immersed finite element approximations for the Stokes interface problem 38

3.1 Space $S_h(\Omega)$ . . . 39

3.2 Space $\tilde{S}_h(\Omega)$ . . . 44

3.3 Q_1/Q_0 particular IFE functions with surface force . . . 46

3.4 Basic properties of the Q_1/Q_0 IFE space $S_h(\Omega)$ . . . 48

4 Immersed finite element method for the Stokes interface problem 56

4.1 Weak formulation and finite element scheme . . . 57

4.2 Numerical simulations . . . 60

4.2.1 Approximation properties of Q_1/Q_0 IFE spaces . . . 60

4.2.2 Convergence of the immersed finite element method . . . 66

5 Immersed finite element method for the Stokes problem with a moving interface 73

5.1 Introduction . . . 73

5.2 IFE implementation . . . 74

5.2.1 Even-odd rule . . . 75

5.2.2 Surface force . . . 76

5.2.3 Immersed finite element method . . . 77

5.3 Numerical simulations . . . 81

6 Axisymmetric Stokes interface problem 89

6.1 IFE spaces and particular functions for the Stokes interface problem . . . 92

6.1.1 Q_1/Q_0 IFE shape functions without surface force . . . 92

6.1.2 Q_1/Q_0 particular IFE functions with surface force . . . 97

6.1.3 Basic properties of the Q_1/Q_0 IFE space . . . 99

6.2 An immersed discontinuous Galerkin method . . . 101

6.3 Numerical simulations . . . 105

6.4 Axisymmetric Stokes problem with a moving interface . . . 107

6.4.1 Surface force 108
6.4.2 Proposed algorithm 109
6.4.3 Numerical experiments 110
6.4.4 Re-sampling of the interface 115

7 Conclusion and future work **117**

7.1 Contribution 117
7.2 Future work 118
7.2.1 High order IFE spaces 118
7.2.2 *A priori* error analysis 119
7.2.3 Three-dimensional interface problems 119

List of Figures

1.1 Piecewise linear IFE shape functions. 8

1.2 A domain Ω and an interface Γ. 11

1.3 Stable finite element functions for the Stokes problem where black dots are the nodes used to construct the Lagrange shape functions. 13

1.4 An immersed finite element mesh (right) and a body-fitted mesh (left). . . . 17

2.1 An example of a Cartesian mesh $\mathscr{T}$ and a vector $\mathbf{v}$ to define mesh orientation with $\partial\Omega^{\pm} = \Gamma_1^{\pm} \cup \Gamma_2^{\pm}$. 24

3.1 Reference interface elements of Type I (left) and Type II (right). 40

3.2 The $(u_1, u_2, p)^T$ components of the IFE shape function $\mathbf{\Phi}_1$ (left) and standard shape function $\mathbf{\Psi}_1$ (right). 43

3.3 An interface element. 52

4.1 The regions $\Delta^{\pm}$, $\Delta_h^{\pm}$ and their intersections. 60

5.1 A rectangular domain Ω with an immersed interface $\Gamma(t)$. 75

5.2 An illustration of the Even-Odd algorithm. 76

5.3 An illustration of the quadratic parametric interpolation used to determine the curvature of the interface and the normal vector to the interface at (x, y). 77

5.4 Flowchart of an algorithm used to solve the Stokes interface problem. 81

5.5 Finite element errors versus the degrees of freedom in the *log-log* scale for adaptive (-o) and uniform (- -□) refinements. 82

5.6 Initial state of the system for Example 5.3.1 (upper left), the mesh used to discretize Stokes problem (5.2a)−(5.2c) at $t = 0$ (upper right) and a blow-up of the mesh near the interface (bottom). 83

5.7 Simulation of Example 5.3.1. 84

5.8 Kinetic energy of the system versus time (left). Volume of the drop versus time (right). 85

5.9 The computational domain for drop deformation in shear flow. 86

5.10 The length L and width W of a deformed drop. 86

5.11 Simulation of Example 5.3.2. 88

6.1 The $3D$ domain V and a typical cylindrical element T^3 (upper left), and their projections onto the meridian plane (upper right). The 3D domain and interface (lower left) and their axisymmetric projection (lower right). 93

6.2 The (u_r, u_z, p) components of the IFE shape function $\mathbf{\Phi}_1$ (left) and standard shape function $\mathbf{\Psi}_1$ (right). 98

6.3 The distance R used to compute the curvature κ_2. 108

6.4 The computational domain for transient drop deformation under extensional flow in three-dimensional domain (left) and its projection onto the meridian plane (right). 111

6.5 Drop deformation in extensional flow versus time (top) and blown-ups, near the interface, of the meshes used to partition the domain Ω (bottom) for Example 6.4.1. 112

6.6 Drop deformation in extensional flow versus time (top) and blown-ups of the three-dimensional drops (bottom) for Example 6.4.1. 113

6.7 Drop deformation in extensional flow versus time for Example 6.4.1. 114

6.8 Control points at $t = 0$ (left) and $t = 1$ (right) for Example 6.4.1 without re-sampling. 114

6.9 Drop deformation in extensional flow versus time for Example 6.4.2 using re-sampling. 115

6.10 Drop volume versus time for Example 6.4.2. 116

List of Tables

2.1 L^2 errors on uniform meshes having $N = 16, 36, 64, 100, 144, 196, 256, 324, 400$ elements and $k = 1, 2, 3, 4$ with $\mathscr{P}^+ g_D$ in (2.16) and g_D in (2.15). 36

2.2 L^2 errors on uniform meshes having $N = 16, 36, 64, 100, 144, 196, 256, 324, 400$ elements and $k = 1, 2, 3, 4$ with g_D in (2.16) and $\mathscr{P}^+ g_D$ in (2.15). 37

4.1 L^2 interpolation errors for Example 4.2.1 using the space $S_h(\Omega)$. 63

4.2 Broken H^1 interpolation errors in the velocity for Example 4.2.1 using the space $S_h(\Omega)$. 64

4.3 L^2 errors of the L^2 projection for Example 4.2.1 using the space $\tilde{S}_h(\Omega)$. . . . 64

4.4 Broken H^1 errors of the L^2 projection of the velocity for Example 4.2.1 using the space $\tilde{S}_h(\Omega)$. 64

4.5 L^2 interpolation errors for Example 4.2.2 using the space $S_h(\Omega)$. 65

4.6 Broken H^1 interpolation errors for Example 4.2.2 using the space $S_h(\Omega)$. . . 65

4.7 L^2 interpolation errors for Example 4.2.3 using the space $S_h(\Omega)$. 66

4.8 Broken H^1 interpolation errors for Example 4.2.3 using the space $S_h(\Omega)$. . . 67

4.9 L^2 IFE errors for SIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$. 67

4.10 L^2 IFE error for NIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$. 68

4.11 Broken H^1 IFE errors for SIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$. 68

4.12 Broken H^1 IFE errors for NIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$. 68

4.13 L^2 IFE errors for SIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$. 68

4.14 L^2 IFE errors for NIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$. 69

4.15 Broken H^1 IFE errors for SIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$. 69

4.16 Broken H^1 IFE errors for NIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$. 69

4.17 L^2 IFE errors for SIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$. 70

4.18 L^2 IFE errors for NIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$. 70

4.19 Broken H^1 IFE errors for SIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$. 71

4.20 Broken H^1 IFE errors for NIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$. 71

4.21 L^2 IFE errors for SIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$. 71

4.22 Broken H^1 IFE errors for SIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$. 71

4.23 L^2 IFE errors for NIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$. 72

4.24 Broken H^1 IFE errors for NIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$. 72

6.1 L^2 interpolation errors for Example 6.3.1. 106

6.2 H^1 interpolation errors for Example 6.3.1. 106

6.3 L^2 IFE errors for NIPG method applied to Example 6.3.1. 106

6.4 H^1 IFE errors for NIPG method applied to Example 6.3.1. 106

6.5 L^2 IFE errors for SIPG method applied to Example 6.3.1. 107

6.6 H^1 IFE errors for SIPG method applied to Example 6.3.1. 107

Chapter 1

Introduction

Numerical methods for partial differential equations have seen important progress over the past several decades. The rapid development of computer hardware played an important role in allowing the implementation of new methods to carry out simulations of complex systems. The recent development allows for faster and more accurate solutions in areas that lack analytical understanding. However, the scientific community still faces many challenges when solving some of these equations numerically; for instance, some numerical methods exhibit instability and inaccuracy when applied to some problems. To circumvent such difficulties, a large amount of resources (time, memory use, complex algorithms) has to be used, and hence there is a need for more efficient methods. In particular, the relatively recent discontinuous Galerkin method (DGM), which exhibits several desirable properties, may be used to solve challenging problems such as problems where shocks form or systems that exhibit sharp transitions. Also, the immersed finite element method, which belongs to a larger class of extended finite element methods, has shown a great potential to solve challenging problems where the PDEs' coefficients are discontinuous. These problems arise if the domain contains multiple materials separated by interfaces.

In this dissertation, we (*i*) study the local discontinuous Galerkin method applied to a second order elliptic problem and carry out an error analysis of the method, (*ii*) construct a new Q_1/Q_0 immersed finite element space and particular IFE functions to capture the non-smoothness of the solution of the Stokes interface problem and (*iii*) derive IFE schemes to solve the Stokes interface problem.

1.1 Discontinuous Galerkin method

In the late 1950's, the finite element (FE) method was developed to solve PDEs arising in structural mechanics (see [61] for a brief history of the FE method). The idea was to ap-

proximate smooth functions using a linear combination of piecewise polynomials. The finite element approximation is determined by solving the linear system of the discretized PDEs. Several finite element formulations can exist for a given differential equation, but as Johnson [74] states, in general one can follow the general framework below:

• transform the differential equation into a variational problem,

• discretize the infinite variational problem in a finite dimensional space,

• implement the method on a computer and solve the discrete problem.

A finite element method usually involves two finite element spaces, the trial space and the test space. If the trial and test spaces are identical, the FE method is called a Galerkin method. The original finite element method was designed such that the numerical solution is forced to be continuous across element boundaries. This makes the method very inefficient or impossible to use in certain situations such as shock formation or discontinuous coefficients. Consult [136] and the references cited therein for more information on FE methods. If the FE solution is not required to be continuous across the element boundaries, we call it a discontinuous Galerkin (DG) method. DG methods are suitable for solving systems that exhibit sharp transitions or discontinuities and is widely used in engineering applications.

Furthermore, the DG method can be seen as a finite volume method using piecewise polynomial basis functions. Therefore, the DG method may be regarded as an extension of the finite volume method to arbitrarily high orders of accuracy, and therefore inherits desirable properties from both the finite element and finite volume methods. The solution space is the span of piecewise polynomials constructed relatively to the mesh and this allows solution discontinuities (e.g., shocks) to be captured sharply relative to element boundaries. The DG method gained popularity because

• it simplifies adaptivity since inter-element continuity is neither required for h-refinement nor p-refinement.

• it can be locally conservative (the method can be designed to conserve the appropriate physical quantities, e.g., mass, momentum, and energy on an elemental basis).

• it is well suited to solve problems on locally refined meshes with hanging nodes.

• it is explicit for some problems, allowing the approximate solutions to be computed on element by element, or in parallel within layers of elements, in accordance with domain of dependence requirements.

• it exhibits strong superconvergence of solutions for hyperbolic [1, 125], elliptic [3] and

convection-diffusion problems [6] that can be used to estimate the discretization error.

The DG method was first introduced in 1973 by Reed and Hill [117] as a technique to solve neutron transport problems

$$\nabla \cdot (\mathbf{a}u) + ru = f, \tag{1.1}$$

where r is a real number and $\mathbf{a}$ a constant vector. The hyperbolic nature of the equation coupled with the "non-continuity across element boundaries" property of the DG method allows to compute the solution locally element by element when elements are suitably ordered with respect to the characteristic direction. The first numerical analysis was carried out by Lesaint [86] in 1975 for a linear advection equation, and has been extensively studied thereafter. The DG method was studied for initial-vlaue problems for ordinary differential equations [5, 25, 87, 122], hyperbolic [21], parabolic [52, 53, 123] and elliptic [120] PDEs. It was also studied for systems of PDEs such as the Stokes problem [43], the Navier-Stokes problem [108] and the elasticity problem [62]. It offers the possibility to use more general mesh configurations and discontinuous basis functions, which simplify both h- and p-refinements. Here, the h-refinement refers to the mesh refinement or coarsening and the p-refinement refers to the method order variation. Furthermore, DG methods have simple communication pattern between neighboring elements which makes it suitable for parallel computation. Also, it can handle problems with complex geometries to high order.

Regardless of the method used, an assessment of the accuracy of the approximate solution has to be obtained, since the exact solution is not given in practice. Also, an efficient refinement strategy cannot be implemented unless an *a posteriori* error estimate is integrated in the method to guide adaptivity and stop the refinement process. These estimates are constructed using superconvergence results. In this section, we present a short survey of *a priori* error estimates, *a posteriori* error estimates and superconvergence results of the DG method.

1.1.1 *A priori* error estimates for the DG method

A priori error estimates give an upper bound in a specific norm of the error that relates the exact solution to the FE solution. Their derivation is usually restricted by the smoothness assumptions made about the exact solutions, and their use is limited since the exact solution is unknown. However, they provide a starting point to construct efficient asymptotically correct *a posteriori* error estimates, which estimates the accuracy of the computed FE solutions.

The first analysis of the DG method, used to solve the neutron transport equations ,was carried out by LeSaint and Raviart in [86], where they proved $O(h^k)$ convergence rate for general triangulations in the L^2 norm and $O(h^{k+1})$ order of convergence for Cartesian meshes where tensor product of polynomials of degree k are used. Johnson at al. [75] improved the

estimate for general triangulations by a factor of $\sqrt{h}$ and their result was verified numerically by Peterson [115]. In practice, the optimal rate of convergence is usually observed. Richter [119] was able to prove optimal order of convergence under some uniformity assumptions for structured non-Cartesian grids. The optimal order of convergence for a model elliptic problem, elasticity and Navier-Stokes problems are obtained by adding penalty terms, see [120] and the references therein for a review of these methods.

The above estimates are constructed under the assumption that the exact solution is sufficiently smooth. Since this is not the case for many applications, error estimates for applications with non-smooth solutions were a subject of investigation by several scientists. The convergence of the method for solutions that admit discontinuities was investigated and the convergence of the method was established in [99]. In [98, 134], the mesh-discontinuity configuration was explored for piecewise linear and piecewise constant approximations and the optimal order of convergence was established under suitable assumptions on the mesh. In [54] Falk and Richter showed $O(h^{k+1/2})$ rate of convergence for Friedrichs systems using general triangulations. Later, Cockburn *et al.* [46] showed how to obtain $O(h^{2k+1})$ rate of convergence on Cartesian meshes by postprocessing the approximate solution.

The DG method was extended to nonlinear systems of PDEs by Cockburn *et al.* in [47], where they proposed to use a variation of the original DG method to solve nonlinear systems of conservation laws and the method was referred to as the Runge-Kutta discontinuous Galerkin (RKDG) method. In fact, it is a DG method in space and a special total variation diminishing (TVD) Runge-Kutta time discretization, coupled with a generalized slope limiter strategy to handle the discontinuities. Cockburn and Shu [48] explored the local discontinuous Galerkin (LDG) method for time dependent convection-diffusion problems. They showed that, by transforming the equation to a first order degenerate system, the LDG method can be seen as an extension of the RKDG method used for purely hyperbolic systems.

1.1.2 Superconvergence error estimates

Superconvergence occurs when the approximate solution converges at certain points, lines, surfaces, or places at higher rates than elsewhere. Therefore, the computational solution is more accurate at these places and converges at a rate higher than $O(h^{k+1})$, which is usually the discretization error in the L^2 norm. Mathematically, superconvergence means that the local discretization errors converge at an $O(h^q)$ rate, where $q > k+1$, h is the mesh size and k is the polynomial degree used to approximate the solution. The superconvergence results can be used to design a post-processing procedure to improve the accuracy of the computational approximations, construct efficient *a posteriori* error estimates or detect discontinuities (*e.g.* align the mesh or use stabilization techniques). These estimates have been investigated for FE methods [129] and for DG methods applied to ordinary differential

equations [87], hyperbolic problems [5, 8, 9] and convection-diffusion problems [6, 37]. In [34] Castillo investigated the superconvergence behavior of the LDG method applied to a two-point elliptic boundary-value problem and showed, using the numerical flux proposed in [35], that on each element the k-degree LDG solution gradient is $O(h^{k+1})$ superconvergent at the shifted roots of the k-degree Legendre polynomial.

In [6], Adjerid and Klauser investigated the superconvergence properties of the LDG method for transient convection-diffusion problems in one-dimension. They proved that the discontinuous finite element solutions of convection-dominated problems are $O(h^{k+2})$ superconvergent at Radau points. For diffusion-dominated problems, the solution derivative is $O(h^{k+2})$ superconvergent at the roots of the derivative of Radau polynomial of degree $k+1$. They used these results to construct several asymptotically exact *a posteriori* finite element error estimates. In [37], a family of FE methods for one-dimensional convection-diffusion problem was investigated, where DG methods with different numerical traces, discontinuous Petrov-Galerkin methods and hybridized mixed methods were included. The authors showed that the numerical traces of the solution and its gradient are superconvergent at the mesh nodes, provided that the traces are conservative. In particular, for the LDG method, they showed that the superconvergence is $O(h^{2k+1})$ when polynomials of degree at most k are used. Extensive computations in [3] suggest that the LDG solution of elliptic problems obtained by interpolating Dirichlet boundary conditions at Radau points is $O(h^{k+1})$ superconvergent for both the solution gradient and the potential.

1.1.3 *A posteriori* error estimates for the DG method

A posteriori error estimates constitute a pillar of the adaptive refinement as they indicate the regions where more or less refinement is needed. Several *a posteriori* error estimates for the DG methods have been developed. However, these estimates can be classified as in [56] into four classes: (*i*) residual error estimates are found by solving local finite element problems created on either an element or a subdomain, (*ii*) flux-projection error estimates are found by calculating a new flux via a post processing of the finite element solution and then taking the difference of the new smoother flux and the original flux, (*iii*) extrapolation error estimates are found by taking the difference of finite element solutions having different orders or different meshes, and (*iv*) interpolation error estimates.

To quantify the efficiency of the *a posteriori* error estimates, the effectivity index is used, which is the ratio of the computed error to the exact error. An effectivity index close to one means that the *a posteriori* estimate used is efficient. This effectivity index should converge to one under h- and p-refinement. The *a posteriori* error estimates can be further sub-classified under local and global estimates. A global *a posteriori* error estimate gives a measure of the global error over the whole domain. On the other hand, a local *a posteriori* estimate gives a measure of the error on every element, which makes it more suitable for

local mesh refinement/coarsening (h-refinement) and local order variation (p-refinement). The latter approach offers several advantages as it is usually easy to compute, asymptotically exact and provides one order higher estimates. The global *a posteriori* error estimates were studied for elliptic [19, 121], parabolic [53, 58] and hyperbolic problems [83]. The local *a posteriori* error estimates were constructed for one-dimensional linear and nonlinear hyperbolic problems [5], two-dimensional hyperbolic problems [8, 9, 16, 22, 81] and elliptic problems [78].

1.2 The immersed finite element method

The immersed finite element (IFE) method was introduced in 1998 by Li [93] to solve one-dimensional second order boundary value problems with discontinuous coefficients. These problems model several phenomena in different areas of engineering and science, and they typically arise if the domain contains multiple materials. In this section, we illustrate the IFE method applied to one-dimensional boundary problem. Then, we present a survey of the history of the IFE method.

1.2.1 Illustration of the IFE method for a one-dimensional elliptic problem

We consider the following elliptic problem defined on $I = [0, 1]$:

$$-(\beta(x)u'(x))' + q(x)u(x) = f(x), \quad \text{for } x \in [0, \alpha) \cup (\alpha, 1], \tag{1.2a}$$

$$u(0) = u(1) = 0. \tag{1.2b}$$

We assume that the coefficient $\beta(x)$ is a piecewise constant defined as follows:

$$\beta(x) = \begin{cases} \beta^-, & \text{if } x < \alpha, \\ \beta^+, & \text{if } x > \alpha, \end{cases}$$

where $\alpha \in (0, 1)$ is a constant that we shall call interface. We further define $I^- = [0, \alpha)$ and $I^+ = (\alpha, 1]$.

The solution of the problem (1.2a-1.2b) is not smooth due to the discontinuity of the coefficient $\beta(x)$. However, the behavior of the solution at the interface α is known [93]. In fact the following jump conditions occur at α:

$$[u(\alpha)] = 0, \tag{1.2c}$$

$$[\beta(\alpha)u'(\alpha)] = 0, \tag{1.2d}$$

where the jump operator $[\cdot]$ is defined as:

$$[u] = \lim_{x \to \alpha-} u(x) - \lim_{x \to \alpha+} u(x).$$

This problem can be solved and optimal order of convergence is obtained if the mesh used to discretize the domain is fitted to the interface (*i.e.* one of the mesh nodes coincides with α). However, if the mesh is constructed regardless of the interface location, the standard finite element space is not guaranteed to capture the essence of the solution due to its non-smoothness. Therefore, there is a need to construct a more suitable space on such meshes. This idea represents the main ingredient of the IFE method. In this section, we show how to construct the IFE space.

As mentioned above, the main idea is to construct a more suitable space that accounts for the non-smoothness of the solution. Let us first partition the domain $[0, 1]$ into N subintervals using a mesh $\mathcal{T}_h$. The subintervals shall be called elements and we denote them as $I_j = [x_{j-1}, x_j]$, $j = 1, 2, ...N$. The element that contains α will be denoted $I_i = [x_{i-1}, x_i]$ and shall be called interface element while the remainder of the elements shall be called non-interface elements.

Naturally, the basis functions constructed on the interface element should be non-smooth at the interface, and in particular, they should obey the jump conditions (1.2c-1.2d). The IFE shape functions will be defined piecewisely on $I_i^- = [x_{i-1}, \alpha]$ and $I_i^+ = [\alpha, x_i]$. On the other hand, the standard FE basis functions will be used on non-interface elements. The IFE space is defined on the interface element I_i as follows:

$$S_h(I_i) = \{\phi : \phi|_{I_i^\pm} \in P_k(I_i^\pm) \text{ and } [\phi(\alpha)] = 0, [\beta(\alpha)\phi'(\alpha)] = 0\},$$

where P_k is the space of polynomials of degree at most k. For instance, the local piecewise linear IFE shape functions ϕ_m, $m = 1, 2$ are constructed on the interface element $I_i = [x_{i-1}, x_i]$ using the following procedure:

- Define the general piecewise basis function as $\phi_m^\pm(x) = a_0^\pm + a_1^\pm x$.
- Enforce the Lagrange conditions $\phi_m^-(x_{i-1}) = \delta_{1,m}$, $\phi_m^+(x_i) = \delta_{2,m}$.
- Enforce the jump conditions (1.2c-1.2d) at α: $[\phi_m(\alpha)] = 0$ and $[\beta\phi_m'(\alpha)] = 0$.

Figure 1.1 shows the piecewise linear IFE shape functions on the reference interface element $[0, 1]$ with $\alpha = 0.3$, $\beta^- = 1$ and $\beta^+ = 4$. The global IFE basis functions are continuous and are constructed using the IFE shape functions $\phi_m(x)$, $m = 1, 2$. Therefore, they are in $H^1(I)$, and a Galerkin scheme using these IFE basis functions is a conforming finite element method. The IFE method has been extended to higher order approximations, and a family of piecewise quadratic IFE basis functions were developed in [32, 101], and their approximation capability was investigated. In [7], IFE shape functions for any polynomial degree were constructed and optimal order of convergence of the IFE method was established.

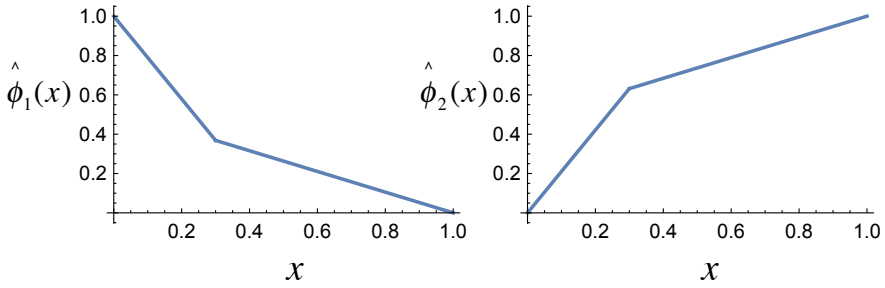


Figure 1.1: Piecewise linear IFE shape functions.

1.2.2 History of the IFE method

Since the time the IFE method was introduced by Li in [93], it has been extended to higher order approximations by Adjerid and Lin [7], and applied to second order elliptic problems [41, 60, 64, 65, 66, 67, 82, 96, 100, 130]. The first piecewise bilinear IFE space on rectangular mesh was introduced in [100] and the approximation capability of the IFE space was discussed in [64] where the optimality of the IFE methods was shown.

Note that these IFE basis functions are not continuous across element boundaries and thus, nonconforming. In [133], rotated IFE basis functions were constructed and the optimality of the IFE method was proved. For elliptic problems with non-homogeneous jump conditions, IFE functions were created to enrich the IFE space and capture the discontinuity [64]. In [96], Li *et al.* have developed both conforming and nonconforming linear IFE basis functions on triangular meshes. The nonconforming linear finite element method has two majors advantages: (i) the construction of the basis function is simple because the continuity across elements is not enforced and (ii) the applicability of the fast solvers to the IFE linear system is possible since it has the same structure as the standard FE linear system. The major drawback is the difficulty to carry out error analysis, while the analysis of the conforming IFE method is straightforward. However, the conforming IFE method yields a stiffness matrix that is denser than its counterpart obtained using the standard FE method, since extra degrees of freedom are added on the interface elements. The approximation capability of the nonconforming spaces is studied in [95], where the authors proved that the interpolation error converges optimally in both L^2 and H^1 norms. Due to the nature of the IFE basis functions, they used a multi-point Taylor expansion instead of the standard scaling argument. The extension of the IFE method to higher order approximation on triangular meshes was done in [4, 20], where the authors showed numerically that interior penalty terms have to be added to reach optimal convergence rates.

The most recent work is an extension of the IFE method to systems of PDEs and time dependent problems. In [105], Lin and Zhang constructed bilinear and linear IFE basis functions for the planar elasticity problem on triangular and Cartesian meshes and observed optimal convergence rates in both L^2 and H^1 norms. However, they showed that for some configurations of the interface location, the basis functions cannot be constructed. To circumvent this, they constructed nonconforming IFE basis functions and proved the unisolvent property [104]. In addition, they observed that, by using these basis functions, the proposed IFE method is a locking-free scheme.

In [12] an IFE method was used to obtain a spatial discretization of a transient semi linear parabolic interface problem with static interfaces and the backward Euler method was used to discretize the problem in time. The authors also attempted an error analysis of the proposed method. In [103] Lin and Sheen proposed an alternative approach to solve the problem by using the Laplace transform. This method yields a set of decoupled interface problems that can be solved independently which makes this approach well suited for parallel computing. He *et al.* [68] proposed a family of Crank-Nicolson (CN) IFE schemes for solving parabolic equations with a moving interface and showed numerically that these schemes achieve optimal order of convergence. For higher order PDEs, IFE methods were developed to solve the one-dimensional beam interface problem and the two-dimensional bi-harmonic interface problem [102]. In order to avoid using higher degree approximations, the fourth order differential equations were reduced to a second order system and the mixed finite element method was used.

1.3 The Stokes interface problem

In this section, we present the two-dimensional Stokes interface problem and its applications. Then, a short survey of numerical methods used to solve the Stokes problem and the Stokes interface problem will be provided. Finally, we discuss our motivations to develop an immersed finite element (IFE) method to solve Stokes interface problem.

1.3.1 Statement of the problem

The Stokes problem can be seen as a linearization of the Navier-Stokes problem and is used to simulate creeping incompressible fluids. In particular it models the velocity $\mathbf{u}(X) = (u_1(X), ..., u_d(X))$ and the pressure $p(X)$ of incompressible fluids in an open domain $\Omega \in \mathbb{R}^d$, $d = 2, 3$ with boundary $\partial\Omega$ such that $X = (x_1, ..., x_d) \in \Omega$. We assume that we have two immiscible fluids characterized by two different viscosities $\nu^{\pm}$. Therefore we assume that the domain is separated by an interface Γ into two sub-domains Ω^+, Ω^- such that each domain contains one fluid, which yields the Stokes interface problem. The closure of the physical domain can be written as $\overline{\Omega} = \overline{\Omega^+ \cup \Omega^- \cup \Gamma}$, see Figure 1.2 for an illustration. The Stokes

interface problem is described by the following equations:

$$-\nabla \cdot \mathbf{S}(\mathbf{u}, p) = \mathbf{f}, \quad \text{in } \Omega^- \cup \Omega^+, \tag{1.3a}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega^- \cup \Omega^+, \tag{1.3b}$$

$$\mathbf{u} = \mathbf{g}, \quad \text{on } \partial\Omega, \tag{1.3c}$$

where

$$\mathbf{S}(\mathbf{u}, p) = \nu \boldsymbol{\epsilon}(\mathbf{u}) - p\mathbf{I}, \tag{1.3d}$$

with

$$\boldsymbol{\epsilon}(\mathbf{u}) = \nabla \mathbf{u} + (\nabla \mathbf{u})^T, \tag{1.3e}$$

$\mathbf{f}$ is the body force assumed to be in $(L^2(\Omega))^d$, $\mathbf{g}$ is the Dirichlet boundary condition assumed to be in $(L^2(\partial\Omega))^d$ and

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \cdots & \frac{\partial u_1}{\partial x_d} \\ & \vdots & \\ \frac{\partial u_d}{\partial x_1} & \cdots & \frac{\partial u_d}{\partial x_d} \end{pmatrix}.$$

The viscosity coefficient ν is discontinuous across the interface Γ:

$$\nu(X) = \begin{cases} \nu^- & \text{if } X \in \Omega^-, \\ \nu^+ & \text{if } X \in \Omega^+, \end{cases}$$

where $\nu^+, \nu^- > 0$.

We complete the definition of the problem by imposing the following physical jump conditions across the interface

$$[\mathbf{S}(\mathbf{u}, p)\mathbf{n}]|_\Gamma = \boldsymbol{\sigma}, \tag{1.3f}$$

$$[\mathbf{u}]\,|_\Gamma = \mathbf{0}, \tag{1.3g}$$

where $\mathbf{n}$ is the unit vector normal to the interface and $\boldsymbol{\sigma}$ is the surface force along the interface. Note that we treat the two-dimensional Stokes interface problem (*i.e.,* $d = 2$) in Chapters 3-5 and the three-dimensional Stokes interface problem (*i.e.,* $d = 3$) in Chapter 6.

1.3.2 Applications of the Stokes interface problem

Incompressible Stokes and Navier-Stokes equations are important to understand the dynamics of multiphase flows with moving interfaces or free boundary problems in computational fluid dynamics (CFD). In general, interface problems arise when the domain contain two fluids such as water and oil, the same fluid at different states such as water and ice, or same fluid separated by a membrane. These models are usually characterized mathematically by the discontinuity of the viscosity across the interface and/or the presence of a surface force term along the interface. Next, we present three applications of the Stokes interface problem.

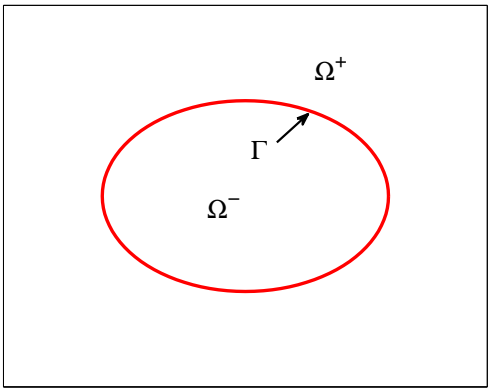


Figure 1.2: A domain Ω and an interface Γ.

Blood flow in the heart

Numerical simulation of the flow pattern in the heart is a challenging task due to the moving nature and complex structure of the heart's walls. In [111], Peskin circumvented this issue by embedding the domain of interest in a larger box filled with fluid and modeled the boundary conditions at the heart valves as forces exerted by the fibers on the fluid. Furthermore, he imposed periodic boundary conditions and showed that the exterior fluid has minor effect on the flow pattern inside the heart. The flow pattern is modeled using the Stokes interface problem with the heart's walls being the interface.

Cochlea (Inner ear) model

Researchers over the years have been trying to understand the complexity of the ear and how it functions. Theories about the auditory system (in particular the inner ear or the Cochlea) operation are still being developed. Computational models based on a coupled fluid/immersed-boundary system have been proposed. The immersed boundaries are used to describe the flexible and rigid parts of the Cochlea, such as the basilar membrane and the surrounding bone. These immersed boundaries exerts forces on the surrounding fluid in a specified manner depending on their location, which forces the Cochlea to move [73]. Thus the interaction between the fluid and the Cochlea describes the response of the Cochlea to vibrational stimulus and is modeled using Stokes interface problem.

Energy production

One of the main applications of the Stokes interface problem is in the field of energy production, where the recovery of oil can be improved by injecting steam, CO_2 or water into the

oil reservoir [50]. This yields a porous media flow problem involving several liquids or gas phases modeled using Stokes interface problem. From an industrial perspective, computational prediction tools are an asset to make choices of potential carbon capture and storage sites, so that the CO_2 will not be released in the atmosphere.

1.3.3 Survey of numerical methods for solving the Stokes problem

Given the large amount of literature related to solving Stokes problem, we present in this section some of the recent progress achieved in this field. It is well known that using the same spaces to approximate the velocity and the pressure usually yields an unstable scheme. Therefore, choosing the appropriate spaces for velocity and pressure is a major step toward constructing a stable scheme. Most of these spaces satisfy the so-called inf-sup condition, which was introduced by Brezzi and Babuska [14, 29] and has led to the development of many finite element formulations. For convenience, we introduce the following spaces Q_k and P_k. Q_k is the space of polynomials of degree at most k in each variable x and y and P_k is the space of polynomials of degree at most k. A finite element function is defined by the pair M/N, such that M is the space used to approximate the velocity and N is the space used to approximate the pressure.

A widely used finite element function is the so-called Taylor-Hood finite element function [11], which is the finite element function Q_2/Q_1 for rectangular elements and P_2/P_1 for triangular elements and was proved to satisfy the inf-sup condition and therefore is stable. This finite element function was extended to higher order approximations, and it has been shown that the finite element functions Q_k/Q_{k-1}, $k > 2$ are stable. However, stabilization techniques have to be used when using the finite element function Q_1/Q_0 [24]. The latter has been proved to be stable when the discontinuous Galerkin method is used [120]. Another stable finite element function is the so-called modified Taylor-Hood finite element function (or Q_2-iso-Q_1/Q_1 finite element function), where Q_1 is used to approximate both velocity and pressure, but the mesh used to approximate the velocity is finer than the mesh used to approximate the pressure. In fact, every element in the mesh used to approximate the pressure has to be subdivided into four congruent elements (if rectangular elements are used) and the new mesh is used to discretize the velocity.

An alternative way to obtain a stabilized finite element is to use the MINI finite element function Q_1^*/Q_1 [10], such that the space Q_1^* consists of the space Q_1 enriched with a bubble function. Figure 1.3 shows an illustration of some of these finite element functions on Cartesian meshes. On the other hand, Hughes *et al.* [72] proposed a Petrov-Galerkin finite element formulation that was shown to be convergent for general C^0 combinations of velocity and pressure, and hence circumvent restrictions of the Babuska-Brezzi condition. Douglas and Wang [51] proposed an absolutely stabilized finite element formulation for the Stokes equation in which solvability and convergence of the method do not depend on the stability

constant. Kechar and Silvester [79] introduced a local stabilization method which includes the jump of the pressure on the boundary of a macroelement.

Recently, the discontinuous Galerkin method has been used to solve Stokes and Navier-Stokes equations. Cockburn *et al.* [45] investigated the LDG method for the Stokes and Navier-Stokes systems in mixed forms and showed that the LDG method can handle meshes with hanging nodes, elements of general shapes, local spaces of different types, and weakly enforce the conservation of mass. Hansbo and Larson [62] introduced a stabilized DG method for the equations of the linear elasticity in the incompressible and nearly compressible case without using pressure variable and proved an optimal rate of convergence in a mesh-dependent norm. Lazrov and Ye [85] enforced the continuity of the velocity along the element boundaries using Lagrange multipliers to obtain optimal orders of convergence for both velocity and pressure in the L^2 norm.

Cockburn *et al.* [43] proposed a hybridizable DG method for Stokes flow in order to reduce the size of the resulting matrix to the numerical trace of the velocity and the mean of the pressure on the element boundaries. They claim that by using an augmented Lagrangian method, the globally coupled unknowns are further reduced to the numerical trace of the velocity only. Furthermore, the approximations of the velocity, pressure and velocity gradient converge with the optimal order of $k+1$ in the L^2 norm. An element by element postprocessing leads to another approximation of the velocity which converges with order $k+2$ in the L^2 norm. An analysis of a family of DG methods is provided in [120] for both Stokes and Navier-Stokes equations, where optimal energy estimates for the velocity and optimal L^2 estimates for the pressure are obtained.

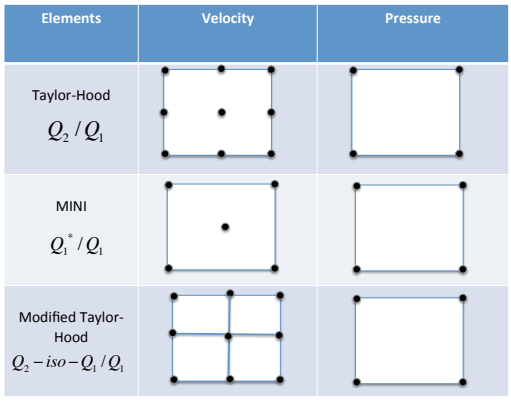


Figure 1.3: Stable finite element functions for the Stokes problem where black dots are the nodes used to construct the Lagrange shape functions.

1.3.4 Numerical methods for solving the Stokes interface problem

Numerous techniques have been proposed to solve the Stokes interface problem. Below, we present a short survey of the existing techniques. Each method has its own advantages and drawbacks. We classify the methods into five main categories: the finite element method, the ghost fluid method, the finite difference method, the immersed boundary method and the least-squares method.

The finite element method

One approach, proposed by Brackbill in [26], consists of regularizing the discontinuities by smoothing out the interface and removing the discontinuity leading to a method that requires finer meshes near interfaces due to its reduced accuracy near the interfaces. In order to avoid this difficulty, Buscaglia *et al.* [13] designed a finite element basis for the pressure that models jumps of the normal component of the surface force assuming that the tangential component of the surface force is continuous. They showed through numerical examples that the interpolation error in the pressure is $O(h^{3/2})$ accurate with first-degree polynomials for the pressure. Reusken [118] developed a modified XFEM space by deleting finite element basis functions having small support which improves the conditioning of the resulting system. Reusken proved that the modified XFEM space is as accurate as the standard XFEM space but with better stability properties. However, these XFEM spaces have to be carefully constructed to preserve accuracy. Belytschko and Fries [57] proposed a generalized/extended finite element method that may be applied to problems whose solutions are not smooth. An alternative method was proposed by Hansbo *et al.* [18] by considering two discrete solutions one on each of the enlarged sub-domains Ω^+ and Ω^- and enforcing the jump conditions weakly using a variant of Nitsche's method. Like the XFEM method, this method may suffer from ill-conditioning depending on the location of the interface. Zahedi *et al.* [63] were able to circumvent the difficulties faced in the latter method by slightly changing the variational formulation. They proved optimal convergence as well as the well-conditioning of the stiffness matrix. The relatively new discontinuous Galerkin method has also been applied to the Stokes interface problem [59] using piecewise polynomials of degree k for the velocity and of degree $k-1$ for the pressure on body-fitted meshes. They established optimal orders of convergence in the L^2 and energy norms, respectively, for pressure and velocity. Ohmori *et al.* [109] studied the Stokes interface problem and provided an error analysis for MINI and modified Taylor-Hood finite element functions.

The immersed interface method

The immersed interface method (IIM) is based on a finite difference formulation that maintains a second order accuracy by incorporating the jump conditions across the interface. It can be seen from (1.3f) that the pressure and the derivatives of the velocity are usually

not continuous across the interface. The IIM takes this fact into account when developing the difference equations. It was originally introduced by Leveque and Li [88] to solve the elliptic interface problem, then it was extended to solve the Stokes interface problem [89]. The main idea is to decouple the velocity and pressure, then use a Cartesian grid immersed finite difference to solve for each variable separately. The immersed finite difference method was further developed in [84, 92, 94, 97]. The main drawbacks of this method are (*i*) the non-symmetry of the linear system used to solve in order to obtain the discrete solution of the Stokes interface problem, (*ii*) the limitation of the method accuracy and (*iii*) the necessity to derive more jump conditions across the interface.

The ghost fluid method

The Ghost Fluid (GF) Method was first introduced by Fedkiw *et al.* [55] to capture the discontinuity across the multimaterial interface in the inviscid compressible Euler equation. It was then extended to solve multiphase incompressible flows [76, 77]. At every point of the domain of interest, a ghost node is created and each grid point contains the mass, momentum and energy for the fluid that exists at that point and a ghost mass, momentum and energy for the other fluid. Once these cells are created, the standard methods can be used to solve the problem regardless of the interface location. The main advantages of this method is that it is simple and relatively easier than the IIM to implement and the system of finite difference equations used from the GF method is symmetric for self adjoint elliptic problems. Unlike the IIM, the fast Poisson solver cannot be used; instead a slower algorithm has to be used to solve the linear system since the finite difference coefficients change at some grid points near the boundary. Also, this method may be only first-order accurate for mixed or non-homogenous Neuman boundary conditions. Moreover, the stability of this method may deteriorate if a higher-order scheme is used.

The immersed boundary method

The immersed boundary (IB) method is a finite difference-based scheme that was originally introduced by Peskin to model the blood flow in a human heart [111]. This method has been applied to many other problems, see [112, 113, 114] for a review of the method and its application. The main idea of this method is to distribute a singular force to nearby grid point using a delta function. Several delta functions have been developed, but the most commonly used functions are the hat function

$$\delta_\epsilon(x) = \begin{cases} (\epsilon - |x|)/\epsilon^2 & \text{if } |x| < \epsilon, \\ 0 & \text{if } |x| \geq \epsilon, \end{cases} \tag{1.4}$$

and Peskin's original discrete cosine delta function

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{4\epsilon}(1+\cos(\frac{\pi x}{2\epsilon})) & \text{if } |x| < 2\epsilon, \\ 0 & \text{if } |x| \geq 2\epsilon. \end{cases} \tag{1.5}$$

Both delta functions are continuous, but the cosine delta function, first introduced by Peskin, is smooth and most used in the literature. This method is robust and simple to implement and its extension to higher dimension is straightforward as we may define $\delta_\epsilon(x,y) = \delta_\epsilon(x)\delta_\epsilon(y)$. However, if the cosine delta function is used, the solution is only first order accurate and is smeared in the neighborhood of the interface. Also, in the formulation used to develop the IB method, the integral term that contains the surface force does not take into account the curvature of the interface which makes it hard to develop second order schemes.

The least-squares method

The least-squares method is attractive for several reasons. The first is that the spaces used to approximate the velocity and pressure don't have to obey the BB condition [31]. Therefore, one can use the same degree polynomial approximation for both pressure and velocity. Furthermore, the resulting algebraic system is symmetric positive definite and easy to precondition and thus, allows the use of efficient iterative schemes. In [33] the authors proposed a body-fitted least-squares method to solve the elliptic interface problem where they transformed the problem into a first order system and created a mesh that is fitted to the interface. They carried out an error analysis and proved that the optimal order of convergence is reached. This idea was extended to the Stokes interface problem in [69] where the author combined the least-squares method with a spectral collocation method to improve accuracy.

1.4 Motivations for studying immersed finite element methods

The second part of this work focuses on developing immersed finite element (IFE) methods for solving the Stokes interface problem on a structured Cartesian mesh. The IFE method can be seen as an alternative to the immersed boundary and immersed interface methods. The main idea is to create the mesh regardless of the location of the interface as shown in Figure 1.4, then design new finite element spaces and IFE particular functions that mimic the behavior of the solution across the interface. These IFE spaces are used only when an element is cut by the interface (such elements will be called interface elements). These new spaces are extremely easy to incorporate into existing codes, and they conserve the global

degrees of freedom. The IFE basis functions have to be locally constructed on every interface element, which is not expensive since the number of interface elements is usually small compared to the total number of elements in a mesh. Unlike the conventional FE method, the IFE method does not require the mesh to be tailored to fit the interface and therefore avoids the expensive re-meshing procedure.

In general, optimal rates of convergence for conventional finite element solutions are attained for fitted meshes where every element contains one fluid. However, this restriction leads to several drawbacks such as:

• excessive re-meshing for problems with moving or random interfaces (many problems may be solved with different interfaces corresponding to different parameters' values)

• excessive mesh refinement to resolve small structures in the domain such as thin layers

• prohibition of the use of uniform meshes to solve problems having nontrivial interfaces

In this dissertation, we try to provide a method that have the following properties:

• stability regardless of the location of the interface

• second order accuracy for the velocity and first order accuracy for the pressure

• flexible refinement of the mesh

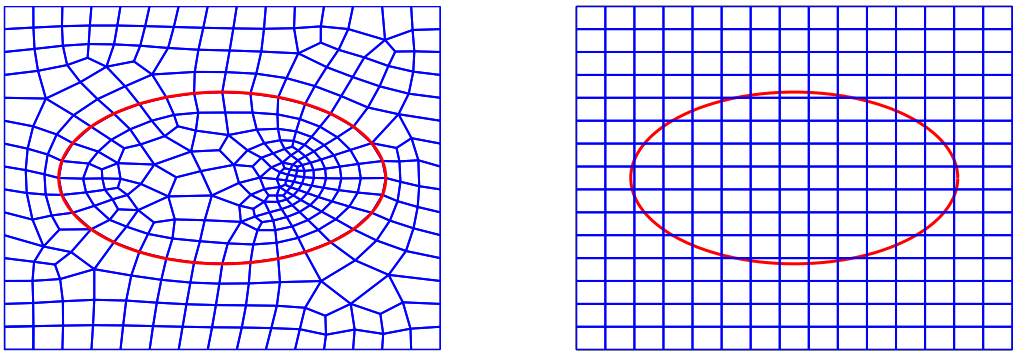


Figure 1.4: An immersed finite element mesh (right) and a body-fitted mesh (left).

The IFE method have been extensively studied for different interface problems (see section

1.2.2) and has exhibited stability and accuracy. In this work, we extend the IFE method to solve the Stokes interface problem and provide a stable and accurate method. This provides an efficient tool since it avoids the time-consuming and complicated process of re-meshing to fit the mesh to moving interfaces with complex geometry. It also allows us to use uniform meshes which might be very attractive for some applications. To obtain the desired convergence rate, we choose to work with the finite element function Q_1/Q_0 which is second order accurate for velocity and first order accurate for pressure. This finite element function will be modified accordingly on interface elements in order to construct the immersed finite element space and the IFE particular functions. The primal continuous formulation exhibits instability when the finite element function Q_1/Q_0 is used [23]. However, this element is shown to be stable when the discontinuous Galerkin (DG) method is used [120]. Thus, we choose to use the DG method with the IFE Q_1/Q_0 spaces. This choice is not solely based on the stability restriction, but also for the following properties of the DG method:

• It does not require continuity across element boundaries.

• It can be locally conservative.

• It is well suited to solve problems on locally refined meshes with hanging nodes.

• It exhibits strong superconvergence that can be used to estimate the discretization error.

• It has a simple communication pattern between elements with a common face that makes it useful for parallel computation.

• It can handle problems with complex geometries to higher order.

In fact, the first feature is very attractive since the IFE basis functions are not continuous across element boundaries as shown in section 3.1. Thus the DG method is useful in this situation and is a natural choice.

The immersed finite element method shares the basic idea with the immersed interface method. However, the immersed interface method is only second order accurate for both velocity and pressure. Its extension to higher order approximations seems to be complicated, if not impossible. The IFE method, on the other hand, can be extended to higher order approximations, which has already been done for elliptic problems [4, 20]. Moreover, using the DG method allows for efficient adaptive mesh refinement which reduces the computational time. In fact, tracking the interface efficiently is an important task to avoid any instability and can only be done if our numerical approximation is accurate next to the interface, which implies that a fine mesh may be needed around the interface. The adaptive mesh refinement provides a very effective tool to refine the mesh only near the interface.

1.5 Outline of the dissertation

At the end of section 4.3 in [44], Cockburn *et al.* stated the following: " These experiments justify our contention that the optimal order of convergence in $\mathbf{q}$ can be reached if the boundary conditions are piecewise polynomials of degree k. Our theoretical analysis does not explain this phenomenon". Extensive computations in [3] suggest that the md-LDG solution of elliptic problems obtained by interpolating Dirichlet boundary conditions at Radau points is $O(h^{k+1})$ superconvergent for both the solution gradient and potential. In Chapter 2, we investigate the local discontinuous Galerkin method applied to an elliptic problem and carry out an error analysis to prove that, by approximating Dirichlet boundary conditions with appropriate projections or interpolations, both the md-LDG solution and its gradient on Cartesian meshes are $O(h^{k+1})$ convergent.

Next, we focus on developing IFE methods to solve Stokes interface problems. In Chapter 3, we construct IFE spaces and IFE particular functions based on Q_1/Q_0 finite element functions on Cartesian meshes for Stokes interface problems. The discussion covers both Lagrange type and null space based basis functions. Then, we investigate properties of the Lagrange type IFE spaces. Fundamental properties such as partition of unity, trace inequalities, and inverse inequalities are established. In Chapter 4, we present numerical schemes based on the previously developed IFE spaces and particular functions to solve the Stokes interface problem. The optimal rates of convergence of the IFE schemes will be validated through numerical experiments. In Chapter 5, we extend our method to the Stokes problem with a moving interface, where we discuss how the interfaces can be tracked using Runge-Kutta methods and we test our method using two benchmark problems. In Chapter 6, we construct an IFE space, IFE particular functions and IFE schemes to solve the axisymmetric three-dimensional Stokes interface problem. Several numerical results are provided to demonstrate the performance of these schemes. We also present an algorithm for solving the axisymmetric Stokes problem with a moving interface. In Chapter 7, we briefly discuss our results and finding and suggest future research directions.

Chapter 2

Local discontinuous Galerkin method

The LDG finite element method for solving one-dimensional convection-diffusion partial differential equations was introduced by Cockburn and Shu [48] and is based on the work by Bassi and Rebay [17]. Castillo *et al.* [35] presented the first *a priori* error analysis for the LDG method applied to a model elliptic problem. They considered arbitrary meshes with hanging nodes and elements of various shapes and studied general numerical fluxes. They showed that, for smooth solutions, the L^2 errors in u and ∇u, respectively, are of order $k+1/2$ and k when polynomials of total degree not exceeding k are used. Later, Cockburn *et al.* [44] presented a superconvergence result for the LDG method for a model elliptic problem on Cartesian meshes. They identified a special numerical flux for which the L^2 norms of the solution gradient and the potential are of orders $k+1/2$ and $k+1$, respectively, when tensor product polynomials of degree at most k are used.

An optimal minimal dissipation LDG (md-LDG) method for one-dimensional convection-diffusion problems was first investigated in [36]. Later, the method was extended to two-dimensions on triangular meshes in [42] where the authors proved $O(h^{k+1})$ and $O(h^k)$ L^2 convergence rates, respectively, for the potential and solution gradient using the complete k-degree polynomial space P_k.

2.1 Problem formulation and notations

Consider the following second-order elliptic boundary-value problem

$$-\Delta u = f, \quad in\ \Omega = [-1,1]^2, \tag{2.1a}$$

$$u = g_D, \quad on\ \partial\Omega_D, \tag{2.1b}$$

$$\nabla u \cdot \mathbf{n} = \mathbf{g}_N \cdot \mathbf{n}, \quad on\ \partial\Omega_N, \tag{2.1c}$$

for some given functions g_D and $\mathbf{g}_N$, where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\mathbf{n}$ is the unit outer normal vector on $\partial\Omega$ and the measure of $\partial\Omega_D$ is nonzero.
In our analysis, we select the boundary conditions and the source term $f(x, y)$ such that the exact solution, $u(x, y)$, is a smooth function. In order to construct the LDG formulation of Cockburn *et al.* [44], we first introduce an auxiliary variable $\mathbf{q} = \nabla u$ and transform the elliptic problem (2.1) to the following system of first-order differential equations:

$$\mathbf{q} = \nabla u \quad \text{in } \Omega, \tag{2.2a}$$
$$-\nabla \cdot \mathbf{q} = f \quad \text{in } \Omega, \tag{2.2b}$$
$$u = g_D \quad \text{on } \partial\Omega_D, \tag{2.2c}$$
$$\mathbf{q} \cdot \mathbf{n} = \mathbf{g}_N \cdot \mathbf{n} \quad \text{on } \partial\Omega_N. \tag{2.2d}$$

Throughout the dissertation, $\Omega \in \mathbb{R}^2$ denotes a bounded open domain. Also, we let $\mathcal{D}(\Omega)$ be the space of C^∞ functions with compact support in Ω. We denote by $\mathcal{D}'(\Omega)$ the dual space of $D(\Omega)$. We also define the weak derivative D^α as follows: Given a multi-index $\alpha = (\alpha_1, \alpha_2)$ and $|\alpha| = \alpha_1 + \alpha_2$,

$$D^\alpha v(\phi) = (-1)^{|\alpha|} \int_\Omega \frac{v \partial^{|\alpha|}\phi}{\partial x^{\alpha_1} \partial y^{\alpha_2}}, \quad \forall \phi \in \mathcal{D}(\Omega).$$

We use the latter to define the norm $||\cdot||_{k,p,\Omega}$ as follows

$$||v||_{k,p,\Omega} = \left(\sum_{|\alpha| \le k} ||D^\alpha v||^p_{0,p,\Omega} \right)^{1/p},$$

where

$$||v||^p_{0,p,\Omega} = \int_\Omega |v|^p dX,$$

and $1 \le p < \infty$. In the case $p = \infty$, we define

$$||v||_{k,\infty,\Omega} = \max_{|\alpha| \le k} ||D^\alpha v||_{0,\infty,\Omega},$$

where

$$||v||_{0,\infty,\Omega} = \operatorname{ess\,sup}\{|v(X)| : X \in \Omega\}.$$

We, then, define the Sobolev space

$$W^{k,p}(\Omega) = \{v : ||v||_{k,p,\Omega} < \infty\},$$

and we call $||\cdot||_{k,p,\Omega}$ the Sobolev norm associated with the Sobolev space $W^{k,p}$. Also, we define the Sobolev semi-norm $|\cdot|_{k,p,\Omega}$ associated with the $W^{k,p}$ space as $|v|_{k,p,\Omega} = \sum_{|\alpha|=k} ||D^\alpha v||_{0,p,\Omega}$. In the case $p = 2$, the sobolev space becomes a Hilbert space and we denote it by $H^k(\Omega) = W^{k,2}(\Omega)$. For simplicity reasons, we omit the index 2 in the associated norm and semi-norm, *i.e* $||v||_{k,2,\Omega} = ||v||_{k,\Omega}$ and $|v|_{k,2,\Omega} = |v|_{k,\Omega}$.

Finally, we define the norm $||v||_{0,e}$ over an arbitrary line $e \in \Omega$ as

$$||v||_{0,e} = \int_e |v|^2 ds.$$

2.2 The md-LDG method

We partition our domain Ω into a rectangular mesh $\mathscr{T}$ consisting of $N = n \times n$ elements $K_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, $i, j = 0, 1, \ldots, n-1$, where $x_i = y_i = -1 + ih$, $i = 0, 1, \ldots, n$ and $h = 2/n$. Now we multiply (2.2a) and (2.2b) by arbitrary smooth test functions v and $\mathbf{r}$, respectively, integrate over an arbitrary element K, and apply the divergence theorem to write

$$\int_K \mathbf{q} \cdot \mathbf{r} dx = -\int_K u \nabla \cdot \mathbf{r} dx + \int_{\partial K} u \mathbf{r} \cdot \mathbf{n} ds, \tag{2.3}$$

$$\int_K \mathbf{q} \cdot \nabla v dx = \int_K f v dx + \int_{\partial K} v \mathbf{q} \cdot \mathbf{n} ds. \tag{2.4}$$

Let $Q_k(K)$ denote the tensor product space on K consisting of polynomials where the degree in each variable does not exceed k, and define

$$M_N := \{\mathbf{q} \in \left(L^2(\Omega)\right)^2 \quad : \quad \mathbf{q}|_K \in Q_k(K)^2\}, \tag{2.5}$$

$$V_N := \{u \in L^2(\Omega) \quad : \quad u|_K \in Q_k(K)\}. \tag{2.6}$$

The LDG formulation consists of finding $(\mathbf{q}_N, u_N) \in M_N \times V_N$ that satisfies

$$\int_K \mathbf{q}_N \cdot \mathbf{r} dx = -\int_K u_N \nabla \cdot \mathbf{r} dx + \int_{\partial K} \hat{u}_N \mathbf{r} \cdot \mathbf{n} ds, \tag{2.7a}$$

$$\int_K \mathbf{q}_N \cdot \nabla v dx = \int_K f v dx + \int_{\partial K} v \hat{\mathbf{q}}_N \cdot \mathbf{n} ds, \tag{2.7b}$$

for all $(\mathbf{r}, v) \in M_N \times V_N$ and for all $K \in \mathscr{T}$. Here $\mathbf{n}$ is the unit outward normal vector to the edges ∂K and the numerical fluxes $\hat{\mathbf{q}}_N$ and $\hat{u}_N$ are the discrete approximations of the traces of $\mathbf{q}$ and u on element boundaries.

In order to complete the definition of the md-LDG method we need to select the fluxes $\hat{\mathbf{q}}_N$ and $\hat{u}_N$ on ∂K. Let the mean value $\{\cdot\}$ and jump $[\![\cdot]\!]$ of a scalar function u_N and a vector $\mathbf{q}_N$ at (x, y) on an edge of ∂K be defined as

$$\{u\}(x,y) = \frac{1}{2}(u^+(x,y) + u^-(x,y)), \quad \{\mathbf{q}\}(x,y) = \frac{1}{2}(\mathbf{q}^+(x,y) + \mathbf{q}^-(x,y)), \tag{2.8}$$

$$[\![u]\!](x,y) = (u^+(x,y) - u^-(x,y))\mathbf{n}, \quad [\![\mathbf{q}]\!](x,y) = (\mathbf{q}^+(x,y) - \mathbf{q}^-(x,y)) \cdot \mathbf{n}, \tag{2.9}$$

where u^+ is the limit of the solution on K and u^- is the limit of the solution of an adjacent element sharing ∂K, *i.e.,* for $(x,y) \in \partial K$, and $\mathbf{n}$ is the outer unit normal on ∂K we have

$$u^+(x,y) = \lim_{\epsilon \to 0^-} u((x,y) + \epsilon \mathbf{n}), \quad u^-(x,y) = \lim_{\epsilon \to 0^+} u((x,y) + \epsilon \mathbf{n}). \tag{2.10}$$

We follow [44] to define the numerical fluxes $\hat{u}$ and $\hat{\mathbf{q}}$ on interior edges by introducing an auxiliary vector $\mathbf{v}$ and write

$$\hat{\mathbf{q}} = \{\mathbf{q}\} - \mathbf{C}_{12}[\![\mathbf{q}]\!], \tag{2.11}$$
$$\hat{u} = \{u\} + \mathbf{C}_{12} \cdot [\![u]\!]. \tag{2.12}$$

The parameter $\mathbf{C}_{12}$ is defined on each edge of ∂K by

$$\mathbf{C}_{12} \cdot \mathbf{n} = \frac{1}{2}\text{sign}(\mathbf{v} \cdot \mathbf{n}). \tag{2.13}$$

We note that the vector $\mathbf{v}$ is an arbitrary but fixed vector with nonzero components as illustrated in Figure 2.1. Without loss of generality, from now on we assume that $\mathbf{v}$ has strictly positive components, thus, $\hat{u}$ on a horizontal edge is the limit of u from below while on a vertical edge it is the limit of u from the left. $\hat{\mathbf{q}}$ on a horizontal edge is the limit of $\mathbf{q}$ from above while $\hat{\mathbf{q}}$ on a vertical edge is the limit from the right. We also define

$$\partial\Omega^- = \{(x,y) \in \partial\Omega \mid \mathbf{v} \cdot \mathbf{n} < 0\}, \quad \partial\Omega^+ = \{(x,y) \in \partial\Omega \mid \mathbf{v} \cdot \mathbf{n} > 0\}. \tag{2.14}$$

If $\partial\Omega_D^{\pm} = \partial\Omega_D \cap \partial\Omega^{\pm}$, we let $\mathscr{E}$, $\mathscr{E}_D$, $\mathscr{E}_N$, $\mathscr{E}^{\pm}$, $\mathscr{E}_D^{\pm}$ respectively, denote the sets of all edges in $\partial\Omega$, $\partial\Omega_D$, $\partial\Omega_N$, $\partial\Omega^{\pm}$, $\partial\Omega_D^{\pm}$. Finally the set of all interior edges is defined as $\mathcal{E}_I = \mathcal{E} \backslash (\mathcal{E}_D \cup \mathcal{E}_N)$.

Letting C_{11} be a positive constant we define the numerical flux on the boundary as

$$\hat{\mathbf{q}}_N = \begin{cases} \mathbf{q}_N^+ & \text{if } e \in \mathcal{E}_D^- \\ \mathbf{q}_N^+ - C_{11}(u_N^+ - g_D) & \text{if } e \in \mathcal{E}_D^+ \\ \mathbf{g}_N & \text{if } e \in \mathcal{E}_N \end{cases}, \tag{2.15}$$

$$\hat{u}_N = \begin{cases} u_N^+ & \text{if } e \in \mathcal{E}_N \\ \mathscr{P}^+ g_D & \text{if } e \in \mathcal{E}_D \end{cases}, \tag{2.16}$$

where $\mathscr{P}^+ = \mathscr{P}_1^+$ if the edge e is parallel to the $x_1 - axis$ and $\mathscr{P}^+ = \mathscr{P}_2^+$ if e is parallel to the $x_2 - axis$ as defined below.

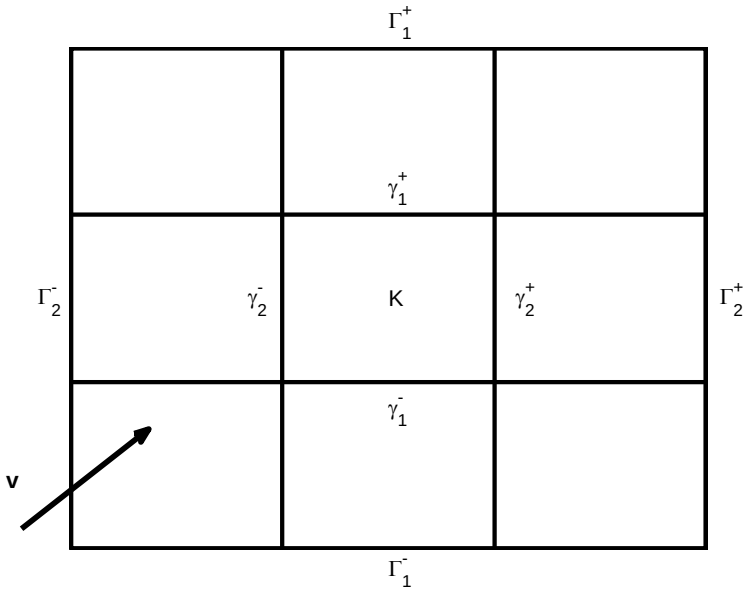


Figure 2.1: An example of a Cartesian mesh $\mathscr{T}$ and a vector $\mathbf{v}$ to define mesh orientation with $\partial\Omega^{\pm} = \Gamma_1^{\pm} \cup \Gamma_2^{\pm}$.

2.3 Error analysis

Let $I = [a^-, a^+]$ be an arbitrary interval, and let $P_k(I)$ be the space of polynomials of degree at most k on I, the projection $\mathscr{P}^{\pm}w \in P_k(I)$ of w is determined by the following $k+1$ conditions

$$\int_I [w(x) - \mathscr{P}^{\pm}w(x)]p(x)dx = 0 \quad \forall\, p \in P_{k-1}(I), \quad \mathscr{P}^{\pm}w(a^{\pm}) = w(a^{\pm}).$$

If $\mathscr{P}$ denotes the L^2 projection onto $P_k(I)$, then on a rectangle $K = I_1 \times I_2$ with $v \in C^0(\bar{K})$ and $\mathbf{r} = (r_1, r_2)^T$, we define $\mathbf{\Pi}^{\pm}\mathbf{r} = (\mathscr{P}_1^{\pm} \otimes \mathscr{P}_2 r_1, \mathscr{P}_1 \otimes \mathscr{P}_2^{\pm} r_2)^T$ and $\pi^{\pm}u = \mathscr{P}_1^{\pm} \otimes \mathscr{P}_2^{\pm}u$, with the subscripts in $\mathscr{P}$ and $\mathscr{P}^{\pm}$ indicate the one-dimensional operators.
Since $\mathbf{v}$ has strictly positive components we use the projections

$$\mathbf{\Pi q}|_K := \mathbf{\Pi}^- \mathbf{q}|_K, \quad \pi u|_K := \pi^+ u|_K, \quad \forall K \in \mathscr{T}, \tag{2.17}$$

having the properties

$$\int_K [\mathbf{r} - \mathbf{\Pi}^- \mathbf{r}] \cdot \nabla v dx = 0, \qquad \forall\, v \in Q^k(K) \tag{2.18a}$$

$$\int_{\gamma_i^-} [\mathbf{r} - \mathbf{\Pi}^- \mathbf{r}] \cdot \mathbf{n} v ds = 0 \qquad \forall\, v \in Q^k(\gamma_i^-),\ i = 1, 2 \tag{2.18b}$$

$$(u - \pi^+ u)|_{\gamma_i^+} = u|_{\gamma_i^+} - \mathscr{P}_i^+ u|_{\gamma_i^+},\ i = 1, 2. \tag{2.18c}$$

In our analysis we need the *a priori* error estimates stated in the following Lemma.

Lemma 2.3.1. *Let $v \in H^{s+2}(K)$ and $\mathbf{r} \in (H^{s+1}(K))^2$, $s \geq 0$. Then for $m = 0, 1$, we have*

$$|v - \pi^{\pm} v|_{m,K} \leq C h^{\min\{s+1,k\}+1-m} ||v||_{s+2,K}, \tag{2.19a}$$

$$||v - \pi^{\pm} v||_{0,e} \leq C h^{\min\{s+1,k\}+\frac{1}{2}} ||v||_{s+2,K}, \qquad \forall\ e \in \partial K, \tag{2.19b}$$

$$|\mathbf{r} - \mathbf{\Pi}^{\pm} \mathbf{r}|_{m,K} \leq C h^{\min\{s,k\}+1-m} ||\mathbf{r}||_{s+1,K}, \tag{2.19c}$$

$$||\mathbf{r} - \mathbf{\Pi}^{\pm} \mathbf{r}||_{0,e} \leq C h^{\min\{s,k\}+\frac{1}{2}} ||\mathbf{r}||_{s+1,K}, \qquad \forall\ e \in \partial K. \tag{2.19d}$$

Furthermore, for any edge e_i parallel to the x_i-axis, $i = 1, 2$, we have

$$||w - \mathscr{P}_i^{\pm} w||_{0,e_i} \leq C h^{\min\{s+\frac{1}{2},k\}+1} ||w||_{s+\frac{3}{2},e_i}, \qquad \forall\ w \in H^{s+\frac{3}{2}}(e_i). \tag{2.19e}$$

If $v \in W^{s+1,\infty}(K)$, then

$$||v - \pi^{\pm} v||_{L^{\infty}(e)} \leq C h^{\min\{s,k\}+1} ||v||_{W^{s+1,\infty}(K)}, \qquad \forall\ e \in \partial K. \tag{2.19f}$$

Let $\mathscr{R}^+ v$ be the k-degree polynomial interpolating v at the roots of the $(k+1)$-degree right Radau polynomial shifted to the edge e and $v \in H^{k+2}(e)$ then

$$||\mathscr{P}^+ v - \mathscr{R}^+ v||_{0,e} \leq C h^{k+2} ||v||_{k+2,e}. \tag{2.20}$$

Proof. See [35, 44] for proofs. For the sake of completeness, we prove the result (2.20). Note that this result was established in [2].
First define the projection $\hat{\pi} u \in P_{k+1}(K)$, which interpolates u at the roots $\{x_{ij}\}$ of right Radau polynomials of degree k and at a point $\bar{x} \neq x_{ij}$ that lies in K. Thus

$$\mathscr{P}^+ \hat{\pi} u = \mathscr{R}^+ \hat{\pi} u = \pi u.$$

Using the standard interpolation error estimates, we obtain

$$||u - \hat{\pi} u||_e \leq C h^{k+2} ||u||_{k+2,e}. \tag{2.21}$$

Next we rewrite,

$$u = u - \hat{\pi} u + \hat{\pi} u.$$

By applying $\mathscr{P}^+$ to the latter and noting that $\mathscr{R}^+ u = \mathscr{P}^+ u$, $\forall u \in P_{k+1}$, (*i.e.* $\mathscr{P}^+ \hat{\pi} u = \mathscr{R}^+ u$) we obtain

$$\begin{aligned} \mathscr{P}^+ u &= \mathscr{P}^+ (u - \hat{\pi} u) + \mathscr{P}^+ \hat{\pi} u \\ &= \mathscr{P}^+ (u - \hat{\pi} u) + \mathscr{R}^+ u. \end{aligned}$$

This yields

$$\mathscr{P}^+ u - \mathscr{R}^+ u = \mathscr{P}^+ (u - \hat{\pi} u). \tag{2.22}$$

Next, we show that

$$||\mathscr{P}^+u||_e \leq C||u||_e \tag{2.23}$$

by writing

$$\begin{aligned} ||\mathscr{P}^+u||_e &= ||\mathscr{P}^+u - u + u||_e \\ &\leq ||\mathscr{P}^+u - u||_e + ||u||_e \\ &\leq Ch^{k+1}||u||_e + ||u||_e \\ &\leq C||u||_e. \end{aligned}$$

Taking the L^2 norm of (2.22), we obtain

$$||\mathscr{P}^+u - \mathscr{R}^+u||_e = ||\mathscr{P}^+(u - \hat{\pi}u)||_e.$$

Combining the standard interpolation (2.21) with the latter result yields (2.20). □

For simplicity, the analysis is done for shape regular uniform Cartesian meshes and the extension to non uniform meshes is straight forward. Furthermore, the analysis will only consider domains Ω such that if $f \in L^2(\Omega)$ and the boundary data are zero, then $u \in H^2(\Omega)$ and $||u||_2 \leq C||f||_0$.

Summing (2.7a) and (2.7b) over all elements the discrete LDG formulation consists of finding $(\mathbf{q}_N, u_N) \in M_N \times V_N$ such that

$$A(\mathbf{q}_N, u_N; \mathbf{r}, v) = \mathscr{F}(\mathbf{r}, v), \quad \forall (\mathbf{r}, v) \in M_N \times V_N, \tag{2.24a}$$

where

$$A(\mathbf{q}, u; \mathbf{r}, v) := a(\mathbf{q}, \mathbf{r}) + b(u, \mathbf{r}) - b(v, \mathbf{q}) + c(u, v), \qquad \mathscr{F}(\mathbf{r}, v) := F(\mathbf{r}) + G(v),$$

and

$$a(\mathbf{q}, \mathbf{r}) = \int_\Omega \mathbf{q} \cdot \mathbf{r} dx,$$

$$b(u, \mathbf{r}) = \sum_{K \in \mathscr{T}} \int_K u \nabla \cdot \mathbf{r} dx - \sum_{e \in \mathcal{E}_I} \int_e (\{u\} + \mathbf{C}_{12}[\![u]\!])[\![\mathbf{r}]\!] ds - \sum_{e \in \mathcal{E}_N} \int_e u \mathbf{r} \cdot \mathbf{n} ds,$$

$$c(u, v) = \sum_{e \in \mathcal{E}_D^+} \int_e C_{11} uv ds$$

$$F(\mathbf{r}) = \sum_{e \in \mathcal{E}_D} \int_e \mathscr{P}^+ g_D \mathbf{r} \cdot \mathbf{n} ds,$$

$$G(v) = \int_K f v dx + \sum_{e \in \mathcal{E}_D^+} \int_e C_{11} v g_D ds + \sum_{e \in \mathcal{E}_N} \int_e v \mathbf{g}_N \cdot \mathbf{n} ds.$$

We note the use of $\mathscr{P}^+ g_D$ in the right hand side F is essential for obtaining superconvergence of the solution gradient $\mathbf{q}$ everywhere. However, The use of either of $\mathscr{P}^+ g_D$ or g_D in the right hand side G yields superconvergence for $\mathbf{q}$.

Furthermore, we note that the true solution satisfies

$$A(\mathbf{q},u;\mathbf{r},v) = a(\mathbf{q},\mathbf{r}) + b(u,\mathbf{r}) - b(v,\mathbf{q}) + c(u,v) = \int_K fv dx + \int_{\partial\Omega_D^+} C_{11} g_D v ds + \int_{\partial\Omega_D} g_D \mathbf{r}\cdot\mathbf{n} ds + \int_{\partial\Omega_N} v\mathbf{g}_N\cdot\mathbf{n} ds, \quad \forall(\mathbf{r},v) \in \left(H^1(\Omega)\right)^2 \times H^2(\Omega). \tag{2.25}$$

Subtracting (2.24a) from (2.25), and letting $\xi_u = u - \pi u$, $\xi_{\mathbf{q}} = \mathbf{q} - \mathbf{\Pi q}$, $e_u = u - u_N$, $\mathbf{e_q} = \mathbf{q} - \mathbf{q}_N$, $\pi e_u = \pi u - u_N$ and $\mathbf{\Pi e_q} = \mathbf{\Pi q} - \mathbf{q}_N$, the following LDG orthogonality condition holds for all $(\mathbf{r}, v) \in M_N \times V_N$

$$A(\mathbf{e_q}, e_u; \mathbf{r}, v) - \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{r}\cdot\mathbf{n} ds = 0.$$

In the remainder of this manuscript we use semi norm

$$|(\mathbf{q},u)|_A^2 = A(\mathbf{q},u;\mathbf{q},u) = ||\mathbf{q}||_0^2 + C_{11}||u||_{0,\partial\Omega_D^+}^2, \tag{2.26}$$

and assume

$$\partial\Omega_N \cap \partial\Omega^- = \emptyset.$$

We further define

$$Z_K(\mathbf{r},u) = \int_K \xi_u \nabla\cdot\mathbf{r} dx - \int_{\gamma_1^-\cup\gamma_1^+} (u - \mathscr{P}_1^+ u)\mathbf{r}\cdot\mathbf{n} ds - \int_{\gamma_2^-\cup\gamma_2^+} (u - \mathscr{P}_2^+ u)\mathbf{r}\cdot\mathbf{n} ds, \tag{2.27}$$

and recall a lemma established in [44] and give the proof for the sake of completeness.

Lemma 2.3.2. *Assume $u \in H^{k+2}(\Omega)$ and $\mathbf{r} \in (H^1(\Omega))^2$. Let π be the operator defined in (2.17) and $Z_K(\mathbf{r},u)$ be the functional defined in (2.27). Then we have*

$$|Z_K(\mathbf{r},u)| \le Ch^{k+1}||u||_{k+2,K}||\mathbf{r}||_{0,K}. \tag{2.28}$$

Proof. We can write

$$Z_K(\mathbf{r},u) = Z_{K,1}(\mathbf{r},u) + Z_{K,2}(\mathbf{r},u),$$

where

$$Z_{K,1}(r_1,u) = \int_K (u - \pi^+ u)\frac{\partial r_1}{\partial x_1} dx_1 dx_2 - \int_{\gamma_2^+} (u - \mathscr{P}_2^+ u) r_1 dx_2 + \int_{\gamma_2^-} (u - \mathscr{P}_2^+ u) r_1 dx_2$$

and

$$Z_{K,2}(r_2,u) = \int_K (u - \pi^+ u)\frac{\partial r_2}{\partial x_2} dx_1 dx_2 - \int_{\gamma_1^+} (u - \mathscr{P}_1^+ u) r_2 dx_1 + \int_{\gamma_1^-} (u - \mathscr{P}_1^+ u) r_2 dx_1.$$

The proof of the approximation results for $Z_{K,1}$ and $Z_{K,2}$ are analogous; therefore, we just present the proof for $Z_{K,1}$. First we consider $Z_{K,1}$ on the reference square $[-1,1]^2$. Then we prove

$$Z_{K,1}(r_1,u)=0 \quad \forall u\in P^{k+1}(K), r_1\in Q^k(K).$$

We then fix $r_1\in Q^k(K)$. Since π^+ and $\mathscr{P}^+$ are polynomial preserving operators, the equality holds true for every $u\in Q^k(K)$. Therefore, we just have to consider the cases $u(x_1,x_2)=x_1^{k+1}$ and $u(x_1,x_2)=x_2^{k+1}$.

Let us start with $u(x_1,x_2)=x_1^{k+1}$. On γ_2^+ we have $u=\mathscr{P}_2^+u=1$, and on γ_2^- we have $u=\mathscr{P}_2^+u=(-1)^{k+1}$. Since $\frac{\partial r_1}{\partial x_1}$ is a polynomial of degree at most k in x_1, we obtain

$$\int_K(u-\pi^+u)\frac{\partial r_1}{\partial x_1}dx_1dx_2=\int_K(u-\mathscr{P}_1^+u)\frac{\partial r_1}{\partial x_1}dx_1dx_2=0.$$

Thus, $Z_{K,1}(r_1,u)=0$ for $u(x_1,x_2)=x_1^{k+1}$.

In the case $u(x_1,x_2)=x_2^{k+1}$, we integrate by parts and obtain

$$\int_K(u-\pi^+u)\frac{\partial r_1}{\partial x_1}dx_1dx_2=\int_K\frac{\partial(u-\pi^+u)}{\partial x_1}r_1dx_1dx_2$$

$$+\int_{\gamma_2^+}(u-\mathscr{P}_2^+u)r_1dx_1-\int_{\gamma_2^-}(u-\mathscr{P}_2^+u)r_1dx_2.$$

Since $\frac{\partial(u-\pi^+u)}{\partial x_1}=0$ and due to the special form of u, we conclude that $Z_{K,1}(r_1,u)=0$ also for $u(x_1,x_2)=x_2^{k+1}$. The proof of (2.28) is completed using a scaling argument. □

The next lemma contains preliminary results needed in our analysis.

Lemma 2.3.3. *Let $u\in H^{k+2}(\Omega)$, $k\geq 0$, and set $\mathbf{q}=\nabla u$. Assume further that $\mathbf{q}\in(W^{k+1,\infty}(\Omega))^2$ and that C_{11} is a positive constant, and let $\mathbf{\Pi}$ and π be the operators defined in (2.17). Then, for all $(\mathbf{r},v)\in M_N\times V_N$, we have*

$$|A(-\xi_{\mathbf{q}},-\xi_u;\mathbf{r},v)+\sum_{e\in\mathcal{E}_D}\int_e(g_D-\mathscr{P}^+g_D)\mathbf{r}\cdot\mathbf{n}ds|\leq Ch^{k+1}|(\mathbf{r},v)|_A,$$

Furthermore, if $\varphi\in H^2$ such that $\varphi|_{\partial\Omega_D}=0$, and $\Phi=\nabla\varphi$ we have

$$|A(\mathbf{r},v;\mathbf{\Phi}-\mathbf{\Pi\Phi},\varphi-\pi\varphi)|\leq C|(\mathbf{r},v)|_A||\varphi||_2,$$

where $C>0$ is a generic constant independent of $h,v,\mathbf{r}$.

Proof. First, we write

$$A(-\xi_{\mathbf{q}}, -\xi_u; \mathbf{r}, v) + \sum_{e\in\mathcal{E}_D}\int_e (g_D - \mathscr{P}^+ g_D)\mathbf{r}\cdot\mathbf{n}ds$$

$$= a(-\xi_{\mathbf{q}}, \mathbf{r}) + b(-\xi_u, \mathbf{r}) - b(v, -\xi_{\mathbf{q}}) + c(-\xi_u, v) + \sum_{e\in\mathcal{E}_D}\int_e (g_D - \mathscr{P}^+ g_D)\mathbf{r}\cdot\mathbf{n}ds$$
$$= T_1 + T_2 + T_3 + T_4,$$

where

$$\begin{aligned}
T_1 &= a(-\xi_{\mathbf{q}}, \mathbf{r}),\\
T_2 &= -b(v, -\xi_{\mathbf{q}}),\\
T_3 &= b(-\xi_u, \mathbf{r}) + \sum_{e\in\mathcal{E}_D}\int_e (g_D - \mathscr{P}^+ g_D)\mathbf{r}\cdot\mathbf{n}ds,\\
T_4 &= -\sum_{e\in\mathcal{E}_D^+}\int_e C_{11}\xi_u v ds.
\end{aligned}$$

Applying Cauchy Schwarz inequality to $T_1 = a(-\xi_{\mathbf{q}}, r)$ leads to

$$\begin{aligned}
|T_1| \le \left|\sum_{K\in\mathscr{T}}\int_K \mathbf{r}.\xi_{\mathbf{q}}dx\right| &\le \left(\sum_{K\in\mathscr{T}}||\mathbf{r}||^2_{0,K}\right)^{1/2}\left(\sum_{K\in\mathscr{T}}||\xi_{\mathbf{q}}||^2_{0,K}\right)^{1/2}\\
&\le C|(\mathbf{r}, v)|_A\left(\sum_{K\in\mathscr{T}} h^{2k+2}||\mathbf{q}||^2_{k+1,K}\right)^{1/2}
\end{aligned} \tag{2.29}$$

Next, we integrate $b(v, -\xi_{\mathbf{q}})$ in (2.24) by parts to write T_2

$$T_2 = -b(v, -\xi_{\mathbf{q}}) = -\sum_{K\in\mathscr{T}}\int_K \nabla v.\xi_{\mathbf{q}}dx + \sum_{e\in\mathcal{E}_D}\int_e v\xi_{\mathbf{q}}.\mathbf{n}ds + \sum_{e\in\mathcal{E}_I}\int_e [\![v]\!]\hat{\xi}_{\mathbf{q}}ds.$$

where $\hat{\xi}_{\mathbf{q}}$ on an interior edge is the limit of $\xi_{\mathbf{q}}$ from the right on a vertical edge and the limit from above on a horizontal edge.

Using the properties (2.18) of the projection $\mathbf{\Pi}^-$, we obtain

$$\int_K \nabla v\cdot\xi_{\mathbf{q}}dx = 0, \quad \forall K\in T, \quad \int_e v\xi_{\mathbf{q}}\cdot\mathbf{n}ds = 0 \quad \forall e\subset\partial\Omega^-,$$

$$\int_e [\![v]\!]\cdot\hat{\xi}_{\mathbf{q}}ds = 0, \quad \forall e\in\mathcal{E}_I.$$

Hence,

$$T_2 = \sum_{e\in\mathcal{E}_D^+}\int_e v\xi_{\mathbf{q}}\cdot\mathbf{n}ds.$$

Multiplying and dividing each term of the sum by $C_{11}^{1/2}$ and applying Cauchy Schwarz inequality, we obtain

$$|T_2| = \left| \sum_{e\in\mathcal{E}_D^+} \int_e v\xi_{\mathbf{q}} \cdot \mathbf{n} ds \right| \leq \left(\sum_{e\in\mathcal{E}_D^+} C_{11}||v||_{0,e}^2 \right)^{1/2} \left(\sum_{e\in\mathcal{E}_D^+} C_{11}^{-1}||\xi_{\mathbf{q}}||_{0,e}^2 \right)^{1/2}$$

$$\text{(by Lemma 2.3.1)} \ \leq C|(\mathbf{r},v)|_A (\sum_{e\in\mathcal{E}_D^+} h^{2k+1}||\mathbf{q}||_{k+1,K}^2)^{1/2}$$

$$\leq C|(\mathbf{r},v)|_A \left(\sum_{e\subset\partial\Omega_D^+} h^{2k+1}h^2||\mathbf{q}||_{W^{k+1,\infty}}^2 \right)^{1/2}$$

$$\leq C|(\mathbf{r},v)|_A \left(\sum_{e\in\mathcal{E}_D^+} h^{2k+3} \right)^{1/2} ||\mathbf{q}||_{W^{k+1,\infty}}$$

$$= C|(\mathbf{r},v)|_A h^{k+1}||\mathbf{q}||_{W^{k+1,\infty}}.$$

For T_3 we write

$$T_3 = b(-\xi_u, \mathbf{r}) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{r}\cdot\mathbf{n} ds =$$

$$-\left(\sum_{K\in\mathscr{T}} \xi_u \nabla\cdot\mathbf{r} dx - \sum_{K\in\mathscr{T}} \left(\sum_{\substack{e\in\mathcal{E}_I \\ e\subset\partial K}} \int_e \hat{\xi}_u \mathbf{r}\cdot\mathbf{n} ds \right) - \sum_{e\in\mathcal{E}_N} \int_e \xi_u \mathbf{r}\cdot\mathbf{n} ds \right) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P} g_D)\mathbf{r}\cdot\mathbf{n} ds,$$

where $\hat{\xi}_u$ on an interior edge is the limit of ξ_u from the left if the edge is vertical and from the bottom if the edge is horizontal.

Using the assumption (2.3) we write

$$\begin{aligned} T_3 &= -\left(\sum_{K\in\mathscr{T}} \xi_u \nabla\cdot\mathbf{r} dx - \sum_{K\in\mathscr{T}} \left(\sum_{\substack{e\in\mathcal{E}_I \\ e\subset\partial K}} \int_e (u - \mathscr{P}^+ u)\mathbf{r}\cdot\mathbf{n} ds \right) - \sum_{e\subset\partial\Omega} \int_e (u - \mathscr{P}^+ u)\mathbf{r}\cdot\mathbf{n} ds \right) \\ &= -\sum_{K\in\mathscr{T}} Z_K(\mathbf{r}, u), \end{aligned} \tag{2.30}$$

where

$$Z_K(\mathbf{r},u) = \int_K \xi_u \nabla\cdot\mathbf{r} dx - \int_{\gamma_1^-\cup\gamma_1^+} (u - \mathscr{P}_1^+ u)\mathbf{r}\cdot\mathbf{n} ds - \int_{\gamma_2^-\cup\gamma_2^+} (u - \mathscr{P}_2^+ u)\mathbf{r}\cdot\mathbf{n} ds, \tag{2.31}$$

with γ_i^+ being the outflow edges and γ_i^- the inflow edges of element K as illustrated in Figure 2.1.

Applying an estimate for Z_K from Lemma 2.3.2 and Cauchy Schwarz inequality we write

$$\sum_{K\in\mathscr{T}} Z_K(\mathbf{r},u) \leq C \sum_{K\in\mathscr{T}} h^{k+1}||u||_{k+2,K}||\mathbf{r}||_{0,K} \leq C|(\mathbf{r},v)|_A h^{k+1}||u||_{k+2}.$$

Thus,

$$|T_3| \leq C|(\mathbf{r},v)|_A h^{k+1}||u||_{k+2}.$$

Applying Cauchy Schwartz inequality we write

$$\begin{aligned}|T_4| &\leq |\sum_{e\in\mathcal{E}_D^+}\int_e C_{11}\xi_u v ds)| \\ &\leq (\sum_{e\in\mathcal{E}_D^+} C_{11}||v||_{0,e}^2)^{1/2}(\sum_{e\in\mathcal{E}_D^+} C_{11}||\xi_u||_{0,e}^2)^{1/2},\end{aligned}$$

leading to

$$|T_4| \leq |(\mathbf{r},v)|_A(\sum_{e\in\mathcal{E}_D^+} C_{11}||\xi_u||_{0,e}^2)^{1/2},$$

which, by Lemma 2.3.1, becomes

$$\begin{aligned}|T_4| &\leq |(\mathbf{r},v)|_A C(\sum_{e\in\mathcal{E}_D^+} h^{2k+1}||u||_{k+2,K}^2)^{1/2} \\ &\leq |(\mathbf{r},v)|_A C h^{k+1}||u||_{W^{k+2,\infty}}.\end{aligned}$$

The proof of (2.3.3) is completed by combining the estimates for T_1, T_2,T_3 and T_4.

Similarly, we establish (2.3.3) by splitting

$$\begin{aligned}A(\mathbf{r},v;\mathbf{\Phi}-\mathbf{\Pi\Phi},\varphi-\pi\varphi) &= a(\mathbf{r},\mathbf{\Phi}-\mathbf{\Pi\Phi}) + b(v,\mathbf{\Phi}-\mathbf{\Pi\Phi}) - b(\varphi-\pi\varphi,\mathbf{r}) + c(v,\varphi-\pi\varphi) \\ &= \tilde{T}_1+\tilde{T}_2+\tilde{T}_3+\tilde{T}_4,\end{aligned}$$

where

$$\tilde{T}_1 = a(\mathbf{r},\xi_{\mathbf{\Phi}}),\ \tilde{T}_2 = -b(v,\xi_{\mathbf{\Phi}}),\ \tilde{T}_3 = b(\xi_\varphi,\mathbf{r}),\ \tilde{T}_4 = \sum_{e\in\mathcal{E}_D^+}\int_e C_{11}\xi_\varphi v ds.$$

We establish the proof by bounding each of terms $\tilde{T}_i, i=1,2,3,4$.

Following (2.29) for T_1 we apply Cauchy Schwarz inequality to $\tilde{T}_1$ and Lemma 2.3.1 to obtain

$$|\tilde{T}_1| \leq |(\mathbf{r},v)|_A \left(\sum_{K\in\mathscr{T}} ||\xi_{\mathbf{\Phi}}||_{0,K}^2\right)^{1/2} \leq |(\mathbf{r},v)|_A||\xi_{\mathbf{\Phi}}||_0 \leq C|(\mathbf{r},v)|_A||\varphi||_2.$$

We follow (2.3) for T_2 and use projection properties (2.18) and Lemma 2.3.1 to write

$$\begin{aligned}|\tilde{T}_2| &= |\sum_{e\in\mathcal{E}_D^+}\int_e v\xi_{\mathbf{\Phi}}\cdot\mathbf{n}ds| \\ &\leq C|(\mathbf{r},v)|_A\left(\sum_{e\in\mathcal{E}_D^+}C_{11}^{-1}||\xi_{\mathbf{\Phi}}||_{0,e}^2\right)^{1/2} \\ &\leq C|(\mathbf{r},v)|_A||\varphi||_2.\end{aligned}$$

Next we will find a bound for $\tilde{T}_3$ as

$$|\tilde{T}_3| = |-\left(\sum_{K\in\mathscr{T}}\xi_\varphi\nabla\cdot\mathbf{r}dx - \sum_{K\in\mathscr{T}}\left(\sum_{e\in\mathcal{E}_I,e\subset\partial K}\int_e\hat{\xi}_\varphi\mathbf{r}\cdot\mathbf{n}ds\right) - \sum_{e\in\mathcal{E}_N}\int_e\xi_\varphi\mathbf{r}\cdot\mathbf{n}ds\right)|,$$

where $\hat{\xi}_\varphi$ is the limit of ξ_φ from the left on a vertical edge and from the bottom on a horizontal edge.

Since $\varphi = 0$ on $\partial\Omega_D$ and $\partial\Omega_N \subset \partial\Omega^+$, $\tilde{T}_3$ can be written as

$$\begin{aligned}\tilde{T}_3 &= -\sum_{K\in\mathscr{T}}Z_K(\mathbf{r},\varphi) \\ &\leq C|(\mathbf{r},v)|_A||\xi_\varphi||_1 \leq C|(\mathbf{r},v)|_A||\varphi||_2.\end{aligned}$$

Finally for $\tilde{T}_4$ we follow (2.3) for T_4 to write

$$\begin{aligned}\tilde{T}_4 &\leq |(\mathbf{r},v)|_A(\sum_{e\in\mathcal{E}_D^+}C_{11}||\xi_\varphi||_{0,e}^2)^{1/2} \\ &\leq C|(\mathbf{r},v)|_A||\varphi||_2. \qquad (2.32)\end{aligned}$$

Combining the bounds for $\tilde{T}_i,\ i = 1,2,3,4$ completes the proof of the lemma. □

The main result of this chapter is summarized in the following theorem where the estimate for $\mathbf{q}-\mathbf{q}_N$ is improved by $\sqrt{h}$ while that for $u-u_N$ is the same as in [44].

Theorem 2.3.4. *Assume that the solution $(\mathbf{q},u)$ of the problem (2.2) is in $(H^{k+1}(\Omega))^2 \times H^{k+2}(\Omega)$ for $k \geq 0$. If $(\mathbf{q}_N,u_N)$ satisfies the LDG weak equations (2.24), then*

$$|(\mathbf{q}-\mathbf{q}_N, u-u_N)|_A \leq Ch^{k+1},$$

and

$$||u-u_N||_0 \leq Ch^{k+1}$$

where the constant $C>0$ is independent of the mesh size h.

Proof. We first prove (2.3.4) which is an improvement of the result in [44] by a factor of $\sqrt{h}$. Since $|(\cdot,\cdot)|_A$ is a semi norm we write

$$|(\mathbf{e_q}, e_u)|_A = |(\mathbf{\Pi e_q} + \xi_\mathbf{q}, \pi e_u + \xi_u)|_A \leq |(\xi_\mathbf{q}, \xi_u)|_A + |(\mathbf{\Pi e_q}, \pi e_u)|_A,$$

where the projection error term

$$|(\xi_\mathbf{q}, \xi_u)|_A = O(h^{k+1}).$$

Next, we establish that $|(\mathbf{\Pi e_q}, \pi e_u)|_A = O(h^{k+1})$ to prove (2.3.4).
Let us write

$$\begin{aligned} |(\mathbf{\Pi e_q}, \Pi e_u)|_A^2 &= A(\mathbf{\Pi e_q}, \pi e_u; \mathbf{\Pi e_q}, \pi e_u) \\ &= A(\mathbf{\Pi q} - \mathbf{q} + \mathbf{e_q}, \pi u - u + e_u; \mathbf{\Pi e_q}, \pi e_u) \\ &= A(\mathbf{\Pi q} - \mathbf{q}, \pi u - u; \mathbf{\Pi e_q}, \pi e_u) + A(\mathbf{e_q}, e_u; \mathbf{\Pi e_q}, \pi e_u) \end{aligned}$$

Applying the LDG orthogonality condition (2.3) with $\mathbf{r} = \mathbf{\Pi e_q}$ and $v = \pi e_u$ we obtain

$$|(\mathbf{\Pi e_q}, \Pi e_u)|_A^2 = A(-\xi_\mathbf{q}, -\xi_u; \mathbf{\Pi e_q}, \pi e_u) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{\Pi e_q} \cdot \mathbf{n} ds.$$

Applying Lemma 2.3.3 yields

$$|A(-\xi_\mathbf{q}, -\xi_u; \mathbf{\Pi e_q}, \pi e_u)) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{\Pi e_q} \cdot \mathbf{n} ds| \leq C h^{k+1} |(\mathbf{\Pi e_q}, \Pi e_u)|_A,$$

which establishes

$$|(\mathbf{\Pi e_q}, \pi e_u)|_A \leq C h^{k+1}.$$

We complete the proof of (2.3.4) by combining (2.3), (2.3) and (2.3).

Next, for $\lambda \in L^2$ we let φ be the solution of the adjoint problem

$$\begin{aligned} -\Delta\varphi &= \lambda \quad \text{in } \Omega, \\ \varphi &= 0 \quad \text{on } \partial\Omega_D \\ \frac{\partial\varphi}{\partial n} &= 0 \quad \text{on } \partial\Omega_N. \end{aligned}$$

If $\mathbf{\Phi} = -\nabla\varphi$ from standard partial differential equations theory we have $||\mathbf{\Phi}||_1 < C||\lambda||_0$.

One can verify that for $\mathbf{s} \in (L^2)^2$ and $w \in L^2$ the following holds

$$A(-\mathbf{\Phi}, \varphi; -\mathbf{s}, w) = (\lambda, w) = \int_\Omega \lambda w dx.$$

Letting $\mathbf{s} = \mathbf{e_q}$, and $w = \lambda = e_u$ we obtain

$$A(-\mathbf{\Phi}, \varphi; -\mathbf{e_q}, e_u) = ||e_u||_0^2,$$

which leads to

$$||e_u||_0^2 = A(\mathbf{e_q}, e_u; \mathbf{\Phi}, \varphi).$$

Applying the orthogonality condition (2.3) yields

$$||e_u||_0^2 = A(\mathbf{e_q}, e_u; \mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{\Pi\Phi}\cdot\mathbf{n}ds.$$

Again we split $\mathbf{e_q} = \mathbf{q} - \mathbf{q}_N = \mathbf{q} - \mathbf{\Pi q} + \mathbf{\Pi q} - \mathbf{q}_N$ and $e_u = u - u_N = u - \pi u + \pi u - u_N$ to obtain

$$||e_u||_0^2 = A(\mathbf{\Pi e_q}, \pi e_u; \mathbf{\Phi}-\mathbf{\Pi\Phi}, \varphi-\pi\varphi)+A(\xi_q, \xi_u; \mathbf{\Phi}-\mathbf{\Pi\Phi}, \varphi-\pi\varphi)+\sum_{e\in\mathcal{E}_D} \int_e (g_D-\mathscr{P}^+ g_D)\mathbf{\Pi\Phi}\cdot\mathbf{n}ds.$$

which can be written as

$$\begin{aligned}
||e_u||_0^2 &= A(\mathbf{\Pi e_q}, \pi e_u; \mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi) \\
&\quad - \left[A(-\xi_q, -\xi_u; \mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)(\mathbf{\Phi} - \mathbf{\Pi\Phi})\cdot\mathbf{n}ds\right] \\
&\quad + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{\Phi}\cdot\mathbf{n}ds \\
&= H_1 + H_2 + H_3. \qquad (2.33)
\end{aligned}$$

where

$$\begin{aligned}
H_1 &= A(\mathbf{\Pi e_q}, \pi e_u; \mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi) \\
H_2 &= -[A(-\xi_q, -\xi_u; \mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi) + \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)(\mathbf{\Phi} - \mathbf{\Pi\Phi})\cdot\mathbf{n}ds], \\
H_3 &= \sum_{e\in\mathcal{E}_D} \int_e (g_D - \mathscr{P}^+ g_D)\mathbf{\Phi}\cdot\mathbf{n}ds.
\end{aligned}$$

Applying Lemma 2.3.3 and the estimate (2.3) we prove that

$$|H_1| \leq C|(\mathbf{\Pi e_q}, \pi e_u)|_A||\varphi||_2 \leq Ch^{k+1}||e_u||_0.$$

By Lemma 2.3.3

$$|H_2| \leq Ch^{k+1}|(\mathbf{\Phi} - \mathbf{\Pi\Phi}, \varphi - \pi\varphi)|_A \leq Ch^{k+1}||\varphi||_2 \leq Ch^{k+1}||e_u||_0.$$

Applying Cauchy Schwarz inequality, the trace theorem [28] and Lemma 2.3.1 we write

$$|H_3| \leq C||g_D - \mathscr{P}^+ g_D||_{0,\partial\Omega_D}\ ||\mathbf{\Phi}||_{0,\partial\Omega_D} \leq C||g_D - \mathscr{P}^+ g_D||_{0,\partial\Omega_D}||\varphi||_2 \leq Ch^{k+1}||e_u||_0.$$

Combining (2.33), (2.3), (2.3) and (2.3) completes the proof of the theorem. □

Instead of the projection $\mathscr{P}^+ g_D$, one could also use the interpolation $\mathscr{R}^+ g_D$ of the boundary condition g_D in (2.16) and still have $O(h^{k+1})$ superconvergence rates for both u and $\mathbf{q}$. The proof will follow the same line of reasoning to prove Lemma 2.3.3 and Theorem 2.3.4. The term T_3 in the proof of Lemma 2.3.3 now contains

$$\sum_{e\in\mathcal{E}_D}\int_e (g_D - \mathscr{R}^+ g_D)\mathbf{r}\cdot\mathbf{n}ds.$$

Adding and subtracting $\sum_{e\in\mathcal{E}_D}\int_e \mathscr{P}^+ g_D\mathbf{r}\cdot\mathbf{n}ds$, we write T_3 as

$$T_3 = -\sum_{K\in\mathscr{T}} Z_k(\mathbf{r},v) + \sum_{e\in\mathcal{E}_D}\int_e (\mathscr{P}^+ g_D - \mathscr{R}^+ g_D)\mathbf{r}\cdot\mathbf{n}ds.$$

Applying the bounds for Z_K used above, the superconvergence result (2.20) and the bound $||\mathbf{r}||_{0,e} \leq C||\mathbf{r}||_{0,K}/\sqrt{h}$ we establish that $|T_3| \leq C|(\mathbf{r},v)|_A h^{k+1}$.

2.4 Numerical results

In order to verify the sharpness of our L^2 *a priori* error estimates in Theorem 2.3.4 we conduct two computational experiments on the standard linear diffusion problem

$$-\Delta u = f, \quad \in \ (-1,1)^2,$$

with Dirichlet boundary conditions and a source term $f(x,y)$ such that the true solution is

$$u(x,y) = \cos(\pi x) + \cos(\pi y).$$

We solve this problem on uniform Cartesian meshes having $N = 16, 36, 64, 100, 144, 196, 256$, 324 ,400 square elements with spaces Q_k, $k = 1,2,3,4$, using the minimal dissipation LDG method with the auxiliary vector $\mathbf{v} = [1,1]^T$, stabilization parameter $C_{11} = 1$, projection $\mathscr{P}^+ g_D$ in (2.16) and g_D in (2.15). In Table 6.1 we present the LDG L^2 errors in u and $\mathbf{q}$ which are in full agreement with Theorem 2.3.4 and obtain $O(h^{k+1})$ convergence rates for both u and its gradient $\mathbf{q}$. We repeat the computational experiment using the true Dirichlet boundary condition g_D in (2.16) and $\mathscr{P}^+ g_D$ in (2.15) and show the LDG L^2 errors for u and $\mathbf{q}$ in Table 6.2. In this case we obtain the suboptimal $O(h^{k+1/2})$ convergence rates for $\mathbf{q}$. This confirms our error analysis that states that $\mathscr{P}^+ g_D$ is only needed in (2.16).

p	N	$\|\|u - u_N\|\|_{0,\Omega}$	Order	$\|\|q_1 - q_{1,N}\|\|_{0,\Omega}$	Order
$k=1$	16	4.4127e-01	NA	8.0770e-01	NA
	36	1.4973e-01	2.6656	3.2685e-01	2.2313
	64	7.5654e-02	2.3729	1.7566e-01	2.1583
	100	4.6725e-02	2.1596	1.0996e-01	2.0992
	144	3.2141e-02	2.0521	7.5482e-02	2.0636
	196	2.3605e-02	2.0026	5.5094e-02	2.0425
	256	1.8119e-02	1.9805	4.2016e-02	2.0295
	324	1.4365e-02	1.9712	3.3115e-02	2.0211
	400	1.1675e-02	1.9680	2.6780e-02	2.0155
$k=2$	16	8.2880e-02	NA	1.0980e-01	NA
	36	2.5288e-02	2.9277	3.5270e-02	2.8009
	64	1.0205e-02	3.1542	1.4991e-02	2.9741
	100	4.9612e-03	3.2324	7.6106e-03	3.0380
	144	2.7357e-03	3.2649	4.3513e-03	3.0664
	196	1.6503e-03	3.2788	2.7064e-03	3.0805
	256	1.0645e-03	3.2838	1.7919e-03	3.0878
	324	7.2300e-04	3.2841	1.2450e-03	3.0916
	400	5.1164e-04	3.2819	8.9873e-04	3.0935
$k=3$	16	6.1615e-03	NA	1.1106e-02	NA
	36	8.5780e-04	4.8628	1.8783e-03	4.3829
	64	2.3538e-04	4.4952	5.5323e-04	4.2490
	100	9.1811e-05	4.2190	2.1883e-04	4.1565
	144	4.3638e-05	4.0796	1.0358e-04	4.1021
	196	2.3498e-05	4.0158	5.5314e-05	4.0698
	256	1.3797e-05	3.9874	3.2210e-05	4.0496
	324	8.6387e-06	3.9751	2.0022e-05	4.0366
	400	5.6856e-06	3.9704	1.3098e-05	4.0277
$k=4$	16	6.3759e-04	NA	8.0053e-04	NA
	36	8.4088e-05	4.9963	1.1493e-04	4.7869
	64	1.8723e-05	5.2215	2.7300e-05	4.9968
	100	5.7411e-06	5.2974	8.8037e-06	5.0716
	144	2.1728e-06	5.3294	3.4706e-06	5.1055
	196	9.5340e-07	5.3436	1.5756e-06	5.1229
	256	4.6672e-07	5.3494	7.9397e-07	5.1325
	324	2.4852e-07	5.3507	4.3350e-07	5.1379
	400	1.4144e-07	5.3494	2.5221e-07	5.1409

Table 2.1: L^2 errors on uniform meshes having $N = 16, 36, 64, 100, 144, 196, 256, 324, 400$ elements and $k = 1, 2, 3, 4$ with $\mathscr{P}^+ g_D$ in (2.16) and g_D in (2.15).

p	N	$\|\|u - u_N\|\|_{0,\Omega}$	Order	$\|\|q_1 - q_{1,N}\|\|_{0,\Omega}$	Order
$k=1$	16	4.6592e-01	NA	8.1159e-01	NA
	36	1.5476e-01	2.7182	3.5707e-01	2.0250
	64	7.6531e-02	2.4478	2.0960e-01	1.8518
	100	4.6716e-02	2.2121	1.4160e-01	1.7575
	144	3.1965e-02	2.0811	1.0373e-01	1.7072
	196	2.3425e-02	2.0165	8.0097e-02	1.6771
	256	1.7969e-02	1.9858	6.4200e-02	1.6568
	324	1.4245e-02	1.9716	5.2911e-02	1.6419
	400	1.1581e-02	1.9656	4.4560e-02	1.6303
$k=2$	16	7.9648e-02	NA	1.3297e-01	NA
	36	2.4826e-02	2.8750	4.4042e-02	2.7252
	64	1.0088e-02	3.1302	1.9528e-02	2.8271
	100	4.9201e-03	3.2180	1.0381e-02	2.8317
	144	2.7179e-03	3.2550	6.2156e-03	2.8131
	196	1.6414e-03	3.2714	4.0433e-03	2.7895
	256	1.0596e-03	3.2780	2.7946e-03	2.7661
	324	7.2008e-04	3.2794	2.0227e-03	2.7446
	400	5.0979e-04	3.2779	1.5178e-03	2.7254
$k=3$	16	6.3200e-03	NA	1.2346e-02	NA
	36	8.7191e-04	4.8852	2.4734e-03	3.9652
	64	2.3655e-04	4.5346	8.4950e-04	3.7149
	100	9.1681e-05	4.2477	3.7923e-04	3.6142
	144	4.3454e-05	4.0951	1.9772e-04	3.5722
	196	2.3373e-05	4.0228	1.1436e-04	3.5521
	256	1.3720e-05	3.9895	7.1269e-05	3.5411
	324	8.5911e-06	3.9745	4.7002e-05	3.5343
	400	5.6554e-06	3.9683	3.2404e-05	3.5298
$k=4$	16	6.2182e-04	NA	1.1241e-03	NA
	36	8.3110e-05	4.9634	1.7174e-04	4.6337
	64	1.8585e-05	5.2065	4.4120e-05	4.7242
	100	5.7104e-06	5.2883	1.5392e-05	4.7191
	144	2.1636e-06	5.3231	6.5369e-06	4.6972
	196	9.5007e-07	5.3389	3.1802e-06	4.6742
	256	4.6532e-07	5.3457	1.7083e-06	4.6538
	324	2.4786e-07	5.3476	9.8947e-07	4.6364
	400	1.4111e-07	5.3468	6.0803e-07	4.6218

Table 2.2: L^2 errors on uniform meshes having $N = 16, 36, 64, 100, 144, 196, 256, 324, 400$ elements and $k = 1, 2, 3, 4$ with g_D in (2.16) and $\mathscr{P}^+ g_D$ in (2.15).

Chapter 3

Q_1/Q_0 Immersed finite element approximations for the Stokes interface problem

In this chapter, we construct two immersed finite element spaces $S_h(\Omega)$ and $\tilde{S}_h(\Omega)$, which will be used to solve the two-dimensional Stokes interface problem (1.3). These spaces are derived using the jump conditions (1.3f) and (1.3g) across the interface. We use two approaches: (i) we apply properties of the standard Lagrange basis functions by enforcing the nodal conditions, (ii) we construct the IFE basis functions by applying solely the jump conditions (1.3f) and (1.3g) and computing the null space of the resulting matrix.

Let $\mathcal{T}_h$ be an arbitrary partition of the domain rectangular Ω and assume that $\mathcal{T}_h$ consists of uniform rectangles. We shall call the elements and edges cut by the interface interface elements and interface edges, respectively and denote their union by the set $\mathcal{T}_h^i$, $\mathcal{E}_h^i$, and $\mathcal{T}_h^n = \mathcal{T}_h \backslash \mathcal{T}_h^i$, $\mathcal{E}_h^n = \mathcal{E}_h \backslash \mathcal{E}_h^i$, respectively. We assume:

(H1) The interface can only intersect an arbitrary element at two edges.

(H2) An arbitrary edge can only intersect the interface at one point or the whole edge is on the interface.

We present the procedure to construct the IFE basis functions. For a typical interface rectangular element $T = \square A_1A_2A_3A_4 \in \mathcal{T}_h^i$ with vertices $A_j = (x_j, y_j)$, $j = 1, 2, 3, 4$ such that the interface Γ intersects two edges at $D = (x_D, y_D)$ and $E = (x_E, y_E)$ referred to as interface points. The interface $\widetilde{DE} = \Gamma \cap T$ is approximated by the line segment $\overline{DE}$ which separates T into two polygonal domains T^+ and T^- such that T^+ contains vertices of T that are in Ω^+. Topologically, there are two types of interface elements: Type I interface elements are those with two adjacent edges cut by the interface and Type II interface elements are with

two opposite edges cut by the interface, see the illustrations in Figure 3.1.

3.1 Space $S_h(\Omega)$

As usual, the construction and analysis of the finite element basis functions are performed on the reference element $\hat{T} = \square\hat{A}_1\hat{A}_2\hat{A}_3\hat{A}_4$ with vertices $\hat{A}_1 = (0,0),\ \hat{A}_2 = (1,0),\ \hat{A}_3 = (0,1),\ \hat{A}_4 = (1,1)$. Let $X = (x,y)$ and $\hat{X} = (\hat{x},\hat{y})$ and let

$$\hat{X} = F(X) = MX + B, \tag{3.1}$$

be the standard affine mapping from an arbitrary element T to the reference element $\hat{T}$ such that $\hat{A}_i = F(A_i),\ i = 1,2,3,4$. We further note that each interface element T of Type I (Type II) is mapped into a reference element of Type I (Type II) shown in Figure 3.1 where $\hat{E} = F(E)$ and $\hat{D} = F(D)$ and $\overline{\hat{D}\hat{E}} = F(\overline{DE})$. The interface points $\hat{E}$ and $\hat{D}$ can be written as

$$\hat{D} = \begin{pmatrix} 0 \\ \hat{d} \end{pmatrix}, \quad \hat{E} = \begin{pmatrix} \hat{e} \\ 0 \end{pmatrix}, \tag{3.2}$$

for an element of Type I and

$$\hat{D} = \begin{pmatrix} \hat{d} \\ 1 \end{pmatrix}, \quad \hat{E} = \begin{pmatrix} \hat{e} \\ 0 \end{pmatrix}, \tag{3.3}$$

for an element of Type II, where $0 < \hat{d}, \hat{e} < 1$ as shown in Figure 3.1.

Now we are ready to describe our procedure for constructing the IFE shape functions on the reference element. Again, a function $\hat{f}(\hat{x},\hat{y})$ defined for $(\hat{x},\hat{y}) \in \hat{T}$ leads to a function $f(x,y) = \hat{f}(F^{-1}(\hat{x},\hat{y}))$ for $(x,y) \in T$ by the affine mapping between the reference element $\hat{T}$ and element T. We first note that the velocity $\mathbf{u}$ and the pressure p are coupled through the jump condition (1.3f), which requires the design of vector-valued shape functions for both $\mathbf{u}$ and p. This means we plan to approximate the solution vector $\hat{\mathbf{U}} = [\hat{\mathbf{u}}, \hat{p}]^T$ by an IFE function $\hat{\mathbf{\Phi}}$ which is a piecewise polynomial vector function of the form

$$\hat{\mathbf{\Phi}}(\hat{x},\hat{y}) = \hat{\mathbf{\Phi}}^s(\hat{x},\hat{y}) = \begin{pmatrix} \hat{\phi}_1^s(\hat{x},\hat{y}) \\ \hat{\phi}_2^s(\hat{x},\hat{y}) \\ \hat{\phi}_3^s(\hat{x},\hat{y}) \end{pmatrix}, \quad \text{for } (\hat{x},\hat{y}) \in \hat{T}^s, \quad s = +,-, \tag{3.4a}$$

where

$$\hat{\phi}_j^s(\hat{x},\hat{y}) = a_j^s + b_j^s\hat{x} + c_j^s\hat{y} + d_j^s\hat{x}\hat{y}, \quad j = 1,2, \quad s = +,-, \tag{3.4b}$$

$$\hat{\phi}_3^s(\hat{x},\hat{y}) = a_3^s, \quad s = +,-. \tag{3.4c}$$

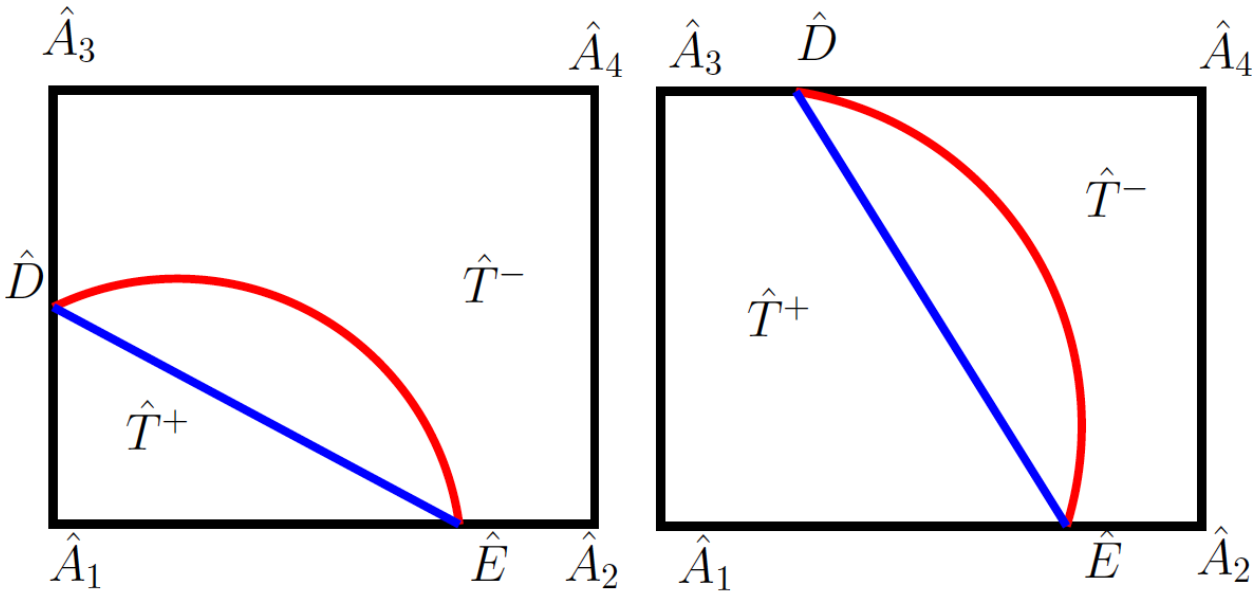


Figure 3.1: Reference interface elements of Type I (left) and Type II (right).

Next we let $\hat{\mathbf{\Theta}} = (\hat{\phi}_1, \hat{\phi}_2)^T$ such that $\hat{\mathbf{\Theta}}|_{T^s}(\hat{x}, \hat{y}) = \hat{\mathbf{\Theta}}^s(\hat{x}, \hat{y}) = \begin{pmatrix} \hat{\phi}_1^s(\hat{x}, \hat{y}) \\ \hat{\phi}_2^s(x, y \end{pmatrix}, \ s = +, -.$

We then discuss the construction of IFE shape functions that will be used to form the local IFE space on the reference element $\hat{T}$. According to (3.4) each IFE function is defined by 18 coefficients $a_j^s, b_j^s, c_j^s, d_j^s$, $j = 1, 2$, $s = +, -$ and a_3^s, $s = +, -$. Hence, we can define IFE shape functions $\hat{\mathbf{\Phi}}_i$, $i = 1, 2, \ldots, 9$ whose coefficients are determined by the following 18 conditions:

- continuity of the velocity component across $\overline{\hat{D}\hat{E}}$ for $\hat{\mathbf{\Theta}}_i = (\hat{\phi}_{1,i}, \hat{\phi}_{2,i})^T$

$$\hat{\mathbf{\Theta}}_i^-(\hat{E}) = \hat{\mathbf{\Theta}}_i^+(\hat{E}), \quad \hat{\mathbf{\Theta}}_i^-(\hat{D}) = \hat{\mathbf{\Theta}}_i^+(\hat{D}), \quad \frac{\partial^2 \hat{\mathbf{\Theta}}_i^-}{\partial \hat{x} \partial \hat{y}} = \frac{\partial^2 \hat{\mathbf{\Theta}}_i^+}{\partial \hat{x} \partial \hat{y}} \tag{3.5a}$$

- weak continuity of the normal stress across $\overline{\hat{D}\hat{E}}$

$$\int_{\overline{\hat{D}\hat{E}}} [\mathbf{S}(\hat{\mathbf{\Theta}}_i, \hat{\phi}_{3,i}) \mathbf{n}_{\overline{\hat{D}\hat{E}}}] ds = 0, \tag{3.5b}$$

  where $\mathbf{n}_{\overline{\hat{D}\hat{E}}}$ is the normal vector to the approximate interface $\overline{\hat{D}\hat{E}}$

- continuity of the divergence of the velocity

$$\hat{\nabla} \cdot \hat{\mathbf{\Theta}}_i^+(\hat{D}) = \hat{\nabla} \cdot \hat{\mathbf{\Theta}}_i^-(\hat{D}) \tag{3.5c}$$

- Lagrange and scaling conditions

$$\hat{\phi}_{1,i}(\hat{A}_j) = \delta_{i,j}, \quad \hat{\phi}_{2,i}(\hat{A}_j) = \delta_{i,j+4}, \ j = 1,2,3,4, \ \textit{and} \ \frac{1}{|\hat{T}|}\int_{\hat{T}} \hat{\phi}_{3,i} = \delta_{i,9}, \tag{3.5d}$$

  where $|\hat{T}|$ is the area of the rectangle $\hat{T}$.

Note that the continuity of the second derivatives in (3.5a) is equivalent to $d_1^+ = d_1^-$ and $d_2^+ = d_2^-$ which by using $d_1^+ = d_1^- = d_1$ and $d_2^+ = d_2^- = d_2$ in (3.4a-3.4b) reduces the number of unknown coefficients to 16.

Conditions (3.5a) - (3.5d) lead to a linear system $\mathbf{M}\mathbf{c}_i = \mathbf{b}_i$ about the coefficient $\mathbf{c}_i = (a_1^+, b_1^+, c_1^+, d_1, a_1^-, b_1^-, c_1^-, a_2^+, b_2^+, c_2^+, d_2, a_2^-, b_2^-, c_2^-, a_3^-, a_3^+)^T$. The matrix $\mathbf{M}$ for Type I element is

$$\begin{pmatrix}
1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & \hat{d} & 0 & -1 & 0 & -\hat{d} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & d & 0 & -1 & 0 & -\hat{d} & 0 & 0 \\
0 & m_{52} & m_{53} & m_{54} & 0 & m_{56} & \nu^-\hat{e} & 0 & -\nu^+\hat{e} & 0 & m_{511} & 0 & \nu^-\hat{e} & 0 & -\hat{d} & \hat{d} \\
0 & 0 & m_{63} & m_{64} & 0 & 0 & \nu^-\hat{d} & 0 & -\nu^+\hat{d} & m_{610} & m_{611} & 0 & \nu^-\hat{d} & m_{614} & -\hat{e} & \hat{e} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{1515} & m_{1516} \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}, \tag{3.6}$$

where

$$m_{52} = -2\nu^+\hat{d}, \quad m_{53} = -\nu^+\hat{e}, \quad m_{54} = \frac{1}{2}(\nu^- - \nu^+)\left(\hat{e}^2 + 2\hat{d}^2\right), m_{56} = 2\nu^-\hat{d},$$

$$m_{511} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d},$$

$$m_{63} = -\nu^+\hat{d}, \quad m_{64} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d}, \quad m_{610} = -2\nu^+\hat{e},$$

$$m_{611} = \frac{1}{2}(\nu^- - \nu^+)\left(2\hat{e}^2 + \hat{d}^2\right), m_{614} = 2\nu^-\hat{e},$$

$$m_{1515} = \frac{1}{2}\hat{e}\hat{d}, \quad m_{1516} = 1 - \frac{1}{2}\hat{e}\hat{d}.$$

For an element of Type II the matrix $\mathbf{M}$ is

$$\begin{pmatrix}
1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & d & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{d} & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 \\
0 & -2\nu^+ & m_{53} & m_{54} & 0 & 2\nu^- & m_{57} & 0 & m_{59} & 0 & m_{510} & 0 & m_{513} & 0 & -1 & 1 \\
0 & 0 & -\nu^+ & m_{64} & 0 & 0 & \nu^- & 0 & -\nu^+ & m_{610} & m_{611} & 0 & \nu^- & m_{614} & \hat{d}-\hat{e} & m_{616} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\hat{d}+\hat{e}}{2} & m_{1516} \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}, \quad (3.7)$$

$$m_{53} = \nu^+(\hat{d}-\hat{e}),\ m_{54} = -\frac{1}{2}(\nu^- - \nu^+)\left(-2+\hat{d}^2-\hat{e}^2\right),\ m_{57} = \nu^-(-\hat{d}+\hat{e}),\ \ m_{59} = \nu^+(\hat{d}-\hat{e}),$$

$$m_{511} = -\frac{1}{2}(\nu^- - \nu^+)(\hat{d}-\hat{e}),\ m_{513} = \nu^-(-\hat{d}+\hat{e}),$$

$$m_{64} = \frac{1}{2}(\nu^- - \nu^+)(\hat{d}+\hat{e}), m_{610} = 2\nu^+(\hat{d}-\hat{e}),\ m_{611} = -\frac{1}{2}(\nu^- - \nu^+)\left(-1+2\hat{d}^2-2\hat{e}^2\right),$$

$$m_{616} = -\hat{d}+\hat{e},\ m_{614} = 2\nu^-(-\hat{d}+\hat{e}), m_{1516} = \frac{1}{2}(2-\hat{d}-\hat{e}).$$

The vector $\mathbf{b}_i$ for both Type I and Type II is given by

$$\mathbf{b}_i = (0,0,0,0,0,0,\delta_{i,1},\delta_{i,2},\delta_{i,3},\delta_{i,4},\delta_{i,5},\delta_{i,6},\delta_{i,7},\delta_{i,8},\delta_{i,9},0)^T.$$

Once these IFE shape functions are constructed on the reference element, the standard affine mapping is applied to obtain the corresponding vector IFE shape functions on an interface element T as $\mathbf{\Phi}_i(x,y) = \hat{\mathbf{\Phi}}_i(F^{-1}(\hat{x},\hat{y})),\ \ i = 1,2,\ldots,9.$

On every non-interface element T we use the standard finite element shape functions $\mathbf{\Psi}_i,\ i = 1,2,\ldots,9$:

$$\mathbf{\Psi}_j = \begin{pmatrix} \psi_j(x,y) \\ 0 \\ 0 \end{pmatrix}, \mathbf{\Psi}_{j+4} = \begin{pmatrix} 0 \\ \psi_j(x,y) \\ 0 \end{pmatrix}, \mathbf{\Psi}_9 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad j = 1,2,3,4, \qquad (3.8)$$

where $\psi_j,\ j = 1,2,3,4$ are the standard Lagrange bilinear shape functions associated with the vertices of T. Figure 3.2 presents illustrations for the shape functions $\mathbf{\Psi}_1$ and $\mathbf{\Phi}_1$. Unlike $\mathbf{\Psi}_1$ used in the standard Q_1/Q_0 finite element space, the components of the IFE shape function $\mathbf{\Phi}_1$ cannot be decoupled, *i.e.,* its second and third components are not always zero.

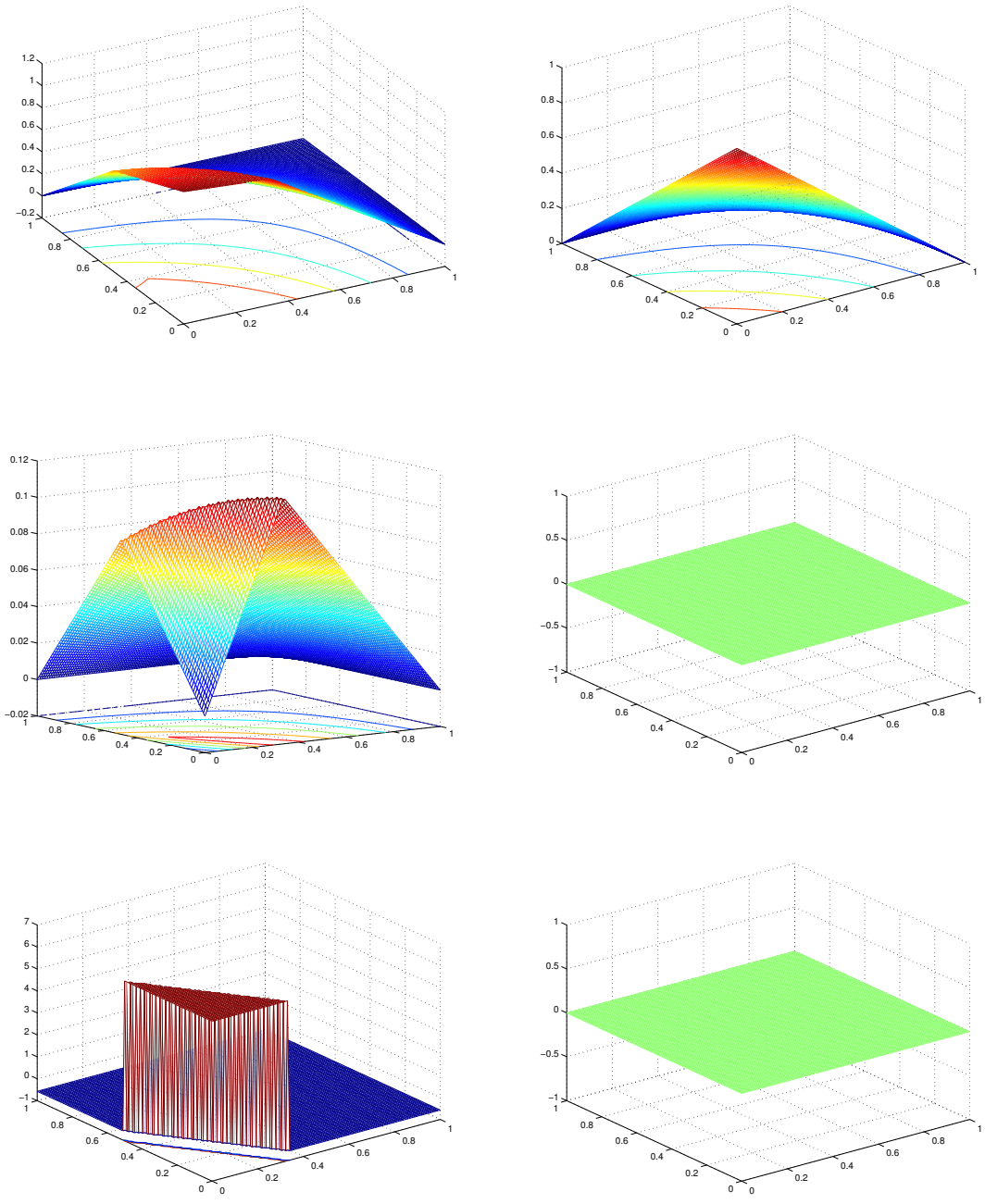


Figure 3.2: The $(u_1, u_2, p)^T$ components of the IFE shape function $\mathbf{\Phi}_1$ (left) and standard shape function $\mathbf{\Psi}_1$ (right).

Then, the shape functions defined above are used to construct the global IFE space on Ω for the Stokes interface problem as follows:

$$S_h(\Omega) = \{\mathbf{U}_h \mid \mathbf{U}_h|_T \in X_h(T)\}, \tag{3.9}$$

where

$$X_h(T) = \begin{cases} \text{span}\{\mathbf{\Phi}_i(x,y), i = 1,2,\ldots,9\}, & \text{if } T \in \mathcal{T}_h^i, \\ \text{span}\{\mathbf{\Psi}_i(x,y), i = 1,2,\ldots,9\}, & \text{if } T \in \mathcal{T}_h^n. \end{cases}$$

Remark. *The constraint (3.5c) is necessary to obtain a square invertible system. Other variations of the constraint such as* $\int_{\overline{DE}} \nabla \cdot \mathbf{u} ds = 0$ *and* $\nabla \cdot \mathbf{u}(D) = 0$ *were investigated yielding (i) a non-invertible system for some mesh-interface configurations and (ii) non-consistent IFE shape functions with the standard Lagrange shape functions.*

3.2 Space $\tilde{S}_h(\Omega)$

In this section, we develop another IFE space using an alternative approach which does not use the divergence condition in (3.5c). In fact, developing higher order approximations will require finding more constraints that might not be easy to do. In addition, the existence of the IFE basis functions becomes trivial.

As in the first approach, we write the piecewise basis associated with the velocity and the pressure on the reference element in the form (3.4a−3.4c). We can see that each IFE basis has 18 coefficients $a_j^s, b_j^s, c_j^s, d_j^s, a_3^s$, $j = 1, 2$, $s = +, -$. On each interface element, these coefficients have to be determined according to jump conditions. We suggest to use the following restrictions to construct the IFE shape functions $\hat{\mathbf{\Phi}}_i$, $i = 1, 2, ..., 10$:

• continuity of the velocity across $\overline{DE}$:

$$\hat{\mathbf{\Theta}}^-(\hat{E}) = \hat{\mathbf{\Theta}}^+(\hat{E}), \quad \hat{\mathbf{\Theta}}^-(\hat{D}) = \hat{\mathbf{\Theta}}^+(\hat{D}), \quad \frac{\partial^2 \hat{\mathbf{\Theta}}^-}{\partial \hat{x} \partial \hat{y}} = \frac{\partial^2 \hat{\mathbf{\Theta}}^+}{\partial \hat{x} \partial \hat{y}} \tag{3.10}$$

• weak continuity of the normal component of the stress tensor across $\overline{DE}$:

$$\int_{\overline{\hat{D}\hat{E}}} [\mathbf{S}(\hat{\mathbf{\Theta}}, \hat{\phi}_3)\mathbf{n}_{\overline{\hat{D}\hat{E}}}] ds = 0. \tag{3.11}$$

Again, the restrictions on the second derivatives of the velocity basis functions in (3.10) is equivalent to $d_1^+ = d_1^-$ and $d_2^+ = d_2^-$, which by using $d_1^+ = d_1^- = d_1$ and $d_2^+ = d_2^- = d_2$ in (3.4a−3.4c), reduces the number of unknowns to 16 coefficients.

Conditions (3.10) and (3.11) yield the algebraic linear system

$$\mathbf{Mc} = \mathbf{b}, \tag{3.12}$$

$$\mathbf{c} = (a_1^+, b_1^+, c_1^+, d_1, a_1^-, b_1^-, c_1^-, a_2^+, b_2^+, c_2^+, d_2, a_2^-, b_2^-, c_2^-, a_3^-, a_3^+)^T, \tag{3.13}$$

$$\mathbf{b} = (0,0,0,0,0,0)^T. \tag{3.14}$$

For Type I elements, we have

$$\mathbf{M} = \begin{pmatrix} 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & \hat{d} & 0 & -1 & 0 & -\hat{d} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \hat{d} & 0 & -1 & 0 & -\hat{d} & 0 & 0 \\ 0 & m_{52} & m_{53} & m_{54} & 0 & m_{56} & \nu^-\hat{e} & 0 & -\nu^+\hat{e} & 0 & m_{511} & 0 & \nu^-\hat{e} & 0 & -\hat{d} & \hat{d} \\ 0 & 0 & m_{63} & m_{64} & 0 & 0 & \nu^-\hat{d} & 0 & -\nu^+\hat{d} & m_{610} & m_{611} & 0 & \nu^-\hat{d} & m_{614} & -\hat{e} & \hat{e} \end{pmatrix},$$

where

$$m_{52} = -2\nu^+\hat{d}, \quad m_{53} = -\nu^+\hat{e}, \quad m_{54} = \frac{1}{2}(\nu^- - \nu^+)\left(\hat{e}^2 + 2\hat{d}^2\right), m_{56} = 2\nu^-\hat{d},$$

$$m_{511} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d},$$

$$m_{63} = -\nu^+\hat{d}, \quad m_{64} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d}, \quad m_{610} = -2\nu^+\hat{e},$$

$$m_{611} = \frac{1}{2}(\nu^- - \nu^+)\left(2\hat{e}^2 + \hat{d}^2\right), m_{614} = 2\nu^-\hat{e}.$$

For Type II elements, we obtain

$$\mathbf{M} = \begin{pmatrix} 1 & \hat{e} & 0 & 0 & -1 & m_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \hat{d} & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{d} & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 \\ 0 & -2\nu^+ & m_{53} & m_{54} & 0 & 2\nu^- & m_{57} & 0 & m_{59} & 0 & m_{511} & 0 & m_{513} & 0 & -1 & 1 \\ 0 & 0 & m_{63} & m_{64} & 0 & 0 & \nu^- & 0 & m_{69} & m_{610} & m_{611} & 0 & \nu^- & m_{614} & \hat{d}-\hat{e} & m_{616} \end{pmatrix}, \tag{3.15}$$

where

$$m_{16} = -\hat{e},\ m_{53} = \nu^+(\hat{d} - \hat{e}),\ m_{54} = -\frac{1}{2}(\nu^- - \nu^+)\left(-2 + \hat{d}^2 - \hat{e}^2\right),\ m_{57} = \nu^-(-\hat{d} + \hat{e}),$$

$$m_{59} = \nu^+(\hat{d} - \hat{e}), m_{511} = -\frac{1}{2}(\nu^- - \nu^+)(\hat{d} - \hat{e}),\ m_{513} = \nu^-(-\hat{d} + \hat{e}),$$

$$m_{64} = \frac{1}{2}(\nu^- - \nu^+)(\hat{d} + \hat{e}), m_{610} = 2\nu^+(\hat{d} - \hat{e}),\ m_{611} = -\frac{1}{2}(\nu^- - \nu^+)\left(-1 + 2\hat{d}^2 - 2\hat{e}^2\right),$$

$$m_{616} = -\hat{d} + \hat{e},\ m_{614} = 2\nu^-(-\hat{d} + \hat{e}), m_{1516} = \frac{1}{2}(2 - \hat{d} - \hat{e}).$$

For both Type I and II elements the matrices $\mathbf{M}$ are full-ranked since the constraints in (3.10) and (3.11) are linearly independent, which will be shown in Proposition 3.4.1. Next, we find a basis $\{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, ..., \boldsymbol{\zeta}_{10}\}$ for the null space of $\mathbf{M}$, where

$$\boldsymbol{\zeta}_i = \left(a^+_{1,i}, b^+_{1,i}, c^+_{1,i}, d_{1,i}, a^-_{1,i}, b^-_{1,i}, c^-_{1,i}, a^+_{2,i}, b^+_{2,i}, c^+_{2,i}, d_{2,i}, a^-_{2,i}, b^-_{2,i}, c^-_{2,i}, a^-_{3,i}, a^+_{3,i}\right)^T.$$

The entries of $\boldsymbol{\zeta}_i$ can therefore be used to construct the IFE basis $\hat{\boldsymbol{\Phi}}_i$, $i = 1, 2, ..., 10$, using (3.4a-3.4c). Then, using the affine mapping between the reference element $\hat{T}$ and physical reference T, we obtain the i^{th} local vector IFE basis function on T as follows:

$$\boldsymbol{\Phi}_i(x, y) = \hat{\boldsymbol{\Phi}}_i(F^{-1}(\hat{x}, \hat{y})), \quad i = 1, 2, ..., 10.$$

These basis functions are used to construct the global IFE space $\tilde{S}_h(\Omega)$ as

$$\tilde{S}_h(\Omega) = \{\mathbf{U}_h \mid \mathbf{U}_h|_T \in \tilde{X}_h(T)\}, \tag{3.16}$$

where

$$\tilde{X}_h(T) = \begin{cases} \text{span}\{\boldsymbol{\Phi}_i(x, y), i = 1, 2, \ldots, 10\}, & \text{if } T \in \mathcal{T}_h^i, \\ \text{span}\{\boldsymbol{\Psi}_i(x, y), i = 1, 2, \ldots, 9\}, & \text{if } T \in \mathcal{T}_h^n, \end{cases}$$

and $\boldsymbol{\Psi}_i$, $i = 1, 2, ..., 9$ are the standard Lagrange finite element basis defined in (3.8).

3.3 Q_1/Q_0 particular IFE functions with surface force

In the case where the jump condition (1.3f) is such that $\boldsymbol{\sigma} \neq \mathbf{0}$, we follow [67] to treat the non-homogeneous term. First, we construct a set of particular IFE functions that can capture the non-homogeneous term modeling the surface force. These particular functions are solely used to construct the IFE solution and will not be included in the test space; hence, they will not increase the size of the discrete IFE problem.

On each interface element T, the particular IFE functions are defined as

$$\Upsilon_j = \begin{pmatrix} \boldsymbol{\Lambda}_j \\ \psi_j \end{pmatrix}, \ j = 1, 2.$$

whose velocity component is

$$\boldsymbol{\Lambda}_j(x, y) = \begin{cases} \boldsymbol{\Lambda}_j^+(x, y) & \text{on } T^+ \\ \boldsymbol{\Lambda}_j^-(x, y) & \text{on } T^- \end{cases}, \quad j = 1, 2, \tag{3.17}$$

with

$$\boldsymbol{\Lambda}_j^s(x, y) = \begin{pmatrix} \Lambda^s_{1,j}(x, y) \\ \Lambda^s_{2,j}(x, y) \end{pmatrix}, \quad \text{on } T^s, \quad s = +, -, \quad j = 1, 2, \tag{3.18}$$

and the pressure component is a piecewise constant function such that

$$\psi_j = \begin{cases} \psi_j^+(x,y), & \text{on } T^+ \\ \psi_j^-(x,y), & \text{on } T^- \end{cases}, \quad \psi_j^s(x,y) = a_{3,j}^s, \quad s = +,-.$$

We then select the velocity component in the particular IFE functions to be piecewise polynomials as

$$\Lambda_{i,j}^s(x,y) = a_{i,j}^s + b_{i,j}^s x + c_{i,j}^s y + d_{i,j}^s xy, \quad j = 1,2, \quad i = 1,2, \quad s = +,-, \tag{3.19}$$

and require the particular IFE functions to satisfy the following constraints

$$\mathbf{\Lambda}_j(A_i) = 0, \quad i = 1,2,3,4, \tag{3.20}$$

$$\mathbf{\Lambda}_j^-(E) = \mathbf{\Lambda}_j^+(E), \quad \mathbf{\Lambda}_j^-(D) = \mathbf{\Lambda}_j^+(D), \quad \frac{\partial^2 \mathbf{\Lambda}_j^-}{\partial x \partial y} = \frac{\partial^2 \mathbf{\Lambda}_j^+}{\partial x \partial y}, \tag{3.21}$$

$$\int_{\overline{DE}} [\mathbf{S}(\mathbf{\Lambda}_j, \psi_j)\mathbf{n}_{\overline{DE}}] ds = \mathbf{e}_j, \ \mathbf{e}_j \text{ is the canonical vector in } \mathbb{R}^2 \tag{3.22}$$

$$\frac{1}{|T|}\int_T \psi_j dX = 0, \nabla \cdot \mathbf{\Lambda}_j^+(D) = \nabla \cdot \mathbf{\Lambda}_j^-(D). \tag{3.23}$$

These conditions lead to a linear system for determining the parameters of the particular IFE functions; and, after mapping (3.20-3.23) to the reference element, the matrix of this linear system is equal to the one used to find the IFE shape functions in section 3.1. However the right-hand side is different and is given as:

$$\mathbf{b}_j = (0,0,0,0,\delta_{j,1},\delta_{j,2},0,0,0,0,0,0,0,0,0,0)^T.$$

For the Stokes interface problem with a nonzero surface force $\boldsymbol{\sigma} \neq \mathbf{0}$, we define its IFE solution $(\mathbf{u}_h, p_h)^T$ to be in the function set $S_h(\Omega) \oplus \{\mathbf{q}_h(x,y)\}$ where $S_h(\Omega)$ is the IFE space defined in the section 3.1 and the vector function $\mathbf{q}_h(x,y)$ is such that

$$\mathbf{q}_h(x,y) = \begin{cases} \kappa_1 \Upsilon_1(x,y) + \kappa_2 \Upsilon_2(x,y), & \text{on T } \in \mathcal{T}_i \\ 0 & \text{elsewhere} \end{cases}, \tag{3.24}$$

where $\kappa_j = \int_{\overline{DE}} \boldsymbol{\sigma}_{h,j} ds$ and $\boldsymbol{\sigma}_h = (\boldsymbol{\sigma}_{h,1}, \boldsymbol{\sigma}_{h,2})^T$ is the linear interpolation of $\boldsymbol{\sigma}$ defined by $\boldsymbol{\sigma}_h(E) = \boldsymbol{\sigma}(E)$ and $\boldsymbol{\sigma}_h(D) = \boldsymbol{\sigma}(D)$. In fact, the IFE solution on an interface element will take the form

$$\begin{pmatrix} \mathbf{u}_h \\ p_h \end{pmatrix} = \sum_{i=1}^{9} c_i \mathbf{\Phi}_i + \kappa_1 \Upsilon_1 + \kappa_2 \Upsilon_2. \tag{3.25}$$

Remark. *The set of particular IFE functions can be used to approximate the IFE solutions if added to the spaces $S_h(\Omega)$ or $\tilde{S}_h(\Omega)$. If necessary, the divergence constraint in (3.23) can be ignored and the particular IFE functions can be constructed using null space-based approach. This work however will not cover this case.*

3.4 Basic properties of the Q_1/Q_0 IFE space $S_h(\Omega)$

In this section, we investigate basic properties of the IFE basis functions for the Stokes interface problem constructed using the first approach (*i.e.* the space $S_h(\Omega)$). We start by showing the existence and uniqueness of IFE shape functions for an arbitrary configuration of ν^-, ν^+ and interface location.

Proposition 3.4.1. *On every interface element T, there exists a unique set of linearly independent IFE shape functions $\{\mathbf{\Phi}_i,\ i = 1, 2, \ldots, 9\}$ as defined in section 3.1.*

Proof. The uniqueness and existence of IFE shape functions are directly guaranteed by the invertibility of the matrices (3.6) and (3.7).

For interface elements of Type I a direct computation of the determinant of the matrix $\mathbf{M}$ in (3.6) yields

$$det(\mathbf{M}) = -\frac{1}{2}(\hat{e}^2 + \hat{d}^2)(Q_1\nu^- + Q_2\nu^+), \tag{3.26}$$

$$\begin{aligned} Q_1 &= (-2\hat{e}^2(-1+\hat{d}) + \hat{e}^3\hat{d} + 2\hat{d}^2 + \hat{e}(-2+\hat{d})\hat{d}^2), \\ Q_2 &= -\hat{e}\hat{d}(-2\hat{e} + \hat{e}^2 + (-2+\hat{d})\hat{d}). \end{aligned}$$

Thus it suffices to show that $Q_1 > 0$ and $Q_2 > 0$. Since $0 < \hat{e}, \hat{d} < 1$ and $\hat{e}^2 \leq \hat{e}$

$$Q_1 > -4\hat{e}\hat{d} + 2\hat{e}^2 + 2\hat{d}^2 + \hat{e}^3\hat{d} + \hat{e}\hat{d}^3.$$

Factoring the first three terms of the right-hand side we obtain

$$Q_1 > 2(\hat{e} - \hat{d})^2 + \hat{e}^3\hat{d} + \hat{e}\hat{d}^3 > 0.$$

Similarly one can easily check that $Q_2 > 0$ by writing Q_2 as

$$Q_2 = -\hat{e}\hat{d}(\hat{e}(-2+\hat{e}) + (-2+\hat{d})\hat{d})$$

and noting that both $(-2+\hat{d})\hat{d} < 0$ and $(-2+\hat{e})\hat{e} < 0$. This leads to $Q_2 > 0$. Hence, $det(\mathbf{M}) \neq 0$, for all $\hat{e}, \hat{d} \in (0, 1)$.

For interface elements of Type II the determinant of the matrix $\mathbf{M}$ in (3.7) is such that

$$det(\mathbf{M}) = -\frac{1}{2}(1 + \hat{d}^2 - 2\hat{d}\hat{e} + \hat{e}^2)(P_1\nu^- + P_2\nu^+), \tag{3.27}$$

where

$$\begin{aligned} P_1 &= (2 + \hat{d}^3 - \hat{e} - \hat{d}^2\hat{e} + \hat{e}^3 - \hat{d}(1+\hat{e}^2)), \\ P_2 &= (-\hat{d}^3 + \hat{e} + 2\hat{e}^2 - \hat{e}^3 + \hat{d}^2(2+\hat{e}) + \hat{d}(1 - 4\hat{e} + \hat{e}^2)). \end{aligned}$$

First, we can write P_1 as

$$P_1 = 2 + \hat{d}^3 + \hat{e}^3 - \hat{e} - \hat{d} - \hat{e}^2\hat{d} - \hat{e}\hat{d}^2,$$

and noting that $-\hat{e}^2\hat{d} > -\hat{e}\hat{d}$ and$-\hat{e}\hat{d}^2 > -\hat{e}\hat{d}$, we obtain

$$P_1 > 2 + \hat{d}^3 + \hat{e}^3 - \hat{e} - \hat{d} - 2\hat{e}\hat{d}.$$

Applying $2\hat{e}\hat{d} < \hat{e}^2 + \hat{d}^2$ we obtain

$$P_1 > 2 + \hat{d}(\hat{d}^2 - \hat{d} - 1) + \hat{e}(\hat{e}^2 - \hat{e} - 1).$$

Since $-1 < x(x^2 - x - 1) < 0$, for $0 < x < 1$ we establish $P_1 > 0$. Next, we write P_2 as follows

$$P_2 = -\hat{d}^3 + \hat{e} + 2\hat{e}^2 - \hat{e}^3 + 2\hat{d}^2 + \hat{d}^2\hat{e} + \hat{d} - 4\hat{d}\hat{e} + \hat{d}\hat{e}^2$$

Note that $\hat{e} > \hat{e}^3$ and $\hat{d} > \hat{d}^3$, thus

$$P_2 > 2\hat{e}^2 + 2\hat{d}^2 + \hat{e}\hat{d}^2 + \hat{d}\hat{e}^2 - 4\hat{d}\hat{e} = 2(\hat{e} - \hat{d})^2 + \hat{e}\hat{d}^2 + \hat{d}\hat{e}^2 > 0.$$

Therefore $det(\mathbf{M}) \neq 0, \quad \forall \quad 0 < \hat{e}, \hat{d} < 1.$ □

The next proposition shows that the velocity component of an IFE function is continuous on each element.

Proposition 3.4.2. *The velocity component of an IFE function, defined in section 3.1, is continuous on each element.*

Proof. The statement is obviously true on each non-interface element because the standard bilinear shape functions are all continuous. On each interface element, the arguments in Proposition 4.1 of [133] can be applied here to show the continuity of the velocity component of each IFE shape function is continuous and this results in the needed proof. □

The next result is easy to verify.

Proposition 3.4.3. *The IFE Q_1/Q_0 space $S_h(\Omega)$, defined in section 3.1, over a mesh $\mathcal{T}_h$ of Ω has the same dimension as the standard Q_1/Q_0 finite element space on the same mesh.*

Next we show that the IFE shape functions on each interface element form a partition of unity.

Proposition 3.4.4. *If $\mathbf{\Phi}_i,\ i = 1, 2, \ldots, 9$ are the IFE shape functions constructed in section 3.1 on an interface element, then*

$$\sum_{i=1}^{4} \mathbf{\Phi}_i(x, y) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \sum_{i=4}^{8} \mathbf{\Phi}_i(x, y) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tag{3.28}$$

$$\mathbf{\Phi}_9(x, y) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \forall (x, y) \in T. \tag{3.29}$$

Proof. The proof is accomplished by a direct verification. □

The next proposition shows that the proposed Q_1/Q_0 IFE shape functions are consistent with the associated standard finite element shape functions.

Proposition 3.4.5. *If $\nu^+ = \nu^-$ on an interface element T and the Q_1/Q_0 IFE finite element shape functions are constructed in section 3.1, then the Q_1/Q_0 IFE and standard Lagrange Q_1/Q_0 finite element shape functions are the same.*

Proof. Noting that, when $\nu^+ = \nu^-$, the jump conditions (3.5a-3.5d) are satisfied by both the IFE and the standard Q_1/Q_0 shape functions, by uniqueness, these two types of shape functions are equal. □

Next, we establish modified trace and inverse inequalities for the IFE basis. The argument used to establish the standard trace and inverse inequalities cannot be used due to non-smoothness of the IFE basis functions. Instead, we use an argument based on Taylor expansion. These two inequalities will be necessary to investigate the global error of the IFE methods.

We start by stating and proving a lemma that relates the gradient of the IFE basis functions on T^- to their counterpart on T^+.

Lemma 3.4.6. *For any $\mathbf{v} \in S_h(\Omega)$ constructed in section 3.1 and defined on an interface element as*

$$\mathbf{v}(x,y) = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix} = \begin{pmatrix} v_1^s(x,y) \\ v_2^s(x,y) \\ v_3^s(x,y) \end{pmatrix}, \quad s = +,-, \tag{3.30a}$$

where

$$v_1^s(x,y) = a_1^s + b_1^s x + c_1^s y + d_1 xy, \tag{3.30b}$$
$$v_2^s(x,y) = a_2^s + b_2^s x + c_2^s y + d_2 xy, \tag{3.30c}$$
$$v_3^s(x,y) = a_3^s, \tag{3.30d}$$

the following equality holds

$$\begin{pmatrix} \nabla v_1^+(D) \\ \nabla v_2^+(D) \\ v_3^+ \end{pmatrix} = A \begin{pmatrix} \nabla v_1^-(D) \\ \nabla v_2^-(D) \\ v_3^- \end{pmatrix}, \tag{3.31}$$

where

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & 0 \\ B_1 & B_2 & B_3 \end{pmatrix},$$

$$A_{11} = \begin{pmatrix} \frac{-(2(\nu^-)n_x^2n_y^2+(\nu^+)(n_x^4+n_y^4))}{\nu^+} & \frac{((\nu^-)-(\nu^+))n_xn_y(n_x^2-n_y^2))}{\nu^+} \\ \frac{2((\nu^-)-(\nu^+))n_x^3n_y)}{\nu^+} & \frac{-((\nu^-)n_x^4-(\nu^-)n_x^2n_y^2+3(\nu^+)n_x^2n_y^2+(\nu^+)n_y^4)}{\nu^+} \end{pmatrix},$$

$$A_{12} = \begin{pmatrix} \frac{((\nu^-)-(\nu^+))n_xn_y(n_x^2-n_y^2))}{\nu^+} & \frac{2((\nu^-)-(\nu^+))n_x^2n_y^2)}{(\nu^+)} \\ \frac{-(((\nu^-)-(\nu^+))n_x^2(n_x^2-n_y^2))}{\nu^+} & \frac{2(-(\nu^-)+(\nu^+))n_x^3n_y)}{\nu^+} \end{pmatrix},$$

$$A_{21} = \begin{pmatrix} \frac{2(-(\nu^-)+(\nu^+))n_xn_y^3)}{\nu^+} & \frac{((\nu^-)-(\nu^+))n_y^2(n_x^2-n_y^2))}{\nu^+} \\ \frac{2((\nu^-)-(\nu^+))n_x^2n_y^2)}{\nu^+} & \frac{-(((\nu^-)-(\nu^+))n_xn_y(n_x^2-n_y^2))}{\nu^+} \end{pmatrix},$$

$$A_{22} = \begin{pmatrix} \frac{-(((\nu^+)n_x^4-(\nu^-)n_x^2n_y^2+3(\nu^+)n_x^2n_y^2+(\nu^-)n_y^4)}{\nu^+} & \frac{(2((\nu^-)-(\nu^+))n_xn_y^3)}{\nu^+} \\ \frac{-((\nu^-)-(\nu^+))n_xn_y(n_x^2-n_y^2))}{\nu^+} & \frac{-(2(\nu^-)n_x^2n_y^2+(\nu^+)(n_x^4+n_y^4))}{\nu^+} \end{pmatrix},$$

$$B_1 = \left(2(\nu^- - \nu^+)n_x^2,\ 2(\nu^- - \nu^+)n_xn_y\right), \quad B_2 = \left(2(\nu^- - \nu^+)n_xn_y,\ (2(\nu^- - \nu^+)n_y^2\right),$$

$$B_3 = -n_x^2 - n_y^2.$$

Proof. The function $\mathbf{v} = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix}$ defined in (3.30) is in $S_h(\Omega)$, hence it satisfies the jump conditions (3.5a), (3.5b) and (3.5c). Therefore

$$[\mathbf{u}]_{\overline{DE}} = \mathbf{0}, \tag{3.32}$$

$$[\mathbf{S}(\mathbf{u}, v_3)\mathbf{n}]_{\overline{DE}} = \mathbf{0}, \tag{3.33}$$

$$[\nabla \cdot \mathbf{u}(D)] = 0. \tag{3.34}$$

Equation (3.32) implies that the jump in the tangential derivative of the velocity $\mathbf{u}$ across the interface is zero. The latter combined with (3.33) and (3.34) yields (3.31). □

Remark. *The entries of the matrices A_{ij}, $1 \leq i, j \leq 2$, defined in Lemma 3.4.6 can be bounded with a constant C independent of the mesh size h and the interface location.*

Next, we state a lemma that relates the coefficients of the basis functions defined on T^- to their counterparts defined on T^+.

Lemma 3.4.7. *For any* $\mathbf{v} = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix} \in S_h(\Omega)$ *defined in (3.30), there exists* $C > 0$ *independent of the mesh size* h *and the interface location such that*

$$|b_i^+| + |c_i^+| + h|d_i| \leq C \sum_{j=1}^{2} \left(|b_j^-| + |c_j^-| + h|d_j| \right), \quad i = 1, 2. \tag{3.35}$$

$$|b_i^-| + |c_i^-| + h|d_i| \leq C \sum_{j=1}^{2} \left(|b_j^+| + |c_j^+| + h|d_j| \right), \quad i = 1, 2. \tag{3.36}$$

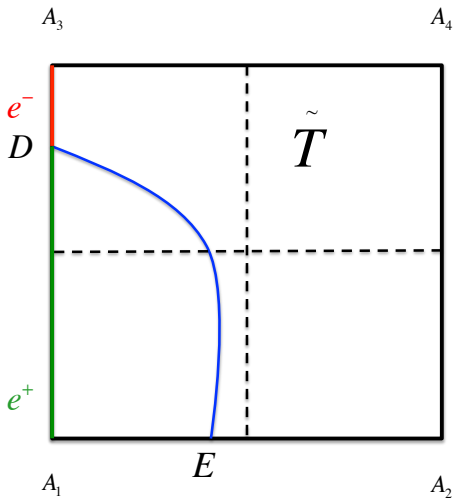


Figure 3.3: An interface element.

Proof. As shown in Figure 3.3, we let $T = \square A_1A_2A_3A_4$ be an interface element with vertices $A_1 = (0,0)$, $A_2 = (h,0)$, $A_3 = (0,h)$ and $A_4 = (h,h)$ and, without loss of generality denote the interface edge of T that contains $D = (x_D, y_D) = (0, \gamma h)$, $0 < \gamma < 1$ as e and denote by e^+ and e^- the segments of the edge e that lie in Ω^+ and Ω^- respectively. Next, let $X = (x,y)^T \in e^+$, then Taylor expansion of $v_i^+(X)$, $i = 1, 2$ around D is:

$$v_i^+(X) = v_i^+(D) + \nabla v_i^+(D) \cdot (X - D) + d_i \left((x - x_D)(y - y_D) \right), \quad i = 1, 2. \tag{3.37}$$

Using the continuity condition (3.5a) and (3.31),

$$\begin{aligned} v_i^+(X) = v_i^-(D) + A_{i1} \nabla v_1^-(D) \cdot (X - D) + A_{i2} \nabla v_2^-(D) \cdot (X - D) \\ + d_i (x - x_D)(y - y_D), \quad i = 1, 2. \end{aligned} \tag{3.38}$$

Let $(d_{ij})_{1 \leq i,j \leq 2}$ be the entries of A_{11} and let $(a_{ij})_{1 \leq i,j \leq 2}$ be the entries of A_{12}. Taking the derivative of $v_1^+(X)$ in (3.38) with respect to x yields

$$v_{1,x}^+(X) = d_{11} v_{1,x}^-(D) + d_{12} v_{1,y}^-(D) + a_{11} v_{2,x}^-(D) + a_{12} v_{2,y}^-(D) + d_1 (y - y_D).$$

Using (3.30) leads to

$$b_1^+ + d_1 y = d_{11}(b_1^- + d_1 y_D) + d_{12}(c_1^- + d_1 x_D) + a_{11}(b_2^- + d_2 y_D)$$
$$+a_{12}(c_2^- + d_2 x_D) + d_1(y - y_D).$$

We note that the term $d_1 y$ cancels out and use the fact that the entries of the matrices A_{11} and A_{12} are bounded by a constant independent of h and the interface location to bound b_1^+ as follows

$$|b_1^+| \leq C\left(|b_1^-| + |c_1^-| + h|d_1| + |b_2^-| + |c_2^-| + h|d_2|\right). \tag{3.39}$$

Similarly by taking the derivative of (3.38) with respect to y and using (3.30), we obtain

$$|c_1^+| \leq C\left(|b_1^-| + |c_1^-| + h|d_1| + |b_2^-| + |c_2^-| + h|d_2|\right). \tag{3.40}$$

Equations (3.39) and (3.40) yield

$$|b_1^+| + |c_1^+| + h|d_1| \leq C\left(|b_1^-| + |c_1^-| + h|d_1| + |b_2^-| + |c_2^-| + h|d_2|\right). \tag{3.41}$$

In a similar way and using $i = 2$ in Equation (3.38), we prove that

$$|b_2^+| + |c_2^+| + h|d_2| \leq C\left(|b_1^-| + |c_1^-| + h|d_1| + |b_2^-| + |c_2^-| + h|d_2|\right).$$

Taking $X \in e^-$ and expanding $v_i^-(x)$ around D as in (3.37) and following the same reasoning as to obtain (3.35) yield

$$|b_i^-| + |c_i^-| + h|d_i| \leq C_2\left(|b_1^+| + |c_1^+| + h|d_1| + |b_2^+| + |c_2^+| + h|d_2|\right), \quad i = 1, 2.$$

□

Divide the element T into 4 congruent rectangles as shown in Figure 3.3. Without loss of generality, let us assume that the upper right rectangle $\tilde{T}$ lies completely in T^-. In the next Lemma, we establish bounds for the coefficients in (3.30b) and (3.30c) in terms of $\mathbf{v}$ defined in (3.30).

Lemma 3.4.8. *For any* $\mathbf{v} = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix} \in S_h(\Omega)$ *defined in (3.30), we have*

$$\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2 \geq Ch\left((b_1^s)^2 + (c_1^s)^2 + h^2(d_1)^2 + (b_2^s)^2 + (c_2^s)^2 + h^2(d_2)^2\right), \quad s = +, -. \tag{3.42}$$

Proof. By direct computations, (3.30) yields

$$||\partial_x v_1||_{0,\tilde{T}}^2 = \frac{h^2}{48}\left(12(b_1^-)^2 + 18 b_1^- d_1 h + 7(hd_1)^2\right)$$

$$\geq \frac{h^2}{48}\left(\left(12 - \frac{9}{\sigma_1}\right)(b_1^-)^2 + (7 - 9\sigma_1)(hd_1)^2\right), \text{ with } \sigma_1 > 0. \text{ (Young's inequality)}$$

Similarly,

$$||\partial_y v_1||^2_{0,\tilde{T}} = \frac{h^2}{48}\left(12(c_1^-)^2 + 18c_1^- d_1 h + 7(hd_1)^2\right)$$
$$\geq \frac{h^2}{48}\left(\left(12 - \frac{9}{\sigma_1}\right)(c_1^-)^2 + (7 - 9\sigma_1)(hd_1)^2\right).$$

Let $\sigma_{1,2} \in (\frac{3}{4}, \frac{7}{9})$, then

$$||\nabla v_1||^2_{0,\tilde{T}} \geq Ch^2\left((b_1^-)^2 + (c_1^-)^2 + (hd_1)^2\right).$$

Following the previous reasoning, a similar lower bound for v_2 can be written as

$$||\nabla v_2||^2_{0,\tilde{T}} \geq Ch^2\left((b_2^-)^2 + (c_2^-)^2 + (hd_2)^2\right).$$

Therefore,

$$||\nabla v_1||^2_{0,\tilde{T}} + ||\nabla v_2||^2_{0,\tilde{T}} \geq Ch^2\left((b_1^-)^2 + (c_1^-)^2 + (hd_1)^2 + (b_2^-)^2 + (c_2^-)^2 + (hd_2)^2\right).$$

Using (3.35) with the equivalence property of $||\cdot||_1$ and $||\cdot||_2$ norms in $\mathbb{R}^n$, we obtain

$$||\nabla v_1||^2_{0,\tilde{T}} + ||\nabla v_2||^2_{0,\tilde{T}} \geq Ch^2\left((b_1^+)^2 + (c_1^+)^2 + (hd_1)^2 + (b_2^+)^2 + (c_2^+)^2 + (hd_2)^2\right).$$

The bound (3.42) follows using $|\tilde{T}| \leq h^2$. □

Next, we state the modified trace inequality for piecewise polynomials in $S_h(T)$.

Theorem 3.4.9. *Let T be an interface element with interface edge e as shown in Figure 3.3. Then, for all $\mathbf{v} = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix} \in S_h(\Omega)$ defined on T as in (3.30), we have*

$$||\nabla \mathbf{u}||_{0,e} \leq C\frac{h^{1/2}}{|T|^{1/2}}||\nabla \mathbf{u}||_{0,T}. \tag{3.43}$$

Proof. We recall that the interface point D in Figure 3.3 is defined as $D = (0, \gamma h)$ with $0 < \gamma < 1$.

By direct computations, we obtain,

$$||v_{i,x}^+||^2_{0,e^+} = (b_i^+)^2\gamma h + b_i^+ d_i \gamma^2 h^2 + \frac{1}{3}\gamma^3 d_i^2 h^3$$
$$\leq C((b_i^+)^2 + (hd_i)^2)h, \quad \text{(Young's inequality)}$$
$$||v_{i,y}^+||^2_{0,e^+} = (c_i^+)^2\gamma h \leq (c_i^+)^2 h,$$
$$||v_{i,x}^-||^2_{0,e^-} = (b_i^-)^2 h(1-\gamma) + b_i^- d_i h^2(1-\gamma^2) + \frac{1}{3}d_i^3 h^3(1-\gamma^3)$$

$$\leq Ch((b_i^-)^2 + (hd_i)^2).$$
$$||v_{i,y}^-||_{0,e^-}^2 = (c_i^-)^2(h - \gamma h) \leq (c_i^-)^2 h,$$

Applying (3.42), we obtain,

$$||v_{i,x}^+||_{0,e^+}^2 \leq C\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2,$$
$$||v_{i,y}^+||_{0,e^+}^2 \leq C\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2,$$
$$||v_{i,x}^-||_{0,e^-}^2 \leq C\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2,$$
$$||v_{i,y}^-||_{0,e^-}^2 \leq C\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2.$$

Therefore,

$$||v_{i,r}||_{0,e}^2 \leq C\frac{h}{|\tilde{T}|}||\nabla \mathbf{u}||_{0,\tilde{T}}^2 \leq C\frac{h}{|T|}||\nabla \mathbf{u}||_{0,T}^2, \quad r = x, y, \quad i = 1, 2.$$

The result follows by summing over all partial derivatives. □

In the next Lemma, we extend the inverse inequality to the IFE basis functions.

Lemma 3.4.10. *For all* $\mathbf{v} = \begin{pmatrix} \mathbf{u}(x,y) \\ v_3(x,y) \end{pmatrix} \in S_h(\Omega)$ *defined in (3.30) on an interface element* T*, there exists a constant* $C > 0$ *independent of the mesh size and the interface location such that*

$$|\mathbf{u}|_{W^{1,\infty}(T)} \leq Ch^{-1}|\mathbf{u}|_{1,T}, \tag{3.44}$$

Proof. Consider an interface element $T = T^- \cup T^+$. Then

$$|\mathbf{u}|_{W^{1,\infty}(T)} = |v_1|_{W^{1,\infty}(T)} + |v_2|_{W^{1,\infty}(T)} = \max\{|\mathbf{u}|_{1,\infty,T^-}, |\mathbf{u}|_{1,\infty,T^+}\}$$

$$\leq \max\{|b_1^-| + |c_1^-| + h|d_1| + |b_2^-| + |c_2^-| + h|d_2|, |b_1^+| + |c_1^+| + h|d_1| + |b_2^+| + |c_2^+| + h|d_2|\}.$$

Using (3.36) and (3.42), we obtain

$$|\mathbf{u}|_{1,\infty,T} \leq C(|b_1^+| + |c_1^+| + h|d_1| + |b_2^+| + |c_2^+| + h|d_2|)$$
$$\leq Ch^{-1}||\nabla \mathbf{u}||_{0,T} \leq Ch^{-1}|\mathbf{u}|_{1,T}.$$

□

Chapter 4

Immersed finite element method for the Stokes interface problem

In this chapter, we introduce the discontinuous Galerkin immersed finite element (DG-IFE) method for Stokes interface problem (1.3). Note that, due to its complexity, the Stokes problem has to be solved with suitably chosen finite element spaces and special finite element functions have to be used in order to get satisfactory approximations. In this chapter, we consider the Q_1/Q_0 finite element function with discontinuous Galerkin methods used in [120] . This choice is motivated by two major reasons

- The primal finite element method with the Q_1/Q_0 FE function is not stable [23], while the discontinuous Galerkin method leads to a stable formulation [120].
- The IFE basis functions are not continuous across element boundaries, which is not required by the discontinuous Galerkin method.

First let us recall the standard Q_1/Q_0 DG finite element method. We partition our domain Ω into a Cartesian mesh $\mathcal{T}_h$. On every element, we use the standard Lagrange bilinear basis functions to approximate the two components of the velocity, and use piecewise constant basis functions to approximate the pressure.

Next, define the spaces

$$\mathcal{H}^m(\Omega) = \{(\mathbf{u}, p) : \mathbf{u} \in C^0(\Omega), \mathbf{u} \in (H^m(A))^2 \text{ and } p \in H^1(A), \quad \forall A \subset \Omega^{\pm}\},$$

where H^m is the standard Sobolev space. Furthermore, we introduce

$$\mathcal{H}_0^m(\Omega) = \{(\mathbf{u}, p) \in \mathcal{H}^m(\Omega) : \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega\}.$$

4.1 Weak formulation and finite element scheme

In order to construct the weak formulation we assume that $(\mathbf{u}, p) \in \mathcal{H}^2(\Omega)$ is the solution of the Stokes interface problem (1.3). On an arbitrary edge $e \in \mathcal{E}_h$ shared by two elements T^R and T^L we use the average defined in (2.8) and define the jump of a vector function $\mathbf{u}$ across an edge e as

$$[\mathbf{u}]_e = (\mathbf{u}|_{T^R} - \mathbf{u}|_{T^L}).$$

On a boundary edge $e \subset \partial\Omega$ we define the jump by

$$[\mathbf{u}]_e = \mathbf{u}|_e.$$

Following the standard procedure given in [120] for instance, we multiply the system (1.3a-1.3b) by $(\mathbf{v}, q) \in \mathcal{H}_0^1(\Omega)$ and integrate over an arbitrary non-interface element $T \in \mathcal{T}_h$ to obtain

$$-\int_T (\nabla \cdot \mathbf{S}(\mathbf{u}, p)) \cdot \mathbf{v} dx - \int_T \nabla \cdot \mathbf{u} q dx = \int_T \mathbf{f} \cdot \mathbf{v} dx. \tag{4.1}$$

$$\tag{4.2}$$

We apply the divergence theorem to integrate the first term by parts and obtain

$$-\int_{\partial T} (\mathbf{S}(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds + \int_T \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} dx - \int_T \nabla \cdot \mathbf{u} q dx = \int_T \mathbf{f} \cdot \mathbf{v} dx, \tag{4.3}$$

where $\mathbf{A} : \mathbf{B} = \sum_{i=1}^2 \sum_{j=1}^2 A_{ij} B_{ij}$ and $(\mathbf{S}(\mathbf{u}, p)\mathbf{n})_i = \sum_{j=1}^2 s_{ij} n_j$ with s_{ij} being the entries of $\mathbf{S}(\mathbf{u}, p)$.

On an interface element T, we apply the divergence theorem on $T \cap \Omega^+$ and $T \cap \Omega^-$ and take into account that the exact solution satisfies the jump conditions (1.3f) and (1.3g) across the interface to obtain

$$-\int_{\partial T} (\mathbf{S}(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds + \int_T \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} dx - \int_T \nabla \cdot \mathbf{u} q dx = \int_T \mathbf{f} \cdot \mathbf{v} dx + \int_{T \cap \Gamma} \boldsymbol{\sigma} \cdot \mathbf{v} ds. \tag{4.4}$$

Summing over all elements leads to

$$-\sum_{T \in \mathcal{T}_h} \int_{\partial T} (\mathbf{S}(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds + \int_\Omega \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} dx - \int_\Omega \nabla \cdot \mathbf{u} q dx = \int_\Omega \mathbf{f} \cdot \mathbf{v} dx + \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} ds. \tag{4.5}$$

By rewriting (4.5), we obtain

$$-\sum_{e \in \mathcal{E}_h} \int_e [(\mathbf{S}(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v}] ds + \int_\Omega \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} dx - \int_\Omega \nabla \cdot \mathbf{u} q dx = \int_\Omega \mathbf{f} \cdot \mathbf{v} dx + \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} ds. \tag{4.6}$$

Applying the identity $ab - cd = \frac{1}{2}(a+c)(b-d) + \frac{1}{2}(a-c)(b+d)$ yields

$$\begin{aligned} &- \sum_{e\in\mathcal{E}_h} \int_e \left([(\mathbf{S}(\mathbf{u},p)\mathbf{n})] \cdot \{\mathbf{v}\} + \{(\mathbf{S}(\mathbf{u},p)\mathbf{n})\} \cdot [\mathbf{v}]\right) ds \\ &+ \int_\Omega \mathbf{S}(\mathbf{u},p) : \nabla \mathbf{v} dx - \int_\Omega \nabla \cdot \mathbf{u} q dx = \int_\Omega \mathbf{f} \cdot \mathbf{v} dx + \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} ds. \end{aligned} \tag{4.7}$$

Every interface edge can be expressed as $e = e^+ \cup e^-$, where $e^\pm = e \cap \Omega^\pm$. Hence

$$\int_e [\mathbf{S}(\mathbf{u},p)\mathbf{n}] \cdot \{\mathbf{v}\} ds = \int_{e^+} [\mathbf{S}(\mathbf{u},p)\mathbf{n}] \cdot \{\mathbf{v}\} ds + \int_{e^-} [\mathbf{S}(\mathbf{u},p)\mathbf{n}] \cdot \{\mathbf{v}\} ds.$$

Since $(\mathbf{u}, p) \in \mathcal{H}^2(\Omega)$, we write

$$\int_e [\mathbf{S}(\mathbf{u},p)\mathbf{n}] \cdot \{\mathbf{v}\} ds = 0, \tag{4.8}$$

which also holds for non-interface edges.

Combining (4.7) and (4.8) with $(\mathbf{u}, p) \in \mathcal{H}^2(\Omega)$ and assuming $\mathbf{u}$ continuous lead to the interior penalty weak formulation

$$\begin{aligned} &\int_\Omega \mathbf{S}(\mathbf{u},p) : \nabla \mathbf{v} dx - \int_\Omega \nabla \cdot \mathbf{u} q dx - \sum_{e\in\mathcal{E}_h} \int_e \{(\mathbf{S}(\mathbf{u},p)\mathbf{n})\} \cdot [\mathbf{v}] ds \\ &+ \sum_{e\in\mathcal{E}_h} \frac{\alpha}{h_e} \int_e \nu [\mathbf{u}] \cdot [\mathbf{v}] ds + \gamma \sum_{e\in\mathcal{E}_h} \int_e \{(\nu \boldsymbol{\epsilon}(\mathbf{v})\mathbf{n})\} \cdot [\mathbf{u}] ds + \sum_{e\in\mathcal{E}_h} \int_e \{q\}[\mathbf{u}] \cdot \mathbf{n} = \int_\Omega \mathbf{f} \cdot \mathbf{v} dx \\ &+ \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} ds + \gamma \sum_{e\subset\partial\Omega} \int_e \{(\nu \boldsymbol{\epsilon}(\mathbf{v})\mathbf{n})\} \cdot [\mathbf{g}] ds + \sum_{e\subset\partial\Omega} \int_e \{q\}[\mathbf{g}] \cdot \mathbf{n} + \sum_{e\subset\partial\Omega} \frac{\alpha}{h_e} \int_e \nu [\mathbf{g}] \cdot [\mathbf{v}] ds, \end{aligned} \tag{4.9}$$

where α is a positive stabilization parameter and $\gamma = 1$ for the nonsymmetric weak formulation (NIPG) while $\gamma = -1$ for the symmetric weak formulation (SIPG).

The advantages of adding penalty terms are discussed in [120]. The fifth and sixth terms are added to ensure the symmetry of the scheme in the case $\gamma = -1$, and are balanced using the third and fourth terms on the right hand side. The fourth term is a penalty term used to obtain a scheme with optimal order of convergence and is balanced using the last term in the right hand side. Note that, when $\gamma = \alpha = 0$, the scheme is not guaranteed to have a unique solution.

The weak form above leads to the interior penalty formulation for the Stokes interface problem consisting of finding $(\mathbf{u}, p) \in \mathcal{H}^1(\Omega)$ with $\mathbf{u} = \mathbf{g}$, on $\partial\Omega$, such that

$$\begin{cases} A(\mathbf{u},\mathbf{v}) + B(\mathbf{v},p) & = L(\mathbf{v}) \\ B(\mathbf{u},q) & = l(q) \end{cases}, \qquad \forall\ (\mathbf{v},q) \in \mathcal{H}^1_0(\Omega), \tag{4.10a}$$

where

$$A(\mathbf{w},\mathbf{v}) = \int_\Omega \nu\boldsymbol{\epsilon}(\mathbf{w}) : \nabla\mathbf{v}dx - \sum_{e\in\mathcal{E}_h}\int_e \nu\{\boldsymbol{\epsilon}(\mathbf{w})\mathbf{n}\}\cdot[\mathbf{v}]ds$$
$$+\gamma\sum_{e\in\mathcal{E}_h}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\}\cdot[\mathbf{w}] + \sum_{e\in\mathcal{E}_h}\frac{\alpha}{h_e}\int_e \nu[\mathbf{u}]\cdot[\mathbf{v}]ds, \tag{4.10b}$$

$$B(\mathbf{v},q) = -\int_\Omega q\nabla\cdot\mathbf{v}dx + \sum_{e\in\mathcal{E}_h}\int_e\{q\}[\mathbf{v}]\cdot\mathbf{n}ds, \tag{4.10c}$$

$$L(\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}dx + \int_\Gamma \boldsymbol{\sigma}\cdot\mathbf{v}ds + \gamma\sum_{e\subset\partial\Omega}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\}\cdot[\mathbf{g}]ds + \sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}]ds, \tag{4.10d}$$

$$l(q) = \sum_{e\subset\partial\Omega}\int_e\{q\}[\mathbf{g}]\cdot\mathbf{n}ds.$$

Then, this formulation leads to the DG-IFE method consisting of finding $(\mathbf{u}_h, p_h)\in S_h(\Omega)\oplus\{\mathbf{q_h}(\mathbf{x},\mathbf{y})\}$, such that

$$\begin{cases} A(\mathbf{u}_h,\mathbf{v}_h) + B(\mathbf{v}_h,p_h) & = L_h(\mathbf{v}_h) \\ B(\mathbf{u}_h,q_h) & = l(q_h) \end{cases}, \qquad \forall\ (\mathbf{v}_h,q_h)\in S_{h,0}(\Omega), \tag{4.11}$$

subject to the boundary conditions:

$$\mathbf{u}_h(R) = \mathbf{g}(R), \quad \text{for all mesh vertices } R\in\partial\Omega,$$

where

$$L_h(\mathbf{v}_h) = \int_\Omega \mathbf{f}\cdot\mathbf{v}_h dx + \sum_{T\in\mathcal{T}_h^i}\int_{\overline{DE}}\boldsymbol{\sigma}_h\cdot\mathbf{v}_h ds + \gamma\sum_{e\subset\partial\Omega}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v}_h)\mathbf{n}\}\cdot[\mathbf{g}]ds$$
$$+\sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}_h]ds, \tag{4.12}$$

$\boldsymbol{\sigma}_h$ is the linear interpolant of $\boldsymbol{\sigma}$ on $\overline{DE}$ using the value of $\boldsymbol{\sigma}$ at D and E and

$$S_{h,0}(\Omega) = \{(\mathbf{v},p)\in S_h(\Omega):\ \mathbf{v}(R) = \mathbf{0},\ \text{ for all mesh vertices } R\in\partial\Omega\}.$$

Remark. *On every interface element $T = \square(A_1A_2A_3A_4)$ with an edge $e = [A_1,A_2]$ lying on the boundary $\partial\Omega$, we use the boundary condition $\mathbf{g} = (g_1,g_2)^T$ and rewrite the IFE solution $(\mathbf{u}_h,p_h)$ in (3.25) as*

$$\begin{pmatrix}\mathbf{u}_h\\ p_h\end{pmatrix} = g_1(A_1)\boldsymbol{\Phi}_1 + g_1(A_2)\boldsymbol{\Phi}_2 + g_2(A_1)\boldsymbol{\Phi}_5 + g_2(A_2)\boldsymbol{\Phi}_6 + \sum_{i=3}^{4}c_i\boldsymbol{\Phi}_i + \sum_{i=7}^{9}c_i\boldsymbol{\Phi}_i + \kappa_1\Upsilon_1 + \kappa_2\Upsilon_2.$$

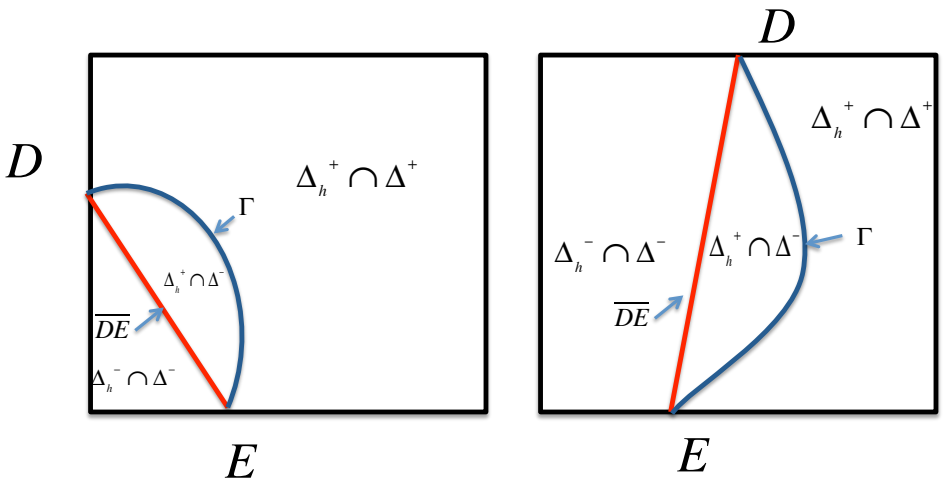


Figure 4.1: The regions $\Delta^{\pm}$, $\Delta_h^{\pm}$ and their intersections.

4.2 Numerical simulations

Here we investigate the approximation properties of the IFE spaces and IFE particular functions and the convergence properties of the proposed DG-IFE method for the Stokes problem. L^2 and H^1 errors on non-interface elements are computed using standard Gauss quadratures. However, extra care is needed near the interface where we assume that every interface element Δ is split by the true interface as $\Delta = \Delta^+ \cup \Delta^-$. We further assume that the piecewise linear approximate interface split the domain Ω into Ω_h^+ and Ω_h^- and $\Delta = \Delta_h^+ \cup \Delta_h^-$. Here we recall the notation $\Delta^{\pm} = \Delta \cap \Omega^{\pm}$ and $\Delta_h^{\pm} = \Delta \cap \Omega_h^{\pm}$. For, instance, the true L^2 error of the pressure on such interface element Δ is computed as

$$||p-p_h||^2_{0,\Delta} = ||p^+-p_h^+||^2_{0,\Delta^+\cap\Delta_h^+} + ||p^- - p_h^-||^2_{0,\Delta^-\cap\Delta_h^-} + ||p^+-p_h^-||^2_{0,\Delta^+\cap\Delta_h^-} + ||p^- - p_h^+||^2_{0,\Delta^-\cap\Delta_h^+}.$$

The subregions $\Delta^+\cap\Delta_h^+$ and $\Delta^-\cap\Delta_h^-$ may be split into triangles with at most one curved side for which Gauss quadratures exist. The subregions $\Delta^+\cap\Delta_h^-$ and $\Delta^-\cap\Delta_h^+$ are bounded by the true interface and its linear approximation. Elementary numerical quadrature methods on such regions can be found in elementary numerical analysis books such as [30]. We illustrate an example of different regions that may arise in our computations in Figure 4.1.

4.2.1 Approximation properties of Q_1/Q_0 IFE spaces

First, we investigate the approximation capabilities of our $S_h(\Omega)$ and $\tilde{S}_h(\Omega)$ spaces by computing interpolation errors and their convergence rates under mesh refinement for the $S_h(\Omega)$ space and by computing the L^2 projection errors and their convergence rates under mesh

refinement for the $\tilde{S}_h(\Omega)$ space. The interpolation error cannot be computed for the space $\tilde{S}_h(\Omega)$ since the nodal property is not satisfied by the corresponding IFE shape functions.

Here we consider $\mathcal{T}_h$ a Cartesian mesh for the domain Ω and, since we are interested in solving Stokes interface problems, we only test the approximation capabilities of our IFE space on functions in the set

$$\mathcal{S}(\Omega) = \{(\mathbf{u}, p) \mid [\mathbf{u}]_\Gamma = \mathbf{0}, [\mathbf{S}(\mathbf{u}, p)\mathbf{n}]|_\Gamma = \boldsymbol{\sigma}\}.$$

The piecewise polynomial interpolant $\mathbf{I}_h(\mathbf{u}, p) \in S_h(\Omega)$ of a vector function $(\mathbf{u}, p) \in \mathcal{S}(\Omega)$ on an element $T = \square A_1A_2A_3A_4$ is defined by

$$\mathbf{I}_h(\mathbf{u}, p)|_T = \begin{cases} \sum_{i=1}^4 u_1(A_i)\boldsymbol{\Phi}_i + \sum_{i=5}^8 u_2(A_{i-4})\boldsymbol{\Phi}_i + p_T\boldsymbol{\Phi}_9 + \sum\limits_{j=1}^{2} \kappa_j \boldsymbol{\Upsilon}_j, & \text{if } T \in \mathcal{T}_h^i, \\ \\ \sum_{i=1}^4 u_1(A_i)\boldsymbol{\Psi}_i + \sum_{i=5}^8 u_2(A_{i-4})\boldsymbol{\Psi}_i + p_T\boldsymbol{\Psi}_9, & \text{if } T \in \mathcal{T}_h^n, \end{cases} \tag{4.13}$$

where $p_T = \int_T p dX$, $\kappa_j = \int_{\overline{DE}} \sigma_{h,j} ds$ and $\boldsymbol{\Phi}_i$, $\boldsymbol{\Psi}_i$ and $\boldsymbol{\Upsilon}_j$, respectively, are the Q_1/Q_0 IFE shape functions, standard shape functions and particular IFE functions associated with element T. The i^{th} component $\mathbf{I}_h(\mathbf{u}, p)$ denoted by $(\mathbf{I}_h(\mathbf{u}, p))_i$ is such that $(\mathbf{I}_h(\mathbf{u}, p))_1 \approx u_1$, $(\mathbf{I}_h(\mathbf{u}, p))_2 \approx u_2$ and $(\mathbf{I}_h(\mathbf{u}, p))_3 \approx p$ with errors $\epsilon(u_i) = u_i - (\mathbf{I}_h(\mathbf{u}, p))_i$, for $i = 1, 2$ and $\epsilon(p) = p - (\mathbf{I}_h(\mathbf{u}, p))_3$.

The L^2 projection, denoted as $\Pi_h(\mathbf{u}, p) \in \tilde{S}_h(\Omega)$, of a vector function $(\mathbf{u}, p) \in \mathcal{S}(\Omega)$ on an interface element $T = \square A_1A_2A_3A_4$ is defined as follows:

$$\Pi_h(\mathbf{u}, p)|_T = \sum_{i=1}^{10} c_i \boldsymbol{\Phi}_i, \tag{4.14}$$

where c_i, $i = 1, 2, ..., 10$ are the entries of the vector $\mathbf{c}$ obtained by solving the linear system

$$\mathbf{A}\mathbf{c} = \mathbf{b}, \tag{4.15}$$

where the entries $a_{i,j}$, $i, j = 1, 2, ..., 10$, of the matrix $\mathbf{A}$ are defined as

$$a_{i,j} = \int_T \boldsymbol{\Phi}_i \cdot \boldsymbol{\Phi}_j,$$

and the entries b_i, $i = 1, 2, ..., 10$ of the vector $\mathbf{b}$ are defined as

$$b_i = \int_T \boldsymbol{\Phi}_i \cdot \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix},$$

and $\boldsymbol{\Phi}_i$, $i = 1, 2, ..., 10$, are the basis functions of the space $\tilde{S}_h(T)$.

If T is a non-interface element, the standard Lagrange basis functions $\mathbf{\Psi}_i$, $i = 1, 2, ..., 9$ are used, and the L^2 projection is obtained as follows

$$\Pi_h(\mathbf{u}, p)|_T = \sum_{i=1}^{9} c_i \mathbf{\Psi}_i, \tag{4.16}$$

where c_i, $i = 1, 2, ..., 9$ are the entries of the vector $\mathbf{c}$ obtained by solving the linear system

$$\mathbf{A}\mathbf{c} = \mathbf{b}, \tag{4.17}$$

where the entries $a_{i,j}$, $i, j = 1, 2, ..., 9$, of the matrix $\mathbf{A}$ are defined as

$$a_{i,j} = \int_T \mathbf{\Psi}_i \cdot \mathbf{\Psi}_j,$$

and the entries b_i, $i = 1, 2, ..., 9$ of the vector $\mathbf{b}$ are defined as

$$b_i = \int_T \mathbf{\Psi}_i \cdot \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}.$$

In the case where the projection error is computed, $\epsilon(u_i)$, $i = 1, 2$ and $\epsilon(p)$ will refer to the L^2 projection errors in velocity and pressure. To investigate the order of convergence, we consider three example functions.

Example 4.2.1.

We consider the domain $\Omega = [-1, 1]^2$ cut by the linear interface $y = -x - \sqrt{0.3}$ which splits Ω into $\Omega^+ = \{(x, y) \in \Omega : y + x + \sqrt{0.3} > 0\}$ and $\Omega^- = \{(x, y) \in \Omega : y + x + \sqrt{0.3} < 0\}$. We select the following function $(\mathbf{u}, p) \in \mathcal{S}(\Omega)$ given by

$$\mathbf{u} = \begin{cases} \mathbf{u}^+ = \begin{pmatrix} x^2 + y + y^2 \\ \\ \frac{-2x(x+\nu^- y)}{\nu^-} \end{pmatrix}, & \text{if } (x, y) \in \Omega^+, \\ \\ \mathbf{u}^- = \begin{pmatrix} \frac{-\nu^+(0.3-x^2+2\sqrt{0.3}y+y^2)+\nu^-(0.3+y+2\sqrt{0.3}y+2y^2)}{\nu^-} \\ \\ \frac{2x(-x+\nu^-(\sqrt{0.3}+x)-\nu^+(\sqrt{0.3}+x+y))}{\nu^-} \end{pmatrix}, & \text{if } (x, y) \in \Omega^-, \end{cases} \tag{4.18a}$$

and the pressure is defined as follows

$$p = \begin{cases} \frac{15(-43+4\sqrt{30})(\nu^-)^2-2\nu^+(310+3\sqrt{30}+30(-43+4\sqrt{30})\sqrt{0.3}+2400x)}{1200\nu^-}+ \\ \frac{\nu^-(620+6\sqrt{30}+60(-43+4\sqrt{30})\sqrt{0.3}-15\nu^+(-43+4\sqrt{30}+320x))}{1200\nu^-}, \quad \text{if } (x,y)\in\Omega^+, \\ \frac{15(37+4\sqrt{30})(\nu^-)^2-2\nu^+(310+3\sqrt{30}+30(37+4\sqrt{30})\sqrt{0.3}+2400x+2400y)}{(1200)\nu^-}+ \\ \frac{\nu^-(620+6\sqrt{30}+60(37+4\sqrt{30})\sqrt{0.3}-15\nu^+(37+4\sqrt{30}+320x)+4800y)}{(1200)\nu^-}, \text{if } (x,y)\in\Omega^-, \end{cases} \tag{4.18b}$$

with $\nu^- = 1$, $\nu^+ = 10$ and $\boldsymbol{\sigma} = \mathbf{0}$. We create uniform Cartesian meshes having $N^2 = 5^2, 10^2, 20^2, 40^2, 80^2, 160^2$ square elements and present L^2 interpolation errors using the space $S_h(\Omega)$ in Table 4.1. Broken H^1 interpolation errors for the velocity are shown in Table 4.2 for the space $S_h(\Omega)$. On the other hand, we present the L^2 errors of the L^2 projection of the space $\tilde{S}_h(\Omega)$ in Table 4.3 and the broken H^1 error of the L^2 projection of the space $\tilde{S}_h(\Omega)$ in Table 4.4. The broken H^1 norm for functions in $\mathcal{S}(\Omega)$ is defined as follows:

$$||\cdot||_{1,\Omega} = ||\cdot||_{1,\Omega^+} + ||\cdot||_{1,\Omega^-}.$$

N	$\|\|\epsilon(u_1)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(p)\|\|_{0,\Omega}$	Order
5	1.8197e-03	NA	5.5240e-03	NA	2.3704e-01	NA
10	4.7976e-04	1.9233	1.4501e-03	1.9296	1.1271e-01	1.0726
20	1.1915e-04	2.0095	3.7080e-04	1.9674	5.1155e-02	1.1396
40	2.9911e-05	1.9941	9.4230e-05	1.9764	2.4874e-02	1.0402
80	7.4878e-06	1.9980	2.3649e-05	1.9944	1.2160e-02	1.0325
160	1.8734e-06	1.9989	5.9242e-06	1.9971	6.0057e-03	1.0177

Table 4.1: L^2 interpolation errors for Example 4.2.1 using the space $S_h(\Omega)$.

This numerical example suggests that the proposed Q_1/Q_0 IFE spaces maintain the optimal convergence rates of Q_1/Q_0 spaces observed for smooth functions in [120] and non-smooth functions on fitted-meshes in [59].

Example 4.2.2.

We consider the domain $\Omega = [-1,1]^2$ cut by the circular interface $x^2+y^2 = 0.3$ that separates Ω into two regions $\Omega^+ = \{(x,y)\in\Omega : x^2+y^2 > 0.3\}$ and $\Omega^- = \{(x,y)\in\Omega : x^2+y^2 < 0.3\}$. The functions $\mathbf{u}$ and p are chosen to satisfy jump conditions (1.3f) and (1.3g) and are given

N	$\|\|\epsilon(u_1)\|\|_{1,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{1,\Omega}$	Order
5	3.0294e-02	NA	4.6821e-02	NA
10	1.5541e-02	0.9629	2.3890e-02	0.9707
20	7.7082e-03	1.0117	1.1949e-02	0.9995
40	3.8663e-03	0.9954	5.9991e-03	0.9940
80	1.9319e-03	1.0010	3.0026e-03	0.9985
160	9.6557e-04	1.0006	1.5021e-03	0.9992

Table 4.2: Broken H^1 interpolation errors in the velocity for Example 4.2.1 using the space $S_h(\Omega)$.

N	$\|\|\epsilon(u_1)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(p)\|\|_{0,\Omega}$	Order
5	2.2505e-02	NA	2.2505e-02	NA	4.1878e-02	NA
10	6.8965e-03	1.7063	6.8965e-03	1.7063	9.2471e-02	-1.1427
20	1.9150e-03	1.8484	1.9150e-03	1.8484	3.3636e-02	1.4589
40	4.9301e-04	1.9576	4.9301e-04	1.9576	1.1397e-02	1.5612
80	1.2549e-04	1.9740	1.2549e-04	1.9740	4.4902e-03	1.3438
160	3.1629e-05	1.9882	3.1629e-05	1.9882	1.8539e-03	1.2762

Table 4.3: L^2 errors of the L^2 projection for Example 4.2.1 using the space $\tilde{S}_h(\Omega)$.

N	$\|\|\epsilon(u_1)\|\|_{1,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{1,\Omega}$	Order
5	1.5918e-01	NA	1.5918564e-01	NA
10	9.3038e-02	0.7748	9.3038e-02	0.7748
20	5.0793e-02	0.8731	5.0793e-02	0.8731
40	2.5777e-02	0.9785	2.5777e-02	0.9785
80	1.3028e-02	0.9844	1.3028e-02	0.9844
160	9.6557e-04	1.0006	1.5021e-03	0.9992

Table 4.4: Broken H^1 errors of the L^2 projection of the velocity for Example 4.2.1 using the space $\tilde{S}_h(\Omega)$.

by

$$\mathbf{u} = \begin{cases} \mathbf{u}^+ = \begin{pmatrix} \frac{y(x^2+y^2-0.3)}{\nu^+} \\ \\ \frac{x(x^2+y^2-0.3)}{\nu^+} \end{pmatrix}, & \text{if } (x,y) \in \Omega^+, \\ \\ \mathbf{u}^- = \begin{pmatrix} \frac{y(x^2+y^2-0.3)}{\nu^-} \\ \\ \frac{x(x^2+y^2-0.3)}{\nu^-} \end{pmatrix}, & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{4.19a}$$

and the pressure is defined by

$$p = \frac{1}{10}(x^3 - y^3). \tag{4.19b}$$

where $\nu^- = 1$, $\nu^+ = 10$ and $\boldsymbol{\sigma} = \mathbf{0}$.

Next we create uniform Cartesian meshes having $N^2 = 5^2, 10^2, 20^2, 40^2, 80^2, 160^2, 320^2$ square elements and present the L^2 interpolation errors using the space $S_h(\Omega)$ in Table 4.5 and the broken H^1 interpolation errors for the velocity in Table 4.6.

N	$\|\|\epsilon(u_1)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(p)\|\|_{0,\Omega}$	Order
5	2.2505e-02	0	2.2505e-02	0	4.1879e-02	0
10	6.8966e-03	1.7063	6.8966e-03	1.7063	9.2471e-02	-1.1428
20	1.9151e-03	1.8485	1.9151e-03	1.8485	3.3637e-02	1.4590
40	4.9302e-04	1.9577	4.9302e-04	1.9577	1.1398e-02	1.5613
80	1.2549e-04	1.9740	1.2549e-04	1.9740	4.4903e-03	1.3439
160	3.1629e-05	1.9883	3.1629e-05	1.9883	1.8539e-03	1.2762
320	7.9416e-06	1.9938	7.9416e-06	1.9938	8.1715e-04	1.1819

Table 4.5: L^2 interpolation errors for Example 4.2.2 using the space $S_h(\Omega)$.

N	$\|\|\epsilon(u_1)\|\|_{1,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{1,\Omega}$	Order
5	1.5919e-01	NA	1.5919e-01	NA
10	9.3038e-02	0.7748	9.3038e-02	0.7748
20	5.0794e-02	0.8731	5.0794e-02	0.8731
40	2.5778e-02	0.9785	2.5778e-02	0.9785
80	1.3029e-02	0.9844	1.3029e-02	0.9844
160	6.5516e-03	0.9917	6.5516e-03	0.9917
320	3.2854e-03	0.9957	3.2854e-03	0.9957

Table 4.6: Broken H^1 interpolation errors for Example 4.2.2 using the space $S_h(\Omega)$.

Here we also observe optimal quadratic convergence for the velocity and linear convergence for the pressure. We recall that the circular interface is approximated by a piecewise linear interface.

Example 4.2.3.

We consider the same domain and interface as in the previous example and define the functions $\mathbf{u}$ and p that satisfy the homogeneous jump condition (1.3g) and the non-homogeneous jump condition (1.3f):

$$\mathbf{u} = \begin{cases} \mathbf{u}^+ = \begin{pmatrix} 1 - \frac{y}{\nu^+}\sin(0.3 - x^2 - y^2) \\ \\ 2 + \frac{x}{\nu^+}\sin(0.3 - x^2 - y^2) \end{pmatrix}, & \text{if } (x,y) \in \Omega^+, \\ \\ \mathbf{u}^- = \begin{pmatrix} 1 - \frac{y}{\nu^-}\sin(0.3 - x^2 - y^2) \\ \\ 2 + \frac{x}{\nu^-}\sin(0.3 - x^2 - y^2) \end{pmatrix}, & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{4.20a}$$

$$p = \begin{cases} p^+ = e^{x+y} - 1.3798535909816816, & \text{if } (x,y) \in \Omega^+, \\ p^- = \sqrt{1 + x^2 + y^2} - 1.3798535909816816, & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{4.20b}$$

We select $\nu^- = 1$ and $\nu^+ = 10$ and compute the IFE interpolation errors on uniform rectangular meshes and show the interpolation errors and their orders of convergence in Tables 4.7 and 4.8.

N	$\|\|\epsilon(u_1)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(p)\|\|_{0,\Omega}$	Order
5	2.0116e-02	NA	1.9965e-02	NA	5.6124e-01	NA
10	6.4161e-03	1.6486	6.4032e-03	1.6406	2.9540e-01	0.9259
20	1.8107e-03	1.8251	1.8098e-03	1.8230	1.4454e-01	1.0312
40	4.6792e-04	1.9522	4.6788e-04	1.9516	7.1237e-02	1.0207
80	1.1936e-04	1.9710	1.1935e-04	1.9709	3.5461e-02	1.0064
160	3.0110e-05	1.9870	3.0110e-05	1.9869	1.7688e-02	1.0034
320	7.5635e-06	1.9931	7.5635e-06	1.9931	8.8340e-03	1.0016

Table 4.7: L^2 interpolation errors for Example 4.2.3 using the space $S_h(\Omega)$.

Again, the numerical results suggest that our IFE spaces and IFE particular functions maintain the optimal convergence rates in the presence of singularity across the interface.

4.2.2 Convergence of the immersed finite element method

We use numerical results to show the approximation capability of our finite element method. First we recall from [59] that using the finite element spaces Q_1/Q_0 on fitted meshes the L^2

N	$\|\|\epsilon(u_1)\|\|_{1,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{1,\Omega}$	Order
5	1.4584e-01	NA	1.4582e-01	NA
10	8.7277e-02	0.7407	8.7276e-02	0.7405
20	4.8239e-02	0.8554	4.8239e-02	0.8554
40	2.4528e-02	0.9757	2.4528e-02	0.9757
80	1.2412e-02	0.9827	1.2412e-02	0.9827
160	6.2452e-03	0.9908	6.2452e-03	0.9908
320	3.1327e-03	0.9953	3.1327e-03	0.9953

Table 4.8: Broken H^1 interpolation errors for Example 4.2.3 using the space $S_h(\Omega)$.

errors of the velocity and pressure, respectively, are $O(h^2)$ and $O(h)$ convergent. Furthermore, the broken H^1 error of the velocity is $O(h)$ convergent. The finite element errors are denoted by $e(u_i) = u_i - u_{i,h}$ for $i = 1, 2$ and $e(p) = p - p_h$.

Example 4.2.4.

First we consider Stokes interface problem where the exact solution is given by (4.18) with a linear interface on $[-1, 1]^2$ and $\nu^- = 1$ and $\nu^+ = 10$. We use the meshes from Example 4.2.1 to carry out several simulations by solving (4.11) with SIPG and NIPG methods using the immersed space $S_h(\Omega)$ and show the L^2 IFE errors in Tables 4.9 and 4.10 and the broken H^1 errors in Tables 4.11 and 4.12. The computational results show optimal quadratic convergence for the velocity and linear convergence for the pressure. The broken H^1 errors show linear convergence for the velocity. On the other hand, we use the same formulation (4.11) with SIPG and NIPG methods using the space $\tilde{S}_h(\Omega)$ and show the L^2 IFE errors in Tables 4.13 and 4.14 and the broken H^1 errors in Tables 4.15 and 4.16. The computational results show optimal quadratic convergence for the velocity in the L^2 norm and linear convergence in the broken H^1 norm. However, in Table 4.14, the order of convergence of the pressure is oscillating. In this case, we use a linear regression of $log(||e(p)||_{0,\Omega})$ against $log(h)$. The order of convergence is 0.96 which satisfies the expected order shown in the previous experiments.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	3.3087e-03	NA	3.8791e-03	NA	2.5120e-01	NA
10	9.0140e-04	1.8760	8.7783e-04	2.1437	1.0615e-01	1.2427
20	2.2736e-04	1.9872	2.0301e-04	2.1124	4.9161e-02	1.1105
40	5.7379e-05	1.9864	4.8446e-05	2.0671	2.3896e-02	1.0407
80	1.4565e-05	1.9780	1.1897e-05	2.0258	1.1894e-02	1.0065
160	3.6736e-06	1.9873	2.9492e-06	2.0122	5.9394e-03	1.0019

Table 4.9: L^2 IFE errors for SIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	3.4272e-03	NA	3.9838e-03	NA	2.7117e-01	NA
10	9.0967e-04	1.9136	8.8799e-04	2.1655	1.1021e-01	1.2990
20	2.2703e-04	2.0025	2.0535e-04	2.1125	4.9935e-02	1.1421
40	5.5829e-05	2.0238	4.8465e-05	2.0830	2.4162e-02	1.0473
80	1.4108e-05	1.9845	1.1846e-05	2.0325	1.1953e-02	1.0154
160	3.5698e-06	1.9825	2.9313e-06	2.0148	5.9517e-03	1.0059

Table 4.10: L^2 IFE error for NIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	3.2948e-02	NA	4.9512e-02	NA
10	1.6537e-02	0.9945	2.4719e-02	1.0021
20	7.9295e-03	1.0604	1.2108e-02	1.0297
40	3.9223e-03	1.0155	6.0275e-03	1.0063
80	1.9478e-03	1.0099	3.0099e-03	1.0019
160	9.7012e-04	1.0056	1.5042e-03	1.0007

Table 4.11: Broken H^1 IFE errors for SIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	3.3416e-02	NA	4.9495e-02	NA
10	1.6596e-02	1.0097	2.4617e-02	1.0076
20	7.9550e-03	1.0609	1.2093e-02	1.0255
40	3.9165e-03	1.0223	6.0254e-03	1.0050
80	1.9458e-03	1.0092	3.0094e-03	1.0016
160	9.6965e-04	1.0049	1.5040e-03	1.0006

Table 4.12: Broken H^1 IFE errors for NIPG method applied to Example 4.2.4 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	2.0218e-02	NA	2.0218e-02	NA	4.7817e-02	NA
10	6.7313e-03	1.5867	6.7313e-03	1.5867	8.5664e-02	-0.841
20	1.6983e-03	1.9868	1.6983e-03	1.9868	2.9682e-02	1.5291
40	4.3537e-04	1.9638	4.3537e-04	1.9638	9.2608e-03	1.6804
80	1.1363e-04	1.9378	1.1363e-04	1.9378	4.6147e-03	1.0049
160	2.8176e-05	2.0119	2.8176e-05	2.0119	1.8664e-03	1.3060

Table 4.13: L^2 IFE errors for SIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	2.0254e-02	NA	2.0254e-02	NA	1.6324e-01	NA
10	6.6006e-03	1.6176	6.6006e-03	1.6176	3.8986e-01	-1.256
20	1.3329e-03	2.3081	1.3329e-03	2.3081	8.4541e-02	2.2052
40	3.3742e-04	1.9819	3.3742e-04	1.9819	1.9153e-02	2.1421
80	8.1939e-05	2.0419	8.1939e-05	2.0419	2.8380e-02	-0.567
160	1.9085e-05	2.1021	1.9085e-05	2.1021	9.9622e-03	1.5104

Table 4.14: L^2 IFE errors for NIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	1.6136e-01	NA	1.6136e-01	NA
10	9.3050e-02	0.7942	9.3050e-02	0.7942
20	5.0189e-02	0.8906	5.0189e-02	0.8906
40	2.5758e-02	0.9623	2.5758e-02	0.9623
80	1.3020e-02	0.9842	1.3020e-02	0.9842
160	6.5483e-03	0.9915	6.5483e-03	0.9915

Table 4.15: Broken H^1 IFE errors for SIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	1.6022e-01	0	1.6022e-01	0
10	9.8574e-02	0.7008	9.8574e-02	0.7008
20	5.0825e-02	0.9556	5.0825e-02	0.9556
40	2.5889e-02	0.9731	2.5889e-02	0.9731
80	1.3260e-02	0.9652	1.3260e-02	0.9652
160	6.6716e-03	0.9909	6.6716e-03	0.9909

Table 4.16: Broken H^1 IFE errors for NIPG method applied to Example 4.2.4 using the space $\tilde{S}_h(\Omega)$.

Example 4.2.5.

Here we consider Stokes interface problem on $[-1,1]^2$ with a circular interface and exact solution given by (4.19) with $\nu^- = 1$ and $\nu^+ = 10$. We solve (4.11) using SIPG and NIPG methods on the uniform meshes from the previous example using the space $S_h(\Omega)$. We present the L^2 errors in Tables 4.17 and 4.18 while the H^1 errors are shown in Tables 4.19 and 4.20. The numerical results again suggest quadratic convergence for the velocity and linear convergence for the pressure in the L^2 norm.

The computational results show that, despite the presence of the singularity, the L^2 errors in the velocity and pressure for the DG-IFE methods maintain the optimal orders of convergence shown in [120] for non-interface problems with smooth solutions.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	2.0045e-02	NA	2.0122e-02	NA	5.6755e-01	NA
10	6.6060e-03	1.6014	6.6230e-03	1.6032	2.8468e-01	0.9954
20	1.6771e-03	1.9778	1.6788e-03	1.9800	1.4354e-01	0.9878
40	4.2978e-04	1.9643	4.2983e-04	1.9656	7.1163e-02	1.0123
80	1.1222e-04	1.9373	1.1217e-04	1.9381	3.5401e-02	1.0073
160	2.7808e-05	2.0127	2.7795e-05	2.0128	1.7677e-02	1.0020

Table 4.17: L^2 IFE errors for SIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	1.5657e-02	NA	1.5274e-02	NA	3.6065e-01	NA
10	4.6777e-03	1.7430	4.7450e-03	1.6866	2.3837e-01	0.5974
20	1.2919e-03	1.8563	1.2980e-03	1.8702	9.6466e-02	1.3051
40	3.4281e-04	1.9140	3.4302e-04	1.9199	3.5601e-02	1.4381
80	9.1878e-05	1.8996	9.1895e-05	1.9002	1.2834e-02	1.4719
160	2.2674e-05	2.0187	2.2676e-05	2.0188	4.6366e-03	1.4689

Table 4.18: L^2 IFE errors for NIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$.

Example 4.2.6.

We solve Stokes interface problem with surface force on the domain $[-1,1]^2$ with the true solution (4.20) using SIPG and NIPG methods, the IFE space $S_h(\Omega)$ and IFE particular functions on uniform rectangular meshes and show the IFE errors and convergence orders in Tables 4.21, 4.22, 4.23 and 4.24.

Again, the numerical results show that our immersed Q_1/Q_0 interior penalty methods maintain the optimal convergence rates observed for non-interface problems with smooth solutions.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	1.4788e-01	NA	1.4740e-01	NA
10	8.6853e-02	0.7678	8.6677e-02	0.7660
20	4.7526e-02	0.8698	4.7488e-02	0.8681
40	2.4497e-02	0.9561	2.4490e-02	0.9553
80	1.2395e-02	0.9828	1.2394e-02	0.9825
160	6.2400e-03	0.9901	6.2396e-03	0.9901

Table 4.19: Broken H^1 IFE errors for SIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	1.5691e-01	NA	1.5949e-01	NA
10	9.0996e-02	0.7860	1.2474e-01	0.3546
20	5.4141e-02	0.7490	7.2397e-02	0.7849
40	2.8510e-02	0.9252	3.9634e-02	0.8691
80	1.4935e-02	0.9328	2.1296e-02	0.8961
160	7.5725e-03	0.9798	1.1135e-02	0.9354

Table 4.20: Broken H^1 IFE errors for NIPG method applied to Example 4.2.5 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	9.2041e-03	0	8.5578e-03	0	5.4909e-01	0
10	4.5395e-03	1.0197	4.5505e-03	0.9112	2.7844e-01	0.9796
20	1.4915e-03	1.6057	1.4932e-03	1.6076	1.4054e-01	0.9863
40	3.9612e-04	1.9128	3.9624e-04	1.9140	7.0953e-02	0.9860
80	1.0697e-04	1.8887	1.0693e-04	1.8897	3.5401e-02	1.0031
160	2.7612e-05	1.9774	2.7613e-05	1.9775	1.4127e-03	1.0600

Table 4.21: L^2 IFE errors for SIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	7.4130e-02	0	7.1057e-02	0
10	5.7993e-02	0.3541	5.7955e-02	0.2940
20	4.2999e-02	0.4315	4.2987e-02	0.4310
40	2.3153e-02	0.8930	2.3152e-02	0.8927
80	1.1935e-02	0.9560	1.1935e-02	0.9559
160	7.5730e-03	0.9799	1.1138e-02	0.9355

Table 4.22: Broken H^1 IFE errors for SIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{0,\Omega}$	Order	$\|\|e(u_2)\|\|_{0,\Omega}$	Order	$\|\|e(p)\|\|_{0,\Omega}$	Order
5	9.2914e-03	0	8.6753e-03	0	5.4880e-01	0
10	3.5277e-03	1.3972	3.5297e-03	1.2973	2.8385e-01	0.9511
20	1.0441e-03	1.7564	1.0436e-03	1.7580	1.4428e-01	0.9762
40	2.8611e-04	1.8676	2.8597e-04	1.8677	7.2363e-02	0.9955
80	7.8308e-05	1.8693	7.8220e-05	1.8702	3.5822e-02	1.0144
160	1.9256e-05	2.0238	1.9228e-05	2.0243	1.7832e-02	1.0064

Table 4.23: L^2 IFE errors for NIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$.

N	$\|\|e(u_1)\|\|_{1,\Omega}$	Order	$\|\|e(u_2)\|\|_{1,\Omega}$	Order
5	7.3792e-02	0	7.0814e-02	0
10	5.7363e-02	0.3633	5.7322e-02	0.3049
20	4.2973e-02	04166	4.2966e-02	0.4158
40	2.3151e-02	0.8923	2.3151e-02	0.8921
80	1.1935e-02	0.9559	1.1934e-02	0.9559
160	6.1779e-03	0.9499	6.1779e-03	0.9499

Table 4.24: Broken H^1 IFE errors for NIPG method applied to Example 4.2.6 using the space $S_h(\Omega)$.

Chapter 5

Immersed finite element method for the Stokes problem with a moving interface

5.1 Introduction

The dispersions of one liquid in another immiscible liquid are called emulsions, and the phenomenon arises in several industries such as medicine, oil recovery and material processing. The emulsions usually yield a system consisting of droplets immersed in a matrix fluid separated by interfaces. An assessment of the drop size and shape helps control the physical properties of the emulsions such as viscosity, stability and transport properties, and thus it is necessary to understand the dynamics of drop deformation. This problem was studied in [39, 106, 110].

Usually, conventional methods with body-fitted meshes can be used to solve interface problems modeling emulsions. This has been done using finite difference schemes [116], finite element methods [70, 71], boundary integral and boundary element methods [49, 80, 126] and discontinuous Galerkin finite element methods [59]. In general, optimal convergence rates for the solutions obtained using conventional methods are attained for fitted meshes where every element essentially contains one fluid [15, 27, 40]. This task is relatively simple if the interface does not change shape or location. However, this is hardly the case when dealing with fluids as they move according to their velocity. As a consequence, the use of conventional methods becomes inefficient as a new mesh has to be generated at every time step which is a time-consuming process in particular for applications with moving interfaces having complex geometries. Also as observed in [68], by regenerating a new mesh, a change in the degrees of freedom as well as their location may occur, which will add an extra computational cost and complexity. Another drawback is the loss of the so-called global

matrices assembling feature. In fact, this feature is one of the most desirable property of the FE methods. This procedure consists of constructing a local matrix over every element and its entries are then distributed to the global matrix. If the mesh changes in time, this simplicity is lost and complicated and time consuming quadratures have to be used. As an alternative, methods that do not require remeshing such as the volume of fluid method [90, 91], the front-tracking method [128] and the level-set method [38] have been developed. These methods differ from conventional methods as they do not model the flow over two domains separated by an interface. Instead, they model the flow over the whole domain and represent the surface force as a body force smeared over a narrow region of the interface.

In this chapter, we construct an IFE method to solve the Stokes interface problem with moving interfaces. As we have already discussed in the previous chapters, the mesh does not depend on the interface and therefore, a structured mesh may be used throughout the computations. In [133], a parabolic problem with a moving interface was solved using an IFE method where *a priori* knowledge of the interface location at any time was assumed. However, solving the Stokes problem with a moving interface is more challenging since an analytical expression for the interface is not available. Instead, the interface is driven by the velocity of the fluids and it satisfies the following system of ODEs

$$\begin{cases} \frac{d}{dt}X^\alpha(t) &= \mathbf{u}(X^\alpha(t),t), \quad t\in[0,T] \\ X^\alpha(0) &= X_0^\alpha \end{cases}, \quad \forall X^\alpha(t)\in\Gamma(t), \tag{5.1}$$

where $\mathbf{u}$ is the solution to the transient Stokes interface problem:

$$-\nabla\cdot\mathbf{S}(\mathbf{u}(X,t),p(X,t)) = \mathbf{0}, \quad \text{if } X\in\Omega^-(t)\cup\Omega^+(t), \quad t>0 \tag{5.2a}$$

$$\nabla\cdot\mathbf{u}(X,t) = 0, \quad \text{if } X\in\Omega^-(t)\cup\Omega^+(t), \quad t>0 \tag{5.2b}$$

$$\mathbf{u}(X,t) = \mathbf{g}(X,t), \text{ on } \partial\Omega, \quad t>0 \tag{5.2c}$$

where $\mathbf{S}$ is defined in (1.3d). Across the interface the following jump conditions occur

$$[\mathbf{S}(\mathbf{u}(X,t),p(X,t))\mathbf{n}] = \boldsymbol{\sigma}(X,t), \quad \text{if } X\in\Gamma(t), \tag{5.2d}$$

$$[\mathbf{u}(X,t)] = \mathbf{0}, \qquad \text{if } X\in\Gamma(t), \tag{5.2e}$$

where $\Gamma(t)$ is the interface that separates the subdomains $\Omega^+(t)$ and $\Omega^-(t)$ illustrated in Figure 5.1 and $\mathbf{n}$ is the unit normal vector to the interface. We omit the dependent variable t and denote the domains $\Omega^\pm(t)$ by $\Omega^\pm$.

5.2 IFE implementation

In this section, we present the algorithm used to solve the Stokes interface problem with moving interfaces. This includes determining the surface force, solving the Stokes interface

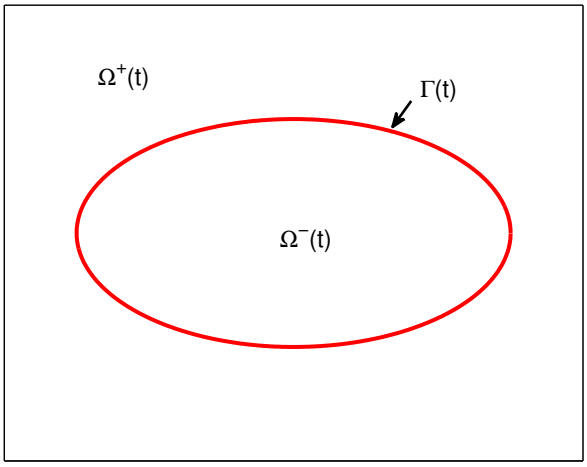


Figure 5.1: A rectangular domain Ω with an immersed interface $\Gamma(t)$.

problem, tracking the interface and updating the type of the elements. We note that since the problem is strongly dependent on time (*i.e.* the interface evolves in time), the IFE space and IFE particular functions evolve in time. Therefore, the tasks described in Sections 5.2.1, 5.2.2 and 5.2.3 must be carried out at every time step with the corresponding IFE space and particular functions. We note that an interface tracking procedure cannot be implemented to track all the points on the interface. Instead, we assume that at the initial time t_0, the interface $\Gamma(t)$ is sampled uniformly using a set of N control points. These control points, which we shall denote $\mathcal{X}(t) = (\mathcal{X}^{(1)}(t), \mathcal{X}^{(2)}(t), ..., \mathcal{X}^{(N)}(t))$, are ordered in space and evolve in time.

5.2.1 Even-odd rule

To implement the IFE method, it is very important to determine whether a point in the domain Ω lies inside Ω^+ or Ω^-. To this end, we introduce the polygon $\tilde{\Gamma}(t)$ obtained using a piecewise linear interpolation of the control points $\mathcal{X}(t)$ and assume that the approximated interface $\tilde{\Gamma}(t)$ is a closed polygon and apply the so-called even-odd algorithm described in [124]. The main idea of this algorithm is to count how many times a horizontal ray passing through the point of interest cuts the closed polygon. If the number of intersections is odd then the point lies inside the polygon, otherwise it lies outside the polygon as illustrated in Figure 5.2. This algorithm allows us to obtain information regarding the interface elements and edges such as their position with respect to the interface and their intersections with the interface.

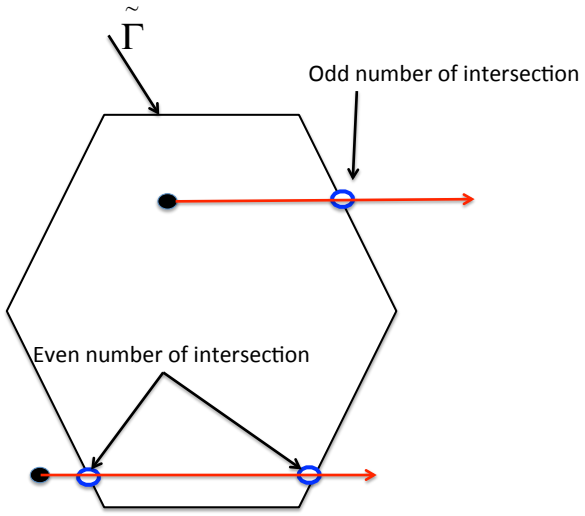


Figure 5.2: An illustration of the Even-Odd algorithm.

5.2.2 Surface force

For a point $(x, y) \in \tilde{\Gamma}(t)$, the surface force is usually given by the expression [90]

$$\boldsymbol{\sigma}(x, y, t) = -\sigma\kappa(x, y, t)\mathbf{n}(x, y, t), \quad t > 0, \tag{5.3}$$

where $\sigma \geq 0$ is the surface tension, $\kappa(x, y, t)$ is the curvature of the interface at the point (x, y) and $\mathbf{n}(x, y, t) = (n_x, n_y)^T$ is the unit vector normal to the interface at (x, y) pointing towards the exterior of the region enclosed by the interface. In this work, we only need to approximate the curvature κ at $(x, y) \in \tilde{\Gamma}(t)$, which will be done using a piecewise quadratic parametric interpolation at three adjacent control points that we denote $\mathcal{X}^{(i-1)}(t)$, $\mathcal{X}^{(i)}(t)$ and $\mathcal{X}^{(i+1)}(t)$, such that $\mathcal{X}^{(i)}(t)$ is the closest to the point $(x, y) \in \tilde{\Gamma}(t)$ as shown in Figure 5.3. We denote the interpolant by $\mathbf{r}(\tau) = (r_x(\tau), r_y(\tau))^T$, and use the following formula to determine the curvature:

$$\kappa(\tau) = \left| \frac{d\vec{T}}{ds} \right|, \tag{5.4}$$

where $\vec{T}$ is the unit tangent vector to $\mathbf{r}(\tau)$ and s is the arc length.

In this work, we assume that the control points are ordered counter clockwise and therefore we use the following formula to determine the unit normal vector $\mathbf{n}(x, y)$:

$$a = \left| r_y'(\tau) \right|, \quad b = \left| r_x'(\tau) \right|,$$

$$\mathbf{c} = (c_1, c_2, c_3)^T = (x_2 - x_1,\ y_2 - y_1,\ 0)^T \times (0, 0, 1)^T,$$

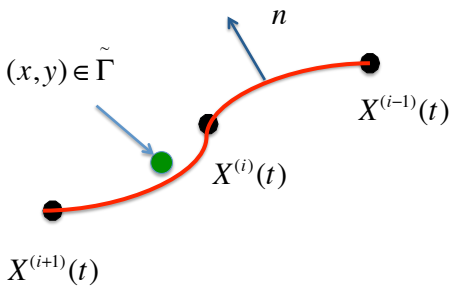


Figure 5.3: An illustration of the quadratic parametric interpolation used to determine the curvature of the interface and the normal vector to the interface at (x, y).

$$n_x = \frac{\text{sign}(c_1)a}{\sqrt{a^2+b^2}}, \quad n_y = \frac{\text{sign}(c_2)b}{\sqrt{a^2+b^2}},$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the control points $\mathcal{X}^{(i-1)}$ and $\mathcal{X}^{(i)}$, respectively.

5.2.3 Immersed finite element method

In this section, we derive the finite element formulation used to solve the Stokes interface problem with a moving interface. Note that due to the motion of the interface, the IFE space and particular functions evolve in time. Thus, we define the spaces

$$\mathcal{H}^m(\Omega, t) = \{(\mathbf{u}, p) : \mathbf{u} \in C^0(\Omega), \mathbf{u} \in (H^m(A))^2 \text{ and } p \in H^1(A), \quad \forall A \subset \Omega^{\pm}\},$$

where H^m is the standard Sobolev space and introduce

$$\mathcal{H}_0^m(\Omega, t) = \{(\mathbf{u}, p) \in \mathcal{H}^m(\Omega, t) : \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega\}.$$

Following the standard procedure, see for instance [120], we multiply the system (5.2a-5.2b) by $(\mathbf{v}, q) \in \mathcal{H}_0^1(\Omega, t)$, at a given time t, and integrate over an arbitrary non-interface element $T \in \mathcal{T}_h$ to obtain

$$-\int_T (\nabla \cdot \mathbf{S}(\mathbf{u}, p)) \cdot \mathbf{v} dx - \int_T \nabla \cdot \mathbf{u} q dx = \mathbf{0}. \tag{5.5}$$

We apply the divergence theorem to integrate the first term by parts and obtain

$$-\int_{\partial T} (\mathbf{S}(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds + \int_T \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} dx - \int_T \nabla \cdot \mathbf{u} q dx = \mathbf{0}, \tag{5.6}$$

where $\mathbf{A} : \mathbf{B} = \sum_{i=1}^2 \sum_{j=1}^2 A_{ij} B_{ij}$.

On an interface element T, we apply the divergence theorem on $T\cap\Omega^+$ and $T\cap\Omega^-$ and take into account that the exact solution satisfies the jump conditions (5.2d) and (5.2e) across the interface to obtain

$$-\int_{\partial T}(\mathbf{S}(\mathbf{u},p)\mathbf{n})\cdot\mathbf{v}ds+\int_T\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx-\int_T\nabla\cdot\mathbf{u}qdx = \int_{T\cap\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds, \tag{5.7}$$

Summing over all elements leads to

$$-\sum_{T\in\mathcal{T}_h}\int_{\partial T}(\mathbf{S}(\mathbf{u},p)\mathbf{n})\cdot\mathbf{v}ds+\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx-\int_\Omega\nabla\cdot\mathbf{u}qdx = \int_{\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds. \tag{5.8}$$

We assume $(\mathbf{u},p)\in\mathcal{H}^2(\Omega,t)$, hence

$$-\sum_{e\in\mathcal{E}_h}\int_e[(\mathbf{S}(\mathbf{u},p)\mathbf{n})\cdot\mathbf{v}]ds+\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx-\int_\Omega\nabla\cdot\mathbf{u}qdx = \int_{\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds. \tag{5.9}$$

Applying the identity $ab-cd=\frac{1}{2}(a+c)(b-d)+\frac{1}{2}(a-c)(b+d)$ yields

$$-\sum_{e\in\mathcal{E}_h}\int_e\left([(\mathbf{S}(\mathbf{u},p)\mathbf{n})]\cdot\{\mathbf{v}\}+\{(\mathbf{S}(\mathbf{u},p)\mathbf{n})\}\cdot[\mathbf{v}]\right)ds+\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx - \int_\Omega\nabla\cdot\mathbf{u}qdx=\int_{\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds. \tag{5.10}$$

Every interface edge can be expressed as $e=e^+\cup e^-$, where $e^\pm=e\cap\Omega^\pm$. Hence

$$\int_e[\mathbf{S}(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}ds=\int_{e^+}[\mathbf{S}(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}ds+\int_{e^-}[\mathbf{S}(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}ds.$$

Since $(\mathbf{u},p)\in\mathcal{H}^2(\Omega,t)$, we write

$$\int_e[\mathbf{S}(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}ds=0, \tag{5.11}$$

which also holds for non-interface elements.

Combining (5.10) and (5.11) with $(\mathbf{u},p)\in\mathcal{H}^2(\Omega,t)$ and assuming $\mathbf{u}$ continuous lead to the interior penalty weak formulation

$$\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx-\int_\Omega\nabla\cdot\mathbf{u}qdx-\sum_{e\in\mathcal{E}_h}\int_e\{(\mathbf{S}(\mathbf{u},p)\mathbf{n})\}\cdot[\mathbf{v}]ds + \sum_{e\in\mathcal{E}_h}\frac{\alpha}{h_e}\int_e\nu[\mathbf{u}]\cdot[\mathbf{v}]ds+\gamma\sum_{e\in\mathcal{E}_h}\int_e\{(\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n})\}\cdot[\mathbf{u}]ds+\sum_{e\in\mathcal{E}_h}\int_e\{q\}[\mathbf{u}]\cdot\mathbf{n}= \int_{\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds+\gamma\sum_{e\subset\partial\Omega}\int_e\{(\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n})\}\cdot[\mathbf{g}]ds+\sum_{e\subset\partial\Omega}\int_e\{q\}[\mathbf{g}]\cdot\mathbf{n}+\sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}]ds, \tag{5.12}$$

where α is a positive stabilization parameter and $\gamma = 1$ for the nonsymmetric weak formulation (NIPG) while $\gamma = -1$ for the symmetric weak formulation (SIPG).

The weak form above leads to the interior penalty formulation for the Stokes problem consisting of finding $(\mathbf{u}(x,y,t), p(x,y,t)) \in \mathcal{H}^1(\Omega,t)$ such that $\mathbf{u} = \mathbf{g}$ on $\partial\Omega$

$$\begin{cases} A(\mathbf{u},\mathbf{v}) + B(\mathbf{v},p) & = L(\mathbf{v}) \\ B(\mathbf{u},q) & = l(q) \end{cases}, \qquad \forall\ (\mathbf{v},q) \in \mathcal{H}_0^1(\Omega,t), \tag{5.13a}$$

where

$$A(\mathbf{w},\mathbf{v}) =$$
$$\int_\Omega \nu\boldsymbol{\epsilon}(\mathbf{w}) : \nabla\mathbf{v}dx - \sum_{e\in\mathcal{E}_h}\int_e \nu\{\boldsymbol{\epsilon}(\mathbf{w})\mathbf{n}\}\cdot[\mathbf{v}]ds + \gamma\sum_{e\in\mathcal{E}_h}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\}\cdot[\mathbf{w}]$$
$$+\sum_{e\in\mathcal{E}_h}\frac{\alpha}{h_e}\int_e \nu[\mathbf{u}]\cdot[\mathbf{v}]ds, \tag{5.13b}$$

$$B(\mathbf{v},q) = -\int_\Omega q\nabla\cdot\mathbf{v}dx + \sum_{e\in\mathcal{E}_h}\int_e\{q\}[\mathbf{v}]\cdot\mathbf{n}ds, \tag{5.13c}$$

$$L(\mathbf{v}) = \int_{\Gamma(t)}\boldsymbol{\sigma}\cdot\mathbf{v}ds + \gamma\sum_{e\subset\partial\Omega}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\}\cdot[\mathbf{g}]ds + \sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}]ds, \tag{5.13d}$$

$$l(q) = \sum_{e\subset\partial\Omega}\int_e\{q\}[\mathbf{g}]\cdot\mathbf{n}ds. \tag{5.13e}$$

Then, this formulation leads to the DG-IFE method consisting of finding $(\mathbf{u}_h, p_h) \in S_h(\Omega,t)\oplus\{\mathbf{q}_h(x,y,t)\}$ such that

$$\begin{cases} A(\mathbf{u}_h,\mathbf{v}_h) + B(\mathbf{v}_h,p_h) & = L_h(\mathbf{v}_h) \\ B(\mathbf{u}_h,q_h) & = l(q_h) \end{cases}, \qquad \forall\ (\mathbf{v}_h,q_h) \in S_{h,0}(\Omega,t), \tag{5.14}$$

subject to the boundary conditions:

$$\mathbf{u}_h(R) = \mathbf{g}(R), \quad \text{for all mesh vertices } R \in \partial\Omega,$$

where

$$L_h(\mathbf{v}_h) = \sum_{T\in\mathcal{T}_h^i}\int_{\overline{DE}}\boldsymbol{\sigma}_h\cdot\mathbf{v}_h ds + \gamma\sum_{e\subset\partial\Omega}\int_e\{\nu\boldsymbol{\epsilon}(\mathbf{v}_h)\mathbf{n}\}\cdot[\mathbf{g}]ds + \sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}_h]ds, \tag{5.15}$$

$\boldsymbol{\sigma}_h$ is the linear interpolant of $\boldsymbol{\sigma}$ on $\overline{DE}$ using the value of $\boldsymbol{\sigma}$ at D and E.

Here $S_h(\Omega, t)$ is the IFE space $S_h(\Omega)$ defined in (3.9) and constructed using the approximated interface $\tilde{\Gamma}(t)$; $\mathbf{q}_h(x, y, t)$ is the IFE vector function $\mathbf{q}_h(x, y)$ defined in (3.24) and constructed using the approximated interface $\tilde{\Gamma}(t)$ and

$$S_{h,0}(\Omega, t) = \{(\mathbf{v}, p) \in S_h(\Omega, t) : \ \mathbf{v}(R) = \mathbf{0}, \ \text{ for all mesh vertices } R \in \partial\Omega\}.$$

Since the interface is evolving in time and is driven by the velocity of the fluids, we propose to track the interface using the approximate velocity $\mathbf{u}_h$, the control points $\mathcal{X}^{(i)}(t)$, $1 \leq i \leq N$, and the ODEs (5.1). The tracking procedure can therefore be written as

$$\begin{cases} \frac{d}{dt}\mathcal{X}^{(i)}(t) & = \mathbf{u}_h(\mathcal{X}^{(i)}(t), t), \quad t \in [0, T] \\ \mathcal{X}^{(i)}(0) & = \mathcal{X}_0^{(i)} \end{cases}, \quad i = 1, 2, ..., N, \tag{5.16}$$

where $\mathbf{u}_h(\mathcal{X}^{(i)}(t), t)$ is the IFE velocity of the fluid at the control point $\mathcal{X}^{(i)}(t)$ and N denotes the total number of control points.

To solve the Stokes problem with a moving interface, we propose to use the following algorithm:

Select a small $\epsilon > 0$, $\Delta t > 0$ and the number of control points N

Determine the initial control points $\mathcal{X}(0)$

Define the partition $t_k = k\Delta t, \quad k = 0, 1, 2, 3, ...$

Construct an initial uniform mesh for Ω.

For $k = 0, 1, 2, ...$

1. *If needed, refine the initial mesh near the interface.*

2. *Use the given control points $\mathcal{X}(t_k)$ to construct the IFE space and particular functions and compute the surface force $\boldsymbol{\sigma}$ at the interface points.*

3. *Solve Stokes problem and determine the velocity $\mathbf{u}_h(\mathcal{X}^{(i)}(t_k), t_k)$ at the control points $\mathcal{X}^{(i)}(t_k)$, $i = 1, 2, ..., N$.*

4. *Stop when $\max_{1 \leq i \leq N} ||\mathbf{u}_h(\mathcal{X}^{(i)}(t_k), t_k) - \mathbf{u}_h(\mathcal{X}^{(i)}(t_{k-1}), t_{k-1})||_2 < \epsilon$.*

5. *Move the control points $\mathcal{X}(t_k)$ according to the ODEs (5.16).*

A flowchart of this algorithm is shown in Figure 5.4.

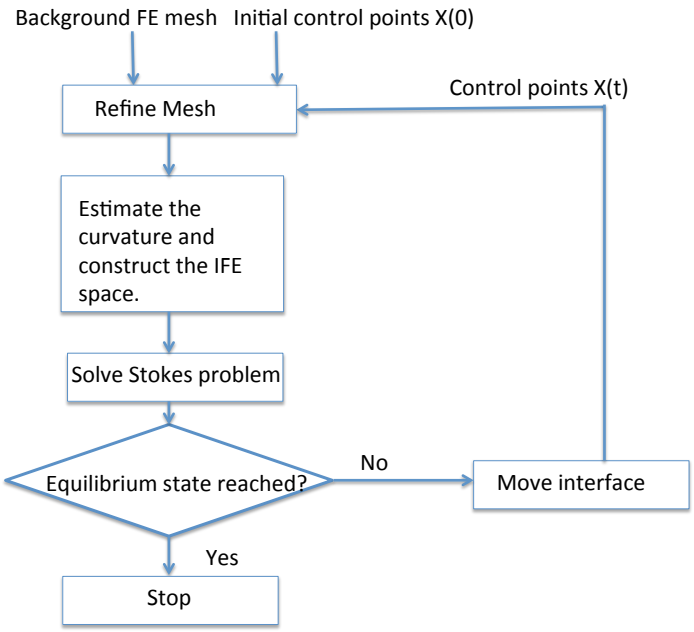


Figure 5.4: Flowchart of an algorithm used to solve the Stokes interface problem.

5.3 Numerical simulations

In this section, we consider two examples. The first is the retraction of drops and the second is the deformation of the drop in shear flows. To solve these two problems, a good approximation of the velocity is needed. Since we are using a DG method, the local refinement of the mesh around the interface is possible yielding hanging nodes. The procedure to create this mesh is simpler to implement than the usual mesh refinement procedures that do not allow hanging nodes.

To illustrate the efficiency of the adaptive refinement we solve Example 4.2.3 using uniform meshes having 5^2, 10^2, 20^2, 40^2, 80^2 and 160^2 elements and use the SIPG method. We also solve the same problem starting with a coarse mesh having 5^2 elements and then we refine the mesh by dividing the elements cut by the interface into four congruent elements. The refinement is carried out eight times and the SIPG method is used. The maximum errors of the velocity at the control points $\mathcal{X}(t)$ are computed for each mesh and plotted against the corresponding degrees of freedom as shown in Figure 5.5. We can see that the error converges faster when the adaptive refinement procedure is used. To take advantage of this feature, we create an initial background mesh and then refine all the elements cut by the interface. The level of refinement shall refer to the depth of refinement of the mesh around the interface. This local refinement procedure is very efficient as it allows the use of

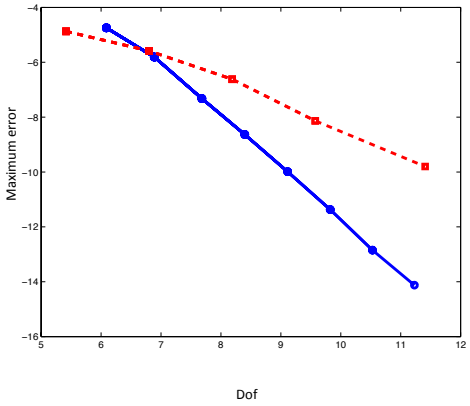


Figure 5.5: Finite element errors versus the degrees of freedom in the *log-log* scale for adaptive (-o) and uniform (- -□) refinements.

structured meshes and uses fewer degrees of freedom than the uniform refinement procedure.

Example 5.3.1. *Retraction of drops*

It is well known that if an elastic drop is stretched out, it will retract to its equilibrium state if no body forces are exerted upon the flow [131]. In this experiment, we consider the square domain $[-1,1]^2$ and a stretched drop modeled using an ellipse centered at the origin with length of major axis $a = \sqrt{3}/3.1$ and length of minor axis $b = \sqrt{3}/5.3$. The viscosity of the fluid inside the drop is $\nu^- = 0.6$ and outside the drop $\nu^+ = 2$. We use a mesh with approximately $3,000$ elements, obtained by partitioning the domain using a 20×20 background mesh and using four levels of refinement as shown in Figure 5.6. We use homogeneous boundary conditions and no body forces so that the motion of the interface is driven by its surface force.

To track the interface in time, we solve the ODE (5.16) using the forward Euler method:

$$\mathcal{X}_{k+1}^{(i)} = \mathcal{X}_k^{(i)} + \Delta t \mathbf{u}_h(\mathcal{X}_k^{(i)}, t_k), \quad i = 1, 2, ..., N, \tag{5.17}$$

where $\mathcal{X}_k^{(i)}$, $i = 1, 2, ..., , N$ are the approximate locations of the control points at time t_k.

Note that the Stokes interface problem is solved using the SIPG method at every time step t_k in order to obtain the approximate velocity $\mathbf{u}_h(\mathcal{X}_k^{(i)}, t_k)$, $i = 1, 2, ..., N$ at the control points. In this simulation, we use a time step $\Delta t = 10^{-2}$, homogenous boundary conditions and as mentioned above we assume that there is no body forces exerted on the system.

We show the DG-IFE pressure and the mesh used to partition the domain at $t = 0, 0.7, 4.27$ in Figure 5.7. The equilibrium state reached by the drop indicates that the drop has retracted

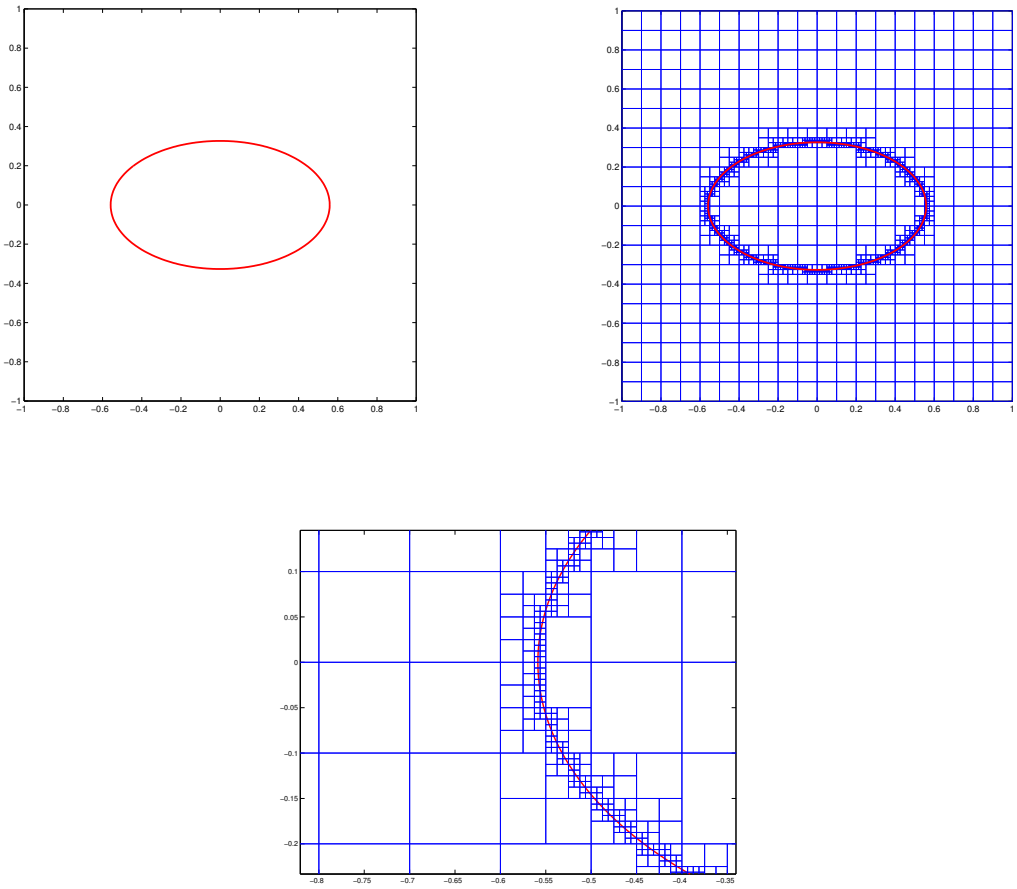
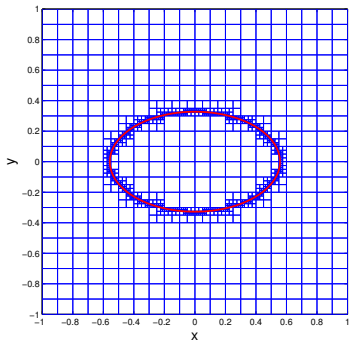
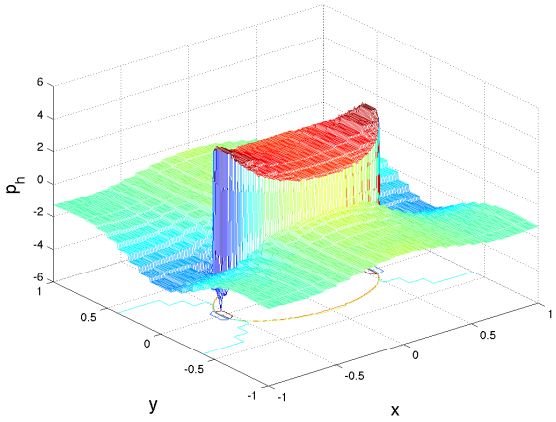


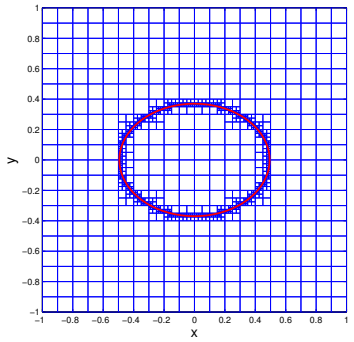
Figure 5.6: Initial state of the system for Example 5.3.1 (upper left), the mesh used to discretize Stokes problem (5.2a)−(5.2c) at $t = 0$ (upper right) and a blow-up of the mesh near the interface (bottom).



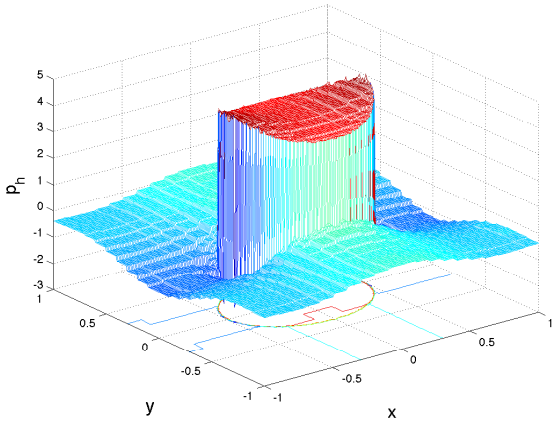
(a) The interface and a mesh at $t = 0$



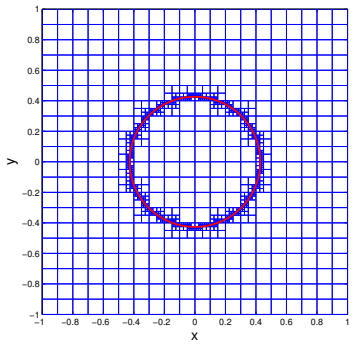
(b) DG-IFE pressure at $t = 0$



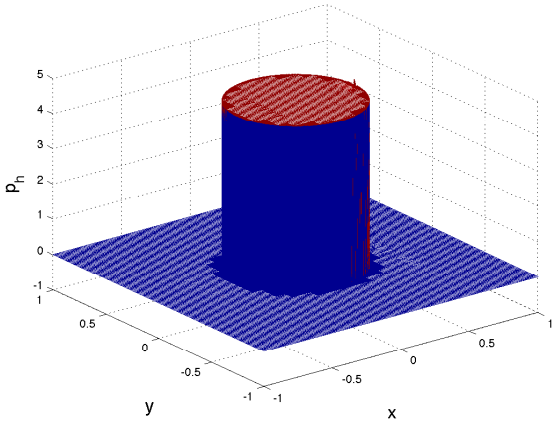
(c) The interface and a mesh at $t = 0.7$



(d) DG-IFE pressure at $t = 0.7$



(e) The interface and a mesh at $t = 4.27$



(f) DG-IFE pressure at $t = 4.27$ (steady state)

Figure 5.7: Simulation of Example 5.3.1.

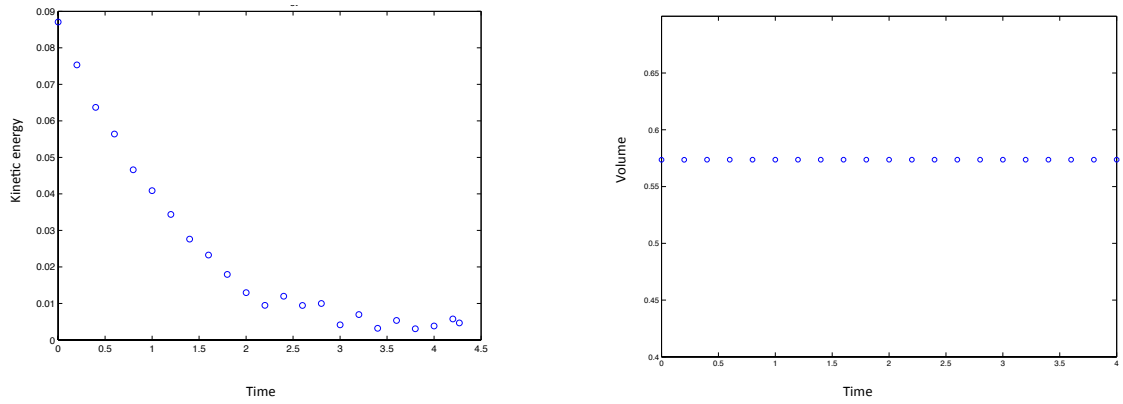


Figure 5.8: Kinetic energy of the system versus time (left). Volume of the drop versus time (right).

to its natural spherical state after 427 iterations. Our stoping criteria is the convergence of the kinetic energy $E(t_k) = ||\mathbf{u}_h(\mathcal{X}_k^{(i)}, t_k)||_{0,\Omega}$ to zero (*i.e.,* $E(t_k) < \epsilon$) as shown in Figure 5.8. Also, since we are dealing with incompressible flows, the volume of the drop should remain unchanged with respect to time. This has been validated in Figure 5.8 which shows the volume inside the drop versus time. The steady state pressure shown in Figure 5.7(f) satisfies the Young-Laplace equation

$$[p]|_\Gamma = \frac{\sigma}{R}, \tag{5.18}$$

where R is the radius of the drop. The surface tension $\sigma = 2$ is used in this experiment, and the radius of the steady state drop is $R \approx 0.4273$. According to (5.18), the jump of the pressure across the interface should be $[p]|_\Gamma \approx 4.6805$. By examining Figure 5.7, we can see that the pressure on each subdomain is constant and the jump of the DG-IFE pressure across the interface is $[p]|_\Gamma \approx 4.68$ which agrees with the expected value.

Example 5.3.2. *Drop deformation in shear flow*

We validate our DG-IFE method by studying the behavior of a drop in shear flow. The domain of interest consists of a drop with initial radius $a = \frac{1}{4}$ centered at the origin, immersed in a rectangle $\Omega = [-\pi, \pi] \times [-1, 1]$ and subjected to a shear flow as shown in Figure 5.9. The boundary conditions are defined as $\mathbf{u} = \mathbf{g} = (\epsilon y, 0)^T$ on $\partial\Omega$ such that ϵ is the shear rate. We define the capillary parameter $Ca = \frac{\nu a \epsilon}{\sigma}$ where ν is the viscosity of the drop and σ is the surface tension. The deformation of the drop depends strongly on Ca and was extensively studied in [135]. The deformation is computed using the following formula:

$$D = \frac{L - W}{L + W}, \tag{5.19}$$

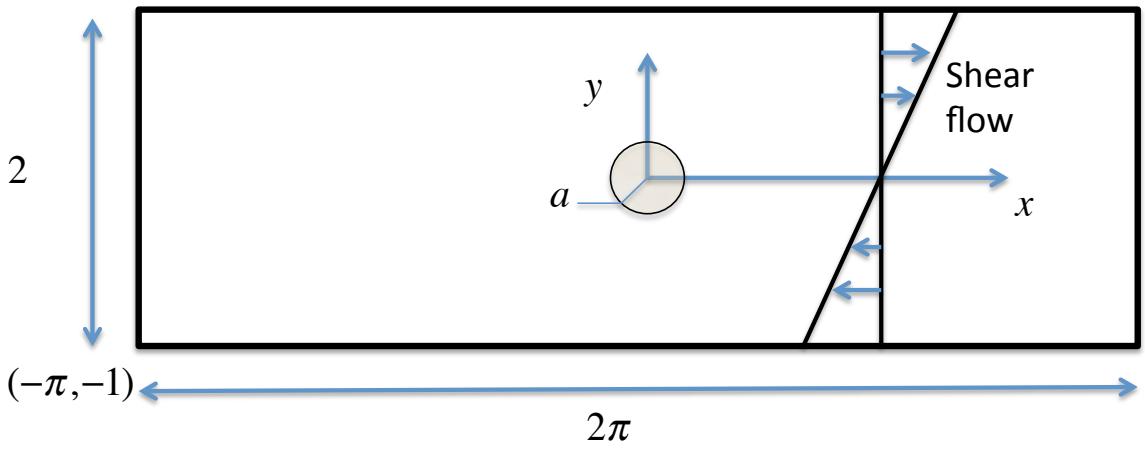


Figure 5.9: The computational domain for drop deformation in shear flow.

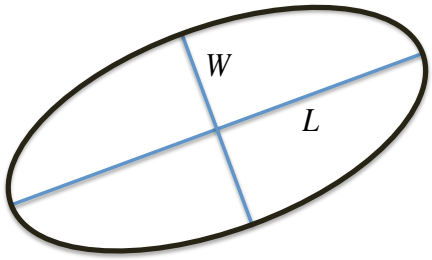


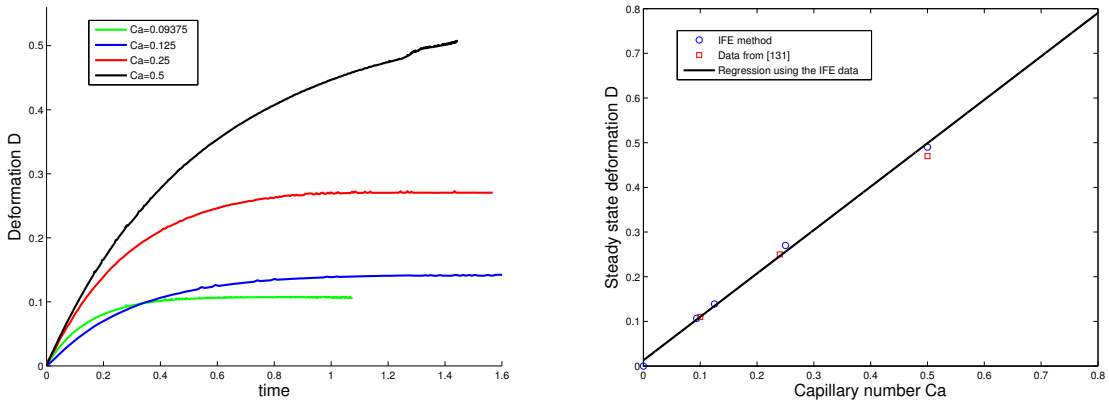
Figure 5.10: The length L and width W of a deformed drop.

where L is the length of the deformed drop and W is its width as illustrated in Figure 5.10.

We solve the ODE system (5.16) using the following adaptive Runge-Kutta method [107]:

$$
\begin{cases}
\textit{Set up the initial time } t_0 = 0, \textit{ initial time step } \Delta t, \ \epsilon_1 \textit{ and } \epsilon_{\max}: \\
\textit{For k=0,1,2,...} \\
\quad \textit{1. Compute the velocity at different stages:} \\
\qquad K_1^{(i)} = \Delta t \mathbf{u}_h\left(\mathcal{X}_k^{(i)}, t_k\right), \\
\qquad K_2^{(i)} = \Delta t \mathbf{u}_h\left(\mathcal{X}_k^{(i)} + \frac{2K_1^{(i)}}{3}, t_k + \frac{2}{3}\Delta t\right), \\
\qquad K_3^{(i)} = \Delta t \mathbf{u}_h\left(\mathcal{X}_k^{(i)} + \frac{2K_2^{(i)}}{3}, t_k + \frac{2}{3}\Delta t\right), \qquad i = 1, 2, ...N. \\
\quad \textit{2. Approximate the location of the control points using methods of order 2 and 3 respectively:} \\
\qquad \hat{X}^{(i)} = \mathcal{X}_k^{(i)} + \frac{K_1^{(i)}}{4} + \frac{3K_2^{(i)}}{4}, \\
\qquad \tilde{X}^{(i)} = \mathcal{X}_k^{(i)} + \frac{K_1^{(i)}}{4} + \frac{3K_2^{(i)}}{8} + \frac{3K_3^{(i)}}{8}, \quad i = 1, 2, ...N. \\
\quad \textit{3.Update the time step:} \\
\qquad \epsilon = \max_{1 \le i \le N} ||\tilde{X}^{(i)} - \hat{X}^{(i)}||_2, \\
\qquad \Delta t = 0.9 \left(\frac{\epsilon_{\max}}{\epsilon}\right)^{1/4} \Delta t, \\
\qquad\quad \textit{if } \epsilon < \epsilon_{\max}: \\
\qquad\qquad \mathcal{X}_{k+1} = \hat{X}, \\
\qquad\qquad \textit{Stop when } \max_{1 \le i \le N} ||\mathbf{u}_h(\mathcal{X}^{(i)}(t_k), t_k) - \mathbf{u}_h(\mathcal{X}^{(i)}(t_{k-1}), t_{k-1})||_2 < \epsilon_1. \\
\qquad\qquad t_{k+1} = t_k + \Delta t, \\
\qquad\quad \textit{else:} \\
\qquad\qquad \textit{Repeat (1).}
\end{cases}
\tag{5.20}
$$

We solve problem 5.3.2 using a structured mesh having at most 3500 elements. The viscosities are $\nu^- = \nu^+ = 1$, and the simulations are carried out using SIPG method, the ODE solver (5.20) and different capillary numbers $Ca = \frac{3}{32}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$. We plot the evolution of the deformation D defined in (5.19) versus time for different values of Ca in Figure 5.11(a). Then we plot the steady state deformation D versus the capillary number in Figure 5.11(b), and compare our results to their counterparts obtained in [131]. Figure 5.11(b) shows excellent agreement between the results obtained in this work and the results obtained in [131]. In fact, the maximum difference in the steady deformation between our results and the ones obtained in [131] is about 0.0087. However, the simulations in [131] were carried out using a much finer mesh having 2048×1024 elements, while the simulations in this work were carried out with at most 3500 elements. This shows that this method solves smaller linear systems and can capture the physics of the problem.



(a) The drop deformation D versus time. (b) Steady state deformation D versus Ca.

Figure 5.11: Simulation of Example 5.3.2.

Chapter 6

Axisymmetric Stokes interface problem

In this chapter, we treat the three-dimensional Stokes interface problem with velocity $\underline{\mathbf{u}} = (u_1, u_2, u_3)^T$, $u_i : \mathbb{R}^3 \to \mathbb{R}$, $i = 1, 2, 3$, and the pressure $p : \mathbb{R}^3 \to \mathbb{R}$ for incompressible flow. In this chapter, we use $\underline{\mathbf{u}}$ to describe a three-dimensional vector and use $\mathbf{u}$ to denote a two-dimensional vector as before. The problem is given by the following set of equations:

$$-\nabla \cdot \mathbf{L}(\underline{\mathbf{u}}, p) = \underline{\mathbf{f}}, \text{ in } V^+ \cup V^-, \tag{6.1a}$$

$$\nabla \cdot \underline{\mathbf{u}} = 0, \text{ in } V^+ \cup V^-, \tag{6.1b}$$

$$\underline{\mathbf{u}} = \underline{\mathbf{g}}, \text{ on } \partial V, \tag{6.1c}$$

where the stress tensor is defined as follows

$$\mathbf{L}(\underline{\mathbf{u}}, p) = \nu(\nabla \underline{\mathbf{u}} + (\nabla \underline{\mathbf{u}})^T) - p\mathbf{I}_3,$$

$\underline{\mathbf{f}}(x, y, z) : \mathbb{R}^3 \to \mathbb{R}^3$ is a body force in $L^2(V)$ and $\mathbf{I}_3$ is the 3×3 identity matrix.

The domain $V = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 \leq 1, -1 \leq z \leq 1\}$ is a three-dimensional cylinder and ∂V denotes its boundary. We assume that the domain is separated by an interface Ξ into two subdomains V^+ and V^-, such that each domain contains one fluid, and we assume that the interface is defined by $\Xi = \{(x, y, z) \in V : G(x, y, z) = 0\}$, where $G : \mathbb{R}^3 \to \mathbb{R}$ is a smooth function. Across the interface Ξ, the following jump conditions occur:

$$[\underline{\mathbf{u}}]|_\Xi = \mathbf{0}, \tag{6.1d}$$

$$[\mathbf{L}(\underline{\mathbf{u}}, p)\underline{\mathbf{n}}]|_\Xi = \underline{\boldsymbol{\sigma}}, \tag{6.1e}$$

where $\underline{\boldsymbol{\sigma}} = (\sigma_1, \sigma_2, \sigma_3)^T$ is the surface force and $\underline{\mathbf{n}}$ is a unit vector normal to the interface.

In cylindrical coordinates (r, θ, z), the Stokes interface problem for the velocity $\underline{\mathbf{u}} = (u_r, u_\theta, u_z)^T$ and the pressure p is defined as follows:

$$-\nabla \cdot \mathbf{L}(\underline{\mathbf{u}}(r,\theta,z), p(r,\theta,z)) = \underline{\mathbf{f}}(r,\theta,z), \text{ in } V^+ \cup V^-, \tag{6.2a}$$
$$\nabla \cdot \underline{\mathbf{u}}(r,\theta,z) = 0, \quad \text{in } V^+ \cup V^-, \tag{6.2b}$$
$$\underline{\mathbf{u}}(r,\theta,z) = \underline{\mathbf{g}}(r,\theta,z), \text{ on } \partial V, \tag{6.2c}$$

and the jump conditions become

$$[\underline{\mathbf{u}}(r,\theta,z)]|_\Xi = \mathbf{0}, \tag{6.2d}$$
$$[\mathbf{L}(\underline{\mathbf{u}}(r,\theta,z), p(r,\theta,z))\underline{\mathbf{n}}]|_\Xi = \underline{\boldsymbol{\sigma}}. \tag{6.2e}$$

We recall the divergence of a vector $\underline{\mathbf{v}} = (v_r, v_\theta, v_z)^T$ in cylindrical coordinates:

$$\nabla \cdot \underline{\mathbf{v}} = \frac{1}{r}\frac{\partial r v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z} \tag{6.2f}$$

and the velocity gradient

$$\nabla \underline{\mathbf{v}} = \begin{pmatrix} \frac{\partial v_r}{\partial r} & \frac{1}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r} & \frac{\partial v_r}{\partial z} \\ \frac{\partial v_\theta}{\partial r} & \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} & \frac{\partial v_\theta}{\partial z} \\ \frac{\partial v_z}{\partial r} & \frac{1}{r}\frac{\partial v_z}{\partial \theta} & \frac{\partial v_z}{\partial z} \end{pmatrix}. \tag{6.2g}$$

In this work, we assume axisymmetry around the z-axis, then:
• we project the three-dimensional domain V onto the meridian plane $\Omega = \{(r,z) : 0 \le r \le 1,\ -1 \le z \le 1\}$.
• we project the interface Ξ onto the meridian plane Ω and denote it as $\Gamma = \{(r,z) : \tilde{G}(r,z) = G(r,0,z) = 0\}$.
• we reduce the three-dimensional velocity $\underline{\mathbf{u}}$ to a two-dimensional velocity $\mathbf{u} = (u_r, u_z)^T$ defined on the meridian plane Ω.
• we reduce the three-dimensional stress tensor $\mathbf{L}(\underline{\mathbf{u}}(r,\theta,z), p(r,\theta,z))$ to an axisymmetric stress tensor $\mathbf{S}(\mathbf{u}(r,z), p(r,z))$ defined on the meridian plane Ω.

We note that the projected interface Γ separates the meridian plane Ω into two sub-domains Ω^- and Ω^+ as shown in Figure 6.1. Then the Stokes interface problem can be reduced in the meridian plane Ω as follows: Find $\mathbf{u} = (u_r, u_z)^T$ and p that satisfy

$$-\nabla \cdot \mathbf{S}(\mathbf{u}(r,z), p(r,z)) = \underline{\mathbf{f}}(r,z), \text{ in } \Omega^+ \cup \Omega^-, \tag{6.3a}$$
$$\nabla \cdot \mathbf{u}(r,z) = 0, \quad \text{in } \Omega^+ \cup \Omega^-, \tag{6.3b}$$
$$\mathbf{u}(r,z) = \mathbf{g}(r,z), \text{ on } \partial\Omega, \tag{6.3c}$$

where $\underline{\mathbf{f}} = (f_r, 0, f_z)^T$ is a body force and $\mathbf{g} = (g_r, g_z)^T$, and the jump conditions across the projected interface Γ become

$$[\mathbf{u}(r,z)]|_\Gamma = \mathbf{0}, \tag{6.3d}$$

$$[\mathbf{S}_1(\mathbf{u}(r,z), p(r,z))\mathbf{n}]|_\Gamma = \boldsymbol{\sigma}. \tag{6.3e}$$

Here, $\mathbf{n} = (n_r, n_z)^T$ is the unit vector normal to the interface Γ, $\boldsymbol{\sigma} = (\sigma_r, \sigma_z)^T$ is the surface force and $\mathbf{S}$ is the projected stress tensor onto the meridian plane Ω and is defined as

$$\mathbf{S}(\mathbf{u}, p) = \nu \boldsymbol{\epsilon}_s(\mathbf{u}) - p\mathbf{I}_3, \tag{6.3f}$$

with

$$\boldsymbol{\epsilon}_s(\mathbf{u}) = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \tag{6.3g}$$

where

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_r}{\partial r} & 0 & \frac{\partial u_r}{\partial z} \\ 0 & \frac{u_r}{r} & 0 \\ \frac{\partial u_z}{\partial r} & 0 & \frac{\partial u_z}{\partial z} \end{pmatrix}. \tag{6.3h}$$

The divergence operator in the axisymmetric coordinates applied to a vector $\mathbf{v} = (v_r, v_z)^T$ is:

$$\nabla \cdot \mathbf{v} = \frac{1}{r}\frac{\partial r v_r}{\partial r} + \frac{\partial v_z}{\partial z}. \tag{6.3i}$$

The stress tensor

$$\mathbf{S}_1(\mathbf{u}, p) = \nu \boldsymbol{\epsilon}_1(\mathbf{u}) - p\mathbf{I}_2, \tag{6.3j}$$

where $\mathbf{I}_2$ is the 2×2 identity matrix and

$$\boldsymbol{\epsilon}_1(\mathbf{u}) = \begin{pmatrix} 2\frac{\partial u_r}{\partial r} & \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \\ \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} & 2\frac{\partial u_z}{\partial z} \end{pmatrix}. \tag{6.3k}$$

It is important to note that the PDEs (6.3a) only yields two PDEs. From hereafter, we denote by $\mathbf{w} = (w_r, w_z)^T$ any two-dimensional vector in Ω, obtained by projecting its three-dimensional counterpart $\underline{\mathbf{w}} = (w_r, 0, w_z)^T$ onto Ω. For instance, the two-dimensional body force $\mathbf{f} = (f_r, f_z)^T$ will refer to the projection of the three-dimensional surface force $\underline{\mathbf{f}} = (f_r, 0, f_z)^T$ onto Ω.

In Section 6.1, the IFE basis functions are derived using the jump conditions (6.3d)-(6.3e). In Section 6.2, we obtain an axisymmetric weak formulation using the three-dimensional model and projecting it onto the meridian plane (r, z).

6.1 IFE spaces and particular functions for the Stokes interface problem

In this section, we develop an immersed finite element space and particular functions for the Stokes interface problem with and without surface force. To this end, we introduce the three-dimensional mesh $\mathcal{T}_h^3$ consisting of cylindrical elements T_{kj}^3, $k = 0, 1, ..., M-1$, $j = 0, 1, ..., N-1$,

$$T_{kj}^3 = \{(r, \theta, z) : 0 \leq \theta < 2\pi, k\Delta r \leq r \leq (k+1)\Delta r,\ j\Delta z \leq z \leq (j+1)\Delta z\}$$

where $\Delta z = \frac{2}{N}$, $\Delta r = \frac{1}{M}$, N is the number of nodes in the z-direction and M is the number of nodes in the r-direction. We shall denote the faces of the elements T^3 as ∂T^3 and their projection onto the meridian plane Ω, T with edges ∂T. The projection of the mesh $\mathcal{T}_h^3$ onto Ω will be denoted $\mathcal{T}_h$, and Γ will refer to the projection of the interface Ξ onto Ω as shown in Figure 6.1. We further classify the elements into two subcategories: (i) the set of all elements cut by the interface, which we shall denote $\mathcal{T}_h^{3,i}$, (ii) the set of all elements not cut by the interface denoted $\mathcal{T}_h^{3,n}$. Next, let $\mathcal{E}_h^3$ denote the set of all element faces in $\mathcal{T}_h^3$ and let $\mathcal{E}_h^{3,i} \subset \mathcal{E}_h^3$ be the set of element faces cut by the interface. We note that the mesh $\mathcal{T}_h^3$ is only used to construct the IFE basis functions and derive the weak formulation and will not be used in any numerical computations.

6.1.1 Q_1/Q_0 IFE shape functions without surface force

We present the procedure to construct the IFE basis functions. A typical cylindrical interface element T^3 is projected onto the meridian plane (r, z), which yields an interface rectangular element $T = \square A_1A_2A_3A_4 \in \mathcal{T}_h^i$ with vertices $A_j = (r_j, z_j)$, $j = 1, 2, 3, 4$. The projected interface Γ intersects two edges at $D = (r_D, z_D)$ and $E = (r_E, z_E)$ as shown in Figure 3.1, and the interface $\widetilde{DE} = \Gamma \cap T$ is approximated by the line segment $\overline{DE}$ which separates T into two polygonal domains T^+ and T^- such that T^+ contains vertices of T that are in Ω^+. Topologically, there are two types of interface elements. Type I interface elements are those with two adjacent edges cut by the interface and Type II interface elements have two opposite edges cut by the interface, see the illustrations in Figure 3.1.

As usual, the construction and analysis of the finite element basis functions are performed on the reference element $\hat{T} = \square\hat{A}_1\hat{A}_2\hat{A}_3\hat{A}_4$ with vertices $\hat{A}_1 = (0, 0)^T$, $\hat{A}_2 = (1, 0)^T$, $\hat{A}_3 = (0, 1)^T$, $\hat{A}_4 = (1, 1)^T$. Let $Y = (r, z)$ and $\hat{Y} = (\hat{r}, \hat{z})$ and let

$$\hat{Y} = F(Y) = MY + B, \tag{6.4}$$

be the standard affine mapping from an arbitrary element T to the reference element $\hat{T}$ such that $\hat{A}_j = F(A_j)$, $j = 1, 2, 3, 4$. We further note that each interface element T of Type I

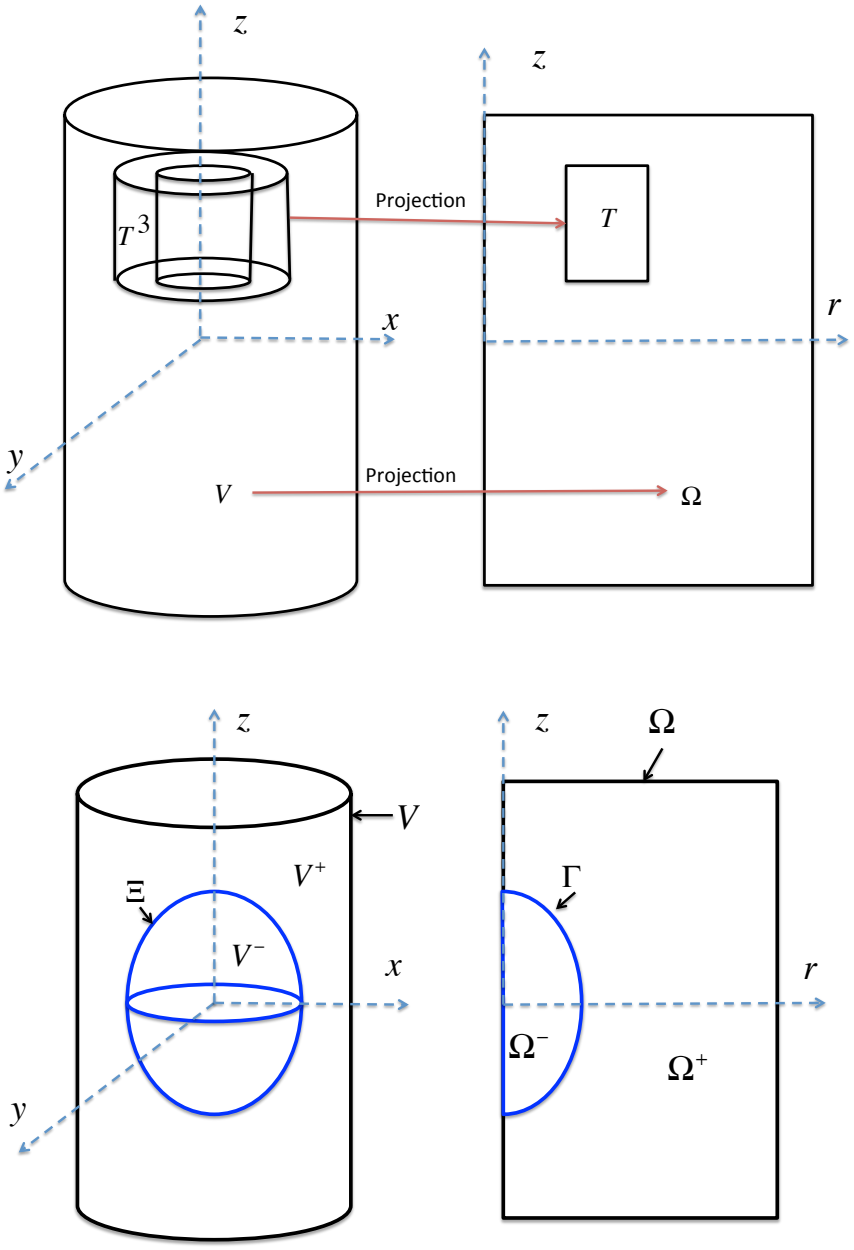


Figure 6.1: The $3D$ domain V and a typical cylindrical element T^3 (upper left), and their projections onto the meridian plane (upper right). The 3D domain and interface (lower left) and their axisymmetric projection (lower right).

(Type II) is mapped into a reference element of Type I (Type II) shown in Figure 3.1 where $\hat{E} = F(E)$ and $\hat{D} = F(D)$ and $\overline{\hat{D}\hat{E}} = F(\overline{DE})$. The interface points $\hat{E}$ and $\hat{D}$ can be written as

$$\hat{D} = \begin{pmatrix} 0 \\ \hat{d} \end{pmatrix}, \quad \hat{E} = \begin{pmatrix} \hat{e} \\ 0 \end{pmatrix}, \tag{6.5}$$

for an element of Type I and

$$\hat{D} = \begin{pmatrix} \hat{d} \\ 1 \end{pmatrix}, \quad \hat{E} = \begin{pmatrix} \hat{e} \\ 0 \end{pmatrix}, \tag{6.6}$$

for an element of Type II, where $0 < \hat{d}, \hat{e} < 1$ as shown in Figure 3.1.

Now we are ready to describe our procedure for constructing the IFE shape functions on the reference element. As usual, a function $\hat{f}(\hat{r}, \hat{z})$ defined for $(\hat{r}, \hat{z}) \in \hat{T}$ leads to a function $f(r, z) = \hat{f}(F^{-1}(\hat{r}, \hat{z}))$ for $(r, z) \in T$ by the affine mapping between the reference element $\hat{T}$ and element T. We first note that the velocity $\mathbf{u} = (u_r, u_z)$ and the pressure p are coupled through the jump condition (6.3e) which requires the design of vector-valued shape functions for both $\mathbf{u}$ and p. This means we plan to approximate the solution vector $\hat{\mathbf{U}} = [\hat{\mathbf{u}}, \hat{p}]^T$ by an IFE function $\hat{\mathbf{\Phi}}$ which is a piecewise polynomial vector function of the form

$$\hat{\mathbf{\Phi}}(\hat{r}, \hat{z}) = \hat{\mathbf{\Phi}}^s(\hat{r}, \hat{z}) = \begin{pmatrix} \hat{\phi}_1^s(\hat{r}, \hat{z}) \\ \hat{\phi}_2^s(\hat{r}, \hat{z}) \\ \hat{\phi}_3^s(\hat{r}, \hat{z}) \end{pmatrix}, \quad \text{for } (\hat{r}, \hat{z}) \in \hat{T}^s, \quad s = +, -, \tag{6.7a}$$

where

$$\hat{\phi}_j^s(\hat{r}, \hat{z}) = a_j^s + b_j^s \hat{r} + c_j^s \hat{z} + d_j^s \hat{r}\hat{z}, \quad j = 1, 2, \quad s = +, -, \tag{6.7b}$$

$$\hat{\phi}_3^s(\hat{r}, \hat{z}) = a_3^s, \quad s = +, -. \tag{6.7c}$$

Next we let $\hat{\mathbf{\Theta}} = (\hat{\phi}_1, \hat{\phi}_2)^T$ and $\hat{\mathbf{\Theta}}|_{T^s}(\hat{r}, \hat{z}) = \hat{\mathbf{\Theta}}^s(\hat{r}, \hat{z}) = \begin{pmatrix} \hat{\phi}_1^s(\hat{r}, \hat{z}) \\ \hat{\phi}_2^s(\hat{r}, \hat{z}) \end{pmatrix}$, $s = +, -$.

We then discuss the construction of IFE shape functions that will be used to form the local IFE space on the reference element $\hat{T}$. According to (6.7) each IFE function is defined by 18 coefficients $a_j^s, b_j^s, c_j^s, d_j^s$, $j = 1, 2$, $s = +, -$ and a_3^s, $s = +, -$. Hence, we can define IFE shape functions $\hat{\mathbf{\Phi}}_i$, $i = 1, 2, \ldots, 9$ whose coefficients are uniquely determined by the following 18 conditions:

- continuity of the velocity component across $\overline{\hat{D}\hat{E}}$ for $\hat{\mathbf{\Theta}}_i = (\hat{\phi}_{1,i}, \hat{\phi}_{2,i})^T$

$$\hat{\mathbf{\Theta}}_i^-(\hat{E}) = \hat{\mathbf{\Theta}}_i^+(\hat{E}), \quad \hat{\mathbf{\Theta}}_i^-(\hat{D}) = \hat{\mathbf{\Theta}}_i^+(\hat{D}), \quad \frac{\partial^2 \hat{\mathbf{\Theta}}_i^-}{\partial \hat{r} \partial \hat{z}} = \frac{\partial^2 \hat{\mathbf{\Theta}}_i^+}{\partial \hat{r} \partial \hat{z}} \tag{6.8a}$$

- weak continuity of the normal stress (6.3e) across $\overline{\hat{D}\hat{E}}$

$$\int_{\overline{\hat{D}\hat{E}}}[\mathbf{S}_1(\hat{\mathbf{\Theta}}_i,\hat{\phi}_{3,i})\mathbf{n}_{\overline{\hat{D}\hat{E}}}]ds = 0 \tag{6.8b}$$

  $\mathbf{n}_{\overline{\hat{D}\hat{E}}} = (n_r, n_z)^T$ is a unit vector normal to the approximate interface $\overline{\hat{D}\hat{E}}$ and $\mathbf{S}_1$ is the stress tensor defined in (6.3j).

- continuity of the divergence of the velocity

$$\hat{\nabla}\cdot\hat{\mathbf{\Theta}}_i^+\left(\frac{\hat{D}+\hat{E}}{2}\right) = \hat{\nabla}\cdot\hat{\mathbf{\Theta}}_i^-\left(\frac{\hat{D}+\hat{E}}{2}\right) \tag{6.8c}$$

- Lagrange and scaling conditions

$$\hat{\phi}_{1,i}(\hat{A}_j) = \delta_{i,j},\quad \hat{\phi}_{2,i}(\hat{A}_j) = \delta_{i,j+4},\ j=1,2,3,4,\ \textit{and}\ \frac{1}{|\hat{T}|}\int_{\hat{T}}\hat{\phi}_{3,i} = \delta_{i,9}. \tag{6.8d}$$

Note that the continuity of the second derivatives in (6.8a) is equivalent to $d_1^+ = d_1^-$ and $d_2^+ = d_2^-$ which by using $d_1^+ = d_1^- = d_1$ and $d_2^+ = d_2^- = d_2$ in (6.7a-6.7b) reduces the number of unknown coefficients to 16.

Conditions (6.8a) - (6.8d) lead to a linear system $\mathbf{M}\mathbf{c}_i = \mathbf{b}_i$ for the coefficients

$$\mathbf{c}_i = (a_1^+, b_1^+, c_1^+, d_1, a_1^-, b_1^-, c_1^-, a_2^+, b_2^+, c_2^+, d_2, a_2^-, b_2^-, c_2^-, a_3^-, a_3^+)^T.$$

The matrix $\mathbf{M}$ for a reference element of Type I is

$$\begin{pmatrix}
1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & \hat{d} & 0 & -1 & 0 & -\hat{d} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \hat{d} & 0 & -1 & 0 & -\hat{d} & 0 & 0\\
0 & m_{52} & m_{53} & m_{54} & 0 & m_{56} & \nu^-\hat{e} & 0 & -\nu^+\hat{e} & 0 & m_{511} & 0 & \nu^-\hat{e} & 0 & -\hat{d} & \hat{d}\\
0 & 0 & m_{63} & m_{64} & 0 & 0 & \nu^-\hat{d} & 0 & -\nu^+\hat{d} & m_{610} & m_{611} & 0 & \nu^-\hat{d} & m_{614} & -\hat{e} & \hat{e}\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{1515} & m_{1516}\\
-1 & -\hat{e} & -\frac{\hat{d}}{2} & 0 & 1 & \hat{e} & \frac{\hat{d}}{2} & 0 & 0 & -\frac{\hat{e}}{2} & 0 & 0 & 0 & \frac{\hat{e}}{2} & 0 & 0
\end{pmatrix}, \tag{6.9}$$

where

$$m_{52} = -2\nu^+\hat{d},\quad m_{53} = -\nu^+\hat{e},\quad m_{54} = \frac{1}{2}(\nu^- - \nu^+)\left(\hat{e}^2 + 2\hat{d}^2\right), m_{56} = 2\nu^-\hat{d},$$

$$m_{511} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d},$$

$$m_{63} = -\nu^+\hat{d}, \quad m_{64} = \frac{1}{2}(\nu^- - \nu^+)\hat{e}\hat{d}, \quad m_{610} = -2\nu^+\hat{e},$$

$$m_{611} = \frac{1}{2}(\nu^- - \nu^+)\left(2\hat{e}^2 + \hat{d}^2\right), m_{614} = 2\nu^-\hat{e},$$

$$m_{1515} = \frac{1}{2}\hat{e}\hat{d}, \quad m_{1516} = 1 - \frac{1}{2}\hat{e}\hat{d}.$$

For a reference element of Type II, the matrix $\mathbf{M}$ is

$$\begin{pmatrix}
1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & d & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{e} & 0 & 0 & -1 & -\hat{e} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{d} & 1 & 0 & -1 & -\hat{d} & -1 & 0 & 0 \\
0 & -2\nu^+ & m_{53} & m_{54} & 0 & 2\nu^- & m_{57} & 0 & m_{59} & 0 & m_{510} & 0 & m_{513} & 0 & -1 & 1 \\
0 & 0 & -\nu^+ & m_{64} & 0 & 0 & \nu^- & 0 & -\nu^+ & m_{610} & m_{611} & 0 & \nu^- & m_{614} & \hat{d}-\hat{e} & m_{616} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\hat{d}+\hat{e}}{2} & m_{1516} \\
-1 & -\hat{e} & -\frac{\hat{d}}{2} & 0 & 1 & \hat{e} & \frac{\hat{d}}{2} & 0 & 0 & -\frac{\hat{e}}{2} & 0 & 0 & 0 & \frac{\hat{e}}{2} & 0 & 0
\end{pmatrix}, \quad (6.10)$$

$$m_{53} = \nu^+(\hat{d}-\hat{e}), \; m_{54} = -\frac{1}{2}(\nu^- - \nu^+)\left(-2 + \hat{d}^2 - \hat{e}^2\right), \; m_{57} = \nu^-(-\hat{d}+\hat{e}), \;\; m_{5,9} = \nu^+(\hat{d}-\hat{e}),$$

$$m_{511} = -\frac{1}{2}(\nu^- - \nu^+)(\hat{d}-\hat{e}), \; m_{513} = \nu^-(-\hat{d}+\hat{e}),$$

$$m_{64} = \frac{1}{2}(\nu^- - \nu^+)(\hat{d}+\hat{e}), m_{610} = 2\nu^+(\hat{d}-\hat{e}), \; m_{611} = -\frac{1}{2}(\nu^- - \nu^+)\left(-1 + 2\hat{d}^2 - 2\hat{e}^2\right),$$

$$m_{614} = 2\nu^-(-\hat{d}+\hat{e}), m_{1516} = \frac{1}{2}(2 - \hat{d} - \hat{e}),$$

and the vector $\mathbf{b}_i$ for both Type I and Type II elements is

$$\mathbf{b}_i = (0,0,0,0,0,0,\delta_{i,1},\delta_{i,2},\delta_{i,3},\delta_{i,4},\delta_{i,5},\delta_{i,6},\delta_{i,7},\delta_{i,8},\delta_{i,9},0)^T.$$

We note that, other than the last row which corresponds to the conservation of mass constraint (6.8c), the matrices $\mathbf{M}$ in (6.9) and (6.10) are exactly the same as the ones defined in Section 3.1. This implies that the shape functions constructed in this section are essentially different from those constructed in Section 3.1 despite several similarities. Once these IFE

shape functions are constructed on the reference element, the standard affine mapping is applied to obtain the corresponding vector IFE shape functions on an interface element T as $\mathbf{\Phi}_i(r,z) = \hat{\mathbf{\Phi}}_i(F^{-1}(\hat{r},\hat{z})), \quad i = 1,2,\ldots,9.$

On every non-interface element T we use the standard finite element shape functions $\mathbf{\Psi}_i$, $i = 1,2,\ldots,9$ defined in (3.8). Figure 6.2 presents illustrations for the shape functions $\mathbf{\Psi}_1$ and $\mathbf{\Phi}_1$. Unlike $\mathbf{\Psi}_1$ used in the standard Q_1/Q_0 finite element space, the components of the IFE shape function $\mathbf{\Phi}_1$ cannot be decoupled, *i.e.,* its second and third components are not zero.

Then, the two-dimensional shape functions defined above are used to construct the global IFE space on Ω for the axisymmetric Stokes interface problem as follows:

$$S_h^s(\Omega) = \{\mathbf{U}_h \mid \mathbf{U}_h|_T \in X_h^s(T)\}, \tag{6.11}$$

where

$$X_h^s(T) = \begin{cases} \text{span}\{\mathbf{\Phi}_i(r,z), i = 1,2,\ldots,9\}, & \text{if } T \in \mathcal{T}_h^i, \\ \text{span}\{\mathbf{\Psi}_i(r,z), i = 1,2,\ldots,9\}, & \text{if } T \in \mathcal{T}_h^n. \end{cases}$$

6.1.2 Q_1/Q_0 particular IFE functions with surface force

In the case where the jump condition (6.3e) is such that $\boldsymbol{\sigma} \neq \mathbf{0}$, we use the idea presented in Section 3.3. On each interface element T, the particular IFE functions are defined as

$$\Upsilon_j = \begin{pmatrix} \mathbf{\Lambda}_j \\ \psi_j \end{pmatrix}, \ j = 1,2.$$

whose velocity component is

$$\mathbf{\Lambda}_j(r,z) = \begin{cases} \mathbf{\Lambda}_j^+(r,z) & \text{on } T^+ \\ \mathbf{\Lambda}_j^-(r,z) & \text{on } T^- \end{cases}, \tag{6.12}$$

with

$$\mathbf{\Lambda}_j^s(r,z) = \begin{pmatrix} \Lambda_{1,j}^s(r,z) \\ \Lambda_{2,j}^s(r,z) \end{pmatrix}, \quad \text{on } T^s, \quad s = +,-, \tag{6.13}$$

and the pressure component is a piecewise constant function such that

$$\psi_j = \begin{cases} \psi_j^+(r,z), & \text{on } T^+ \\ \psi_j^-(r,z), & \text{on } T^- \end{cases}, \quad \psi_j^s(r,z) = a_3^s, \quad s = +,-.$$

We then define the velocity component of the particular IFE functions to be piecewise polynomials

$$\Lambda_{i,j}^s(r,z) = a_{i,j}^s + b_{i,j}^s r + c_{i,j}^s z + d_{i,j}^s rz, \quad j = 1,2, \quad i = 1,2, \quad s = +,-. \tag{6.14}$$

We further require that the particular IFE functions satisfy the following constraints

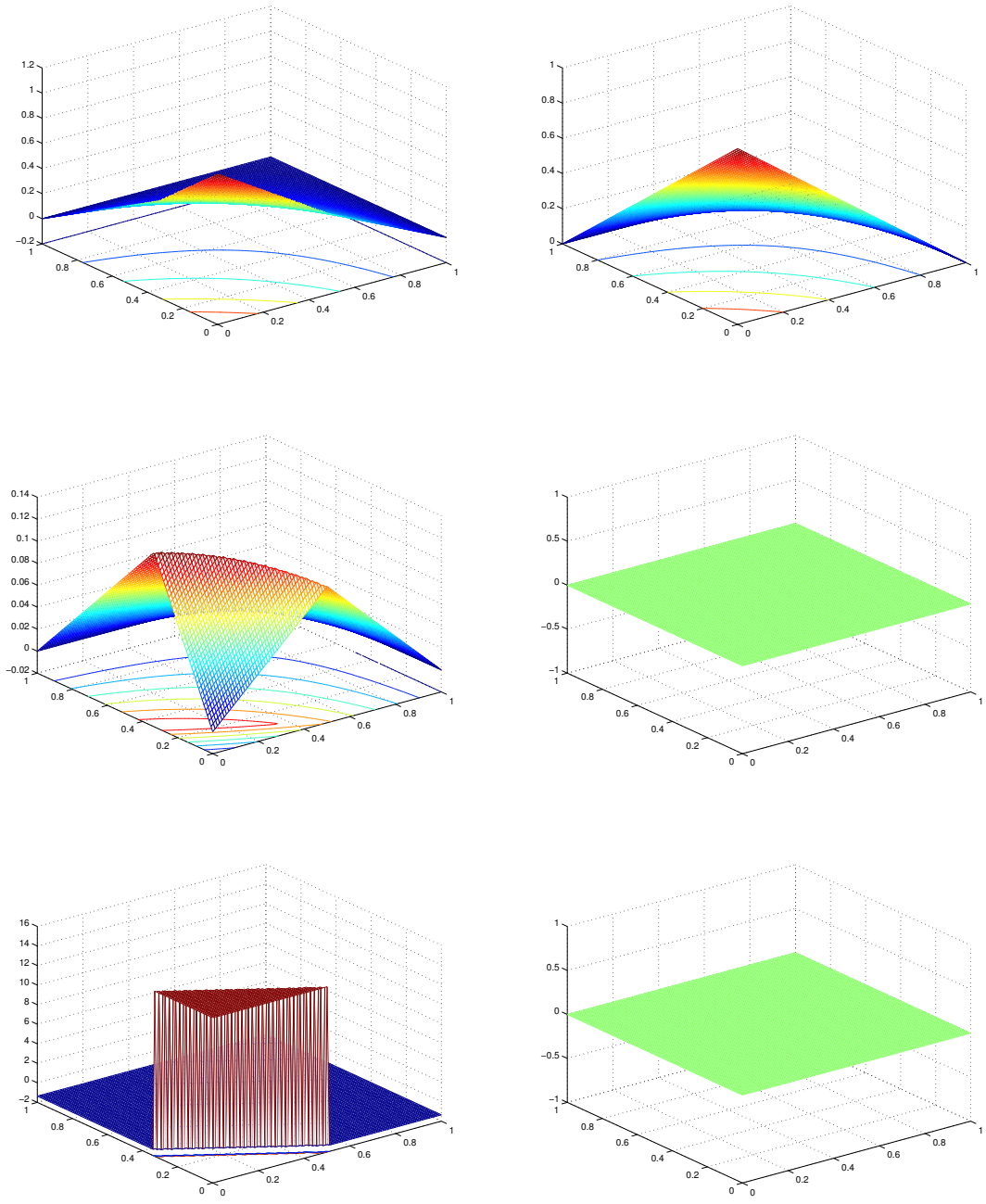


Figure 6.2: The (u_r, u_z, p) components of the IFE shape function $\mathbf{\Phi}_1$ (left) and standard shape function $\mathbf{\Psi}_1$ (right).

$$\mathbf{\Lambda}_j(A_i) = 0, \quad i = 1, 2, 3, 4, \tag{6.15}$$

$$\mathbf{\Lambda}_j^-(E) = \mathbf{\Lambda}_j^+(E), \quad \mathbf{\Lambda}_j^-(D) = \mathbf{\Lambda}_j^+(D), \quad \frac{\partial^2 \mathbf{\Lambda}_j^-}{\partial r \partial z} = \frac{\partial^2 \mathbf{\Lambda}_j^+}{\partial r \partial z}, \tag{6.16}$$

$$\int_{\overline{DE}} [\mathbf{S}_1(\mathbf{\Lambda}_j, \psi_j)\mathbf{n}_{\overline{DE}}] ds = \mathbf{e}_j, \tag{6.17}$$

$$\frac{1}{|T|}\int_T \psi_j dX = 0, \nabla \cdot \mathbf{\Lambda}_j^+ \left(\frac{D+E}{2}\right) = \nabla \cdot \mathbf{\Lambda}_j^- \left(\frac{D+E}{2}\right), \tag{6.18}$$

where $\mathbf{n}_{\overline{DE}}$ is the unit normal vector used in (6.8b), $\mathbf{e}_j$ is the canonical vector in $\mathbb{R}^2$ and $\mathbf{S}_1$ is defined in (6.3j). Note that these conditions lead to a linear system for determining the parameters of a particular IFE function, and the matrix of this linear system, after mapping of (6.15)-(6.18) to the reference element, is exactly the same as the one used to find the IFE shape functions in the previous section. However, the right hand-side $\mathbf{b}_j$ is different and is given by

$$\mathbf{b}_j = (0, 0, 0, 0, \delta_{j,1}, \delta_{j,2}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T.$$

For the Stokes interface problem with a nonzero interface surface force $\boldsymbol{\sigma} \neq \mathbf{0}$, we look for an IFE solution $(\mathbf{u}_h, p_h)$ in $S_h^s(\Omega) \oplus \{\mathbf{q}_h(r, z)\}$ where $S_h^s(\Omega)$ is the IFE space defined in section 6.1.1 and the vector function $\mathbf{q}_h(r, z)$ is such that

$$\mathbf{q}_h(r, z) = \begin{cases} \kappa_1 \Upsilon_1(r, z) + \kappa_2 \Upsilon_2(r, z), & \text{on T } \in \mathcal{T}_h^i \\ 0 & \text{elsewhere} \end{cases}, \tag{6.19}$$

where $\kappa_1 = \int_{\overline{DE}} \sigma_{h,r} ds$, $\kappa_2 = \int_{\overline{DE}} \sigma_{h,z} ds$ and $\boldsymbol{\sigma}_h = (\sigma_{h,r}, \sigma_{h,z})^T$ is the linear interpolation of $\boldsymbol{\sigma} = (\sigma_r, \sigma_z)^T$ defined by $\boldsymbol{\sigma}_h(E) = \boldsymbol{\sigma}(E)$ and $\boldsymbol{\sigma}_h(D) = \boldsymbol{\sigma}(D)$. In fact, the IFE solution on an interface element will take the form

$$\begin{pmatrix} \mathbf{u}_h \\ p_h \end{pmatrix} = \sum_{i=1}^{9} c_i \mathbf{\Phi}_i + \kappa_1 \Upsilon_1 + \kappa_2 \Upsilon_2, \tag{6.20}$$

where $\mathbf{u}_h = (u_{h,r}, u_{h,z})^T$.

We note that since the velocity and pressure of the fluids modeled using axisymmetric Stokes problem don't depend on the variable θ, our two-dimensional IFE approximation in the meridian plane is sufficient to capture the main features of the three-dimensional system.

6.1.3 Basic properties of the Q_1/Q_0 IFE space

In this section, we investigate the basic properties of the IFE shape functions for the axisymmetric Stokes interface problem. We start by showing the existence and uniqueness of IFE shape functions for an arbitrary configuration of ν^-, ν^+ and interface location.

Proposition 6.1.1. *On every interface element T, there exists a unique set of linearly independent IFE shape functions $\{\mathbf{\Phi}_i,\ i = 1, 2, \ldots, 9\}$ as defined in section 6.1.1.*

Proof. The uniqueness and existence of IFE shape functions are directly guaranteed by the invertibility of the matrices (6.9) and (6.10).

For interface elements of Type I a direct computation of the determinant of the matrix $\mathbf{M}$ in (6.9) yields

$$det(\mathbf{M}) = -\frac{1}{4}\hat{e}(\hat{e}^2 + \hat{d}^2)(Q_1\nu^- + Q_2\nu^+), \tag{6.21}$$

$$\begin{aligned} Q_1 &= (-2\hat{e}^2(-1+\hat{d}) + \hat{e}^3\hat{d} + 2\hat{d}^2 + \hat{e}(-2+\hat{d})\hat{d}^2), \\ Q_2 &= -\hat{e}\hat{d}(-2\hat{e} + \hat{e}^2 + (-2+\hat{d})\hat{d}). \end{aligned}$$

Thus it suffices to show that $Q_1 > 0$ and $Q_2 > 0$. This is done by noting that Q_1 and Q_2 are exactly the same as in the proof of Proposition 3.4.1.

For interface elements of Type II the determinant of the matrix $\mathbf{M}$ in (6.10) is such that

$$det(\mathbf{M}) = -\frac{1}{4}\left(\hat{d}^3 - \hat{d}^2\hat{e} - \hat{d}\hat{e}^2 + \hat{d} + \hat{e}^3 + \hat{e}\right)(P_1\nu^- + P_2\nu^+), \tag{6.22}$$

where

$$P_1 = (2 + \hat{d}^3 - \hat{e} - \hat{d}^2\hat{e} + \hat{e}^3 - \hat{d}(1+\hat{e}^2)),$$
$$P_2 = (-\hat{d}^3 + \hat{e} + 2\hat{e}^2 - \hat{e}^3 + \hat{d}^2(2+\hat{e}) + \hat{d}(1 - 4\hat{e} + \hat{e}^2)).$$

Again, we note that P_1 and P_2 are the same as in Proposition 3.4.1, therefore $P_1, P_2 > 0$, for all $0 < \hat{e}, \hat{d} < 1$. Also, for all $0 < \hat{e}, \hat{d} < 1$,

$$\begin{aligned} \left(\hat{d}^3 - \hat{d}^2\hat{e} - \hat{d}\hat{e}^2 + \hat{d} + \hat{e}^3 + \hat{e}\right) &\geq \left(\hat{d}^3 - 2\hat{d}\hat{e} + \hat{d} + \hat{e}^3 + \hat{e}\right) \\ &\geq d^3 + e^3 > 0. \end{aligned}$$

Therefore, $det(\mathbf{M}) > 0$ and the proof is completed. □

The next proposition shows that the velocity component of an IFE function is continuous on each element.

Proposition 6.1.2. *The velocity component of an IFE function is continuous on each element.*

Proof. The result is established using a similar argument as in Proposition 3.4.2. □

The next result is fairly easy to verify.

Proposition 6.1.3. *The IFE Q_1/Q_0 space $S_h^s(\Omega)$ over a mesh $\mathcal{T}_h$ of Ω defined in section 6.1.1 has the same dimension as the standard Q_1/Q_0 finite element space on the same mesh.*

Next we show that the IFE shape functions on each interface element form a partition of unity.

Proposition 6.1.4. *If $\mathbf{\Phi}_i,\ i = 1, 2, \ldots, 9$ are the IFE shape functions on an interface element, then*

$$\sum_{i=1}^{4} \mathbf{\Phi}_i(r,z) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \sum_{i=4}^{8} \mathbf{\Phi}_i(r,z) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tag{6.23}$$

$$\mathbf{\Phi}_9(r,z) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \forall (r,z) \in T. \tag{6.24}$$

Proof. The proof is accomplished by a direct verification. □

The next proposition shows that the proposed Q_1/Q_0 IFE shape functions are consistent with the associated standard finite element shape functions.

Proposition 6.1.5. *If $\nu^+ = \nu^-$ on an interface element T, then the Q_1/Q_0 IFE and standard Lagrange Q_1/Q_0 finite element shape functions are the same.*

Proof. The result is established by using the same argument as in Proposition 3.4.5. □

6.2 An immersed discontinuous Galerkin method

In this section, we derive the weak formulation for the axisymmetric Stokes interface problem. We define the following spaces for the velocity $\underline{\mathbf{u}} = (u_1, u_2, u_3)^T$ and the pressure p

$$\underline{\mathcal{H}}^m(V) = \{(\underline{\mathbf{u}}, p) : \underline{\mathbf{u}} \in C^0(V),\ \underline{\mathbf{u}} \in (H^m(A))^3 \text{ and } p \in H^1(A),\ \ \forall A \subset V^{\pm}\},$$

where H^m is the standard Sobolev space,

$$\underline{\mathcal{H}}_0^m(V) = \{(\underline{\mathbf{u}}, p) \in \underline{\mathcal{H}}^m(V) :\ \underline{\mathbf{u}}|_{\partial V} = \mathbf{0}\}.$$

We further define on Ω the following spaces for the velocity $\mathbf{u} = (u_r, u_z)^T$ and pressure p,

$$\mathcal{H}^m(\Omega) = \{(\mathbf{u}, p) : \mathbf{u} \in C^0(\Omega),\ \mathbf{u} \in (H^m(A))^2 \text{ and } p \in H^1(A),\ \ \forall A \subset \Omega^{\pm}\},$$

and

$$\mathcal{H}_0^m(\Omega) = \{(\mathbf{u}, p) \in \mathcal{H}^m(\Omega) :\ \mathbf{u}|_{\partial\Omega} = \mathbf{0}\}.$$

In order to construct the weak formulation we assume that $(\underline{\mathbf{u}}, p) \in \underline{\mathcal{H}}^2(V)$ is the solution of Stokes problem (6.1a-6.1b). On an arbitrary face $e^3 \in \mathcal{E}_h^3$ shared by two elements T_R^3 and T_L^3 we define the average and jump of a function $\underline{\mathbf{u}}$ across a face e^3 as

$$\{\underline{\mathbf{u}}\}_{e^3} = \frac{1}{2}(\underline{\mathbf{u}}|_{T_R^3} + \underline{\mathbf{u}}|_{T_L^3}),$$

$$[\underline{\mathbf{u}}]_{e^3} = (\underline{\mathbf{u}}|_{T_R^3} - \underline{\mathbf{u}}|_{T_L^3}).$$

If the face e^3 coincides with the domain boundary (*i.e.* $e^3 \subset \partial V$) we define the average and jump by

$$\{\underline{\mathbf{u}}\}_{e^3} = \underline{\mathbf{u}}|_{e^3},$$

and

$$[\underline{\mathbf{u}}]_{e^3} = \underline{\mathbf{u}}|_{e^3}.$$

Following the standard procedure, for instance, in [120] we multiply the system (6.1a-6.1b) by $(\underline{\mathbf{v}}, q) \in \underline{\mathcal{H}}_0^1(V)$ and integrate over an arbitrary non-interface element $T^3 \in \mathcal{T}_h^3$ to obtain

$$-\int_{T^3} (\nabla \cdot \mathbf{L}(\underline{\mathbf{u}}, p)) \cdot \underline{\mathbf{v}} dX - \int_{T^3} \nabla \cdot \underline{\mathbf{u}} q dX = \int_{T^3} \underline{\mathbf{f}} \cdot \underline{\mathbf{v}} dX, \tag{6.25}$$

where $X = (x, y, z)$ denotes the three-dimensional Cartesian coordinates. Then, We apply the divergence theorem to integrate the first term by parts and obtain

$$-\int_{\partial T^3} (\mathbf{L}(\underline{\mathbf{u}}, p)\underline{\mathbf{n}}) \cdot \underline{\mathbf{v}} dA + \int_{T^3} \mathbf{L}(\underline{\mathbf{u}}, p) : \nabla \underline{\mathbf{v}} dX - \int_{T^3} \nabla \cdot \underline{\mathbf{u}} q dX = \int_{T^3} \underline{\mathbf{f}} \cdot \underline{\mathbf{v}} dX, \tag{6.26}$$

where $\mathbf{A} : \mathbf{B} = \sum_{i=1}^3 \sum_{j=1}^3 A_{ij} B_{ij}$. and $\underline{\mathbf{n}} = (n_x, n_y, n_z)^T$ is the outwards normal vector to the cylindrical element T^3.

On an interface element T^3, we apply the divergence theorem on $T^3 \cap V^+$ and $T^3 \cap V^-$ and take into account that the exact solution satisfies the jump conditions (6.1d) and (6.1e) across the interface to obtain

$$-\int_{\partial T^3} (\mathbf{L}(\underline{\mathbf{u}}, p)\underline{\mathbf{n}}) \cdot \underline{\mathbf{v}} dA + \int_{T^3} \mathbf{L}(\underline{\mathbf{u}}, p) : \nabla \underline{\mathbf{v}} dX - \int_{T^3} \nabla \cdot \underline{\mathbf{u}} q dX = \int_{T^3} \underline{\mathbf{f}} \cdot \underline{\mathbf{v}} dX + \int_{T^3 \cap \Xi} \underline{\boldsymbol{\sigma}} \cdot \underline{\mathbf{v}} dA. \tag{6.27}$$

Using the symmetry of the problem around the z-axis, we project (6.27) onto the meridian plane Ω and sum over all elements of the mesh $\mathcal{T}_h$ to obtain

$$2\pi \left(-\sum_{T \in \mathcal{T}_h} \int_{\partial T} (\mathbf{S}_1(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} r ds + \int_\Omega \mathbf{S}(\mathbf{u}, p) : \nabla \mathbf{v} r dr dz - \int_\Omega \nabla \cdot \mathbf{u} q r dr dz \right) = 2\pi \left(\int_\Omega \mathbf{f} \cdot \mathbf{v} r dr dz + \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} r ds \right), \tag{6.28}$$

where $\mathbf{n} = (n_r, n_z)^T$ is the outward unit vector normal to the projected element T in the meridian plane, $\mathbf{u} = (u_r, u_z)^T$, $\mathbf{v} = (v_r, v_z)^T$, $\mathbf{S}$ and $\mathbf{S}_1$ are the stress tensors defined in (6.3f) and (6.3j), respectively and $\boldsymbol{\sigma} = (\sigma_r, \sigma_z)^T$ is the surface force. The operators $\nabla\cdot$ and ∇ are defined in (6.3i) and (6.3h).

By rewriting (6.28), we obtain

$$-\sum_{e\in\mathcal{E}_h}\int_e[(\mathbf{S}_1(\mathbf{u},p)\mathbf{n})\cdot\mathbf{v}]rds+\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}rdrdz-\int_\Omega\nabla\cdot\mathbf{u}qrdrdz = \int_\Omega\mathbf{f}\cdot\mathbf{v}rdrdz+\int_\Gamma\boldsymbol{\sigma}\cdot\mathbf{v}rds. \tag{6.29}$$

Applying the identity $ab-cd=\frac{1}{2}(a+c)(b-d)+\frac{1}{2}(a-c)(b+d)$ yields

$$-\sum_{e\in\mathcal{E}_h}\int_e\left([(\mathbf{S}_1(\mathbf{u},p)\mathbf{n})]\cdot\{\mathbf{v}\}+\{(\mathbf{S}_1(\mathbf{u},p)\mathbf{n})\}\cdot[\mathbf{v}]\right)rds$$
$$+\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}rdrdz-\int_\Omega\nabla\cdot\mathbf{u}qrdrdz=\int_\Omega\mathbf{f}\cdot\mathbf{v}rdrdz+\int_\Gamma\boldsymbol{\sigma}\cdot\mathbf{v}rds. \tag{6.30}$$

Every interface edge can be expressed as $e=e^+\cup e^-$, where $e^\pm=e\cap\Omega^\pm$. Hence

$$\int_e[\mathbf{S}_1(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}rds=\int_{e^+}[\mathbf{S}_1(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}rds+\int_{e^-}[\mathbf{S}_1(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}rds.$$

Since $(\mathbf{u},p)\in\mathcal{H}^2(\Omega)$, we write

$$\int_e[\mathbf{S}_1(\mathbf{u},p)\mathbf{n}]\cdot\{\mathbf{v}\}rds=0, \tag{6.31}$$

which also holds for non-interface elements.

Combining (6.30) and (6.31) with $(\mathbf{u},p)\in\mathcal{H}^2(\Omega)$ and assuming $\mathbf{u}$ continuous lead to the interior penalty weak formulation

$$\int_\Omega\mathbf{S}(\mathbf{u},p):\nabla\mathbf{v}dx-\int_\Omega\nabla\cdot\mathbf{u}qdx-\sum_{e\in\mathcal{E}_h}\int_e\{(\mathbf{S}_1(\mathbf{u},p)\mathbf{n})\}\cdot[\mathbf{v}]ds$$
$$+\sum_{e\in\mathcal{E}_h}\frac{\alpha}{h_e}\int_e\nu[\mathbf{u}]\cdot[\mathbf{v}]ds+\gamma\sum_{e\in\mathcal{E}_h}\int_e\{(\nu\boldsymbol{\epsilon}_1(\mathbf{v})\mathbf{n})\}\cdot[\mathbf{u}]ds+\sum_{e\in\mathcal{E}_h}\int_e\{q\}[\mathbf{u}]\cdot\mathbf{n}=\int_\Omega\mathbf{f}\cdot\mathbf{v}dx$$
$$+\int_\Gamma\boldsymbol{\sigma}\cdot\mathbf{v}ds+\gamma\sum_{e\subset\partial\Omega}\int_e\{(\nu\boldsymbol{\epsilon}_1(\mathbf{v})\mathbf{n})\}\cdot[\mathbf{g}]ds+\sum_{e\subset\partial\Omega}\int_e\{q\}[\mathbf{g}]\cdot\mathbf{n}+\sum_{e\subset\partial\Omega}\frac{\alpha}{h_e}\int_e\nu[\mathbf{g}]\cdot[\mathbf{v}]ds, \tag{6.32}$$

where α is a positive stabilization parameter and $\gamma=1$ for the nonsymmetric weak formulation (NIPG) while $\gamma=-1$ for the symmetric weak formulation (SIPG).

The weak form above leads to the interior penalty formulation for the Stokes problem consisting of finding $(\mathbf{u}, p) \in \mathcal{H}^1(\Omega)$ such that $\mathbf{u} = \mathbf{g}$ on $\partial\Omega$ and

$$\begin{cases} A(\mathbf{u}, \mathbf{v}) + B(\mathbf{v}, p) & = L(\mathbf{v}) \\ B(\mathbf{u}, q) & = l(q) \end{cases}, \qquad \forall\ (\mathbf{v}, q) \in \mathcal{H}_0^1(\Omega), \tag{6.33a}$$

where

$$A(\mathbf{w}, \mathbf{v}) = \int_\Omega \nu \boldsymbol{\epsilon}_s(\mathbf{w}) : \nabla \mathbf{v} r dr dz - \sum_{e \in \mathcal{E}_h} \int_e \nu \{\boldsymbol{\epsilon}_1(\mathbf{w})\mathbf{n}\} \cdot [\mathbf{v}] r ds$$
$$+\gamma \sum_{e \in \mathcal{E}_h} \int_e \{\nu \boldsymbol{\epsilon}_1(\mathbf{v})\mathbf{n}\} \cdot [\mathbf{w}] + \sum_{e \in \mathcal{E}_h} \frac{\alpha}{h_e} \int_e \nu [\mathbf{w}] \cdot [\mathbf{v}] r ds, \tag{6.33b}$$

$$B(\mathbf{v}, q) = -\int_\Omega q \nabla \cdot \mathbf{v} r dr dz + \sum_{e \in \mathcal{E}_h} \int_e \{q\} [\mathbf{v}] \cdot \mathbf{n} r ds, \tag{6.33c}$$

$$L(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} r dr dz + \int_\Gamma \boldsymbol{\sigma} \cdot \mathbf{v} r ds + \gamma \sum_{e \subset \partial\Omega} \int_e \{\nu(\boldsymbol{\epsilon}_1(\mathbf{v})\mathbf{n})\} \cdot [\mathbf{g}] r ds$$
$$+ \sum_{e \subset \partial\Omega} \frac{\alpha}{h_e} \int_e \nu [\mathbf{g}] \cdot [\mathbf{v}] r ds, \tag{6.33d}$$

$$l(q) = \sum_{e \subset \partial\Omega} \int_e \{q\} [\mathbf{g}] \cdot \mathbf{n}. \tag{6.33e}$$

Then, this formulation leads to the DG-IFE method consisting of finding $(\mathbf{u}_h, p_h) \in S_h^s(\Omega) \oplus \{\mathbf{q}_h(r, z)\}$ such that

$$\begin{cases} A(\mathbf{u}_h, \mathbf{v}_h) + B(\mathbf{v}_h, p_h) & = L_h(\mathbf{v}_h) \\ B(\mathbf{u}_h, q_h) & = l(q_h) \end{cases}, \qquad \forall\ (\mathbf{v}_h, q_h) \in S_{h,0}^s(\Omega), \tag{6.34}$$

subject to the boundary conditions:

$$\mathbf{u}_h(R) = \mathbf{g}(R), \quad \text{for all mesh vertices } R \in \partial\Omega,$$

where

$$L_h(\mathbf{v}_h) = \int_\Omega \mathbf{f} \cdot \mathbf{v}_h dx + \sum_{T \in \mathcal{T}_h^i} \int_{\overline{DE}} \boldsymbol{\sigma}_h \cdot \mathbf{v}_h ds + \gamma \sum_{e \subset \partial\Omega} \int_e \{\nu \boldsymbol{\epsilon}(\mathbf{v}_h)\mathbf{n}\} \cdot [\mathbf{g}] ds + \sum_{e \subset \partial\Omega} \frac{\alpha}{h_e} \int_e \nu [\mathbf{g}] \cdot [\mathbf{v}_h] ds, \tag{6.35}$$

$\boldsymbol{\sigma}_h$ is the linear interpolant of $\boldsymbol{\sigma}$ on $\overline{DE}$ using the value of $\boldsymbol{\sigma}$ at D and E and

$$S_{h,0}^s(\Omega) = \{(\mathbf{v}, p) \in S_h^s(\Omega) :\ \mathbf{v}(R) = \mathbf{0}, \ \text{ for all mesh vertices } R \in \partial\Omega\}.$$

6.3 Numerical simulations

Example 6.3.1.

We consider the domain $\Omega = [0,1] \times [-1,1]$ cut by the half-circular interface $r^2 + z^2 = 0.3$, which separates Ω into two regions $\Omega^+ = \{(r,z) \in \Omega : r^2 + z^2 > 0.3\}$ and $\Omega^- = \{(r,z) \in \Omega : r^2 + z^2 < 0.3\}$. The functions $\mathbf{u} = (u_r, u_z)^T$ and p are chosen to satisfy jump conditions (6.3d) and (6.3e) and are given by

$$\mathbf{u} = \begin{cases} \mathbf{u}^- = \begin{pmatrix} 2rz\frac{-s^2+r^2+z^2}{\nu^- s^2} \\ \\ \frac{1}{\nu^-}(-s^2+r^2+z^2)(1-\frac{3r^2}{s^2}-\frac{z^2}{s^2}) \end{pmatrix}, & \text{if } (r,z) \in \Omega^- \\ \\ \mathbf{u}^+ = \begin{pmatrix} 2rz\frac{-s^2+r^2+z^2}{\nu^+ s^2} \\ \\ \frac{1}{\nu^+}(-s^2+r^2+z^2)(1-\frac{3r^2}{s^2}-\frac{z^2}{s^2}) \end{pmatrix}, & \text{if } (r,z) \in \Omega^+ \end{cases}, \tag{6.36a}$$

and

$$p = \begin{cases} e^{r^2+z^2} - 3.48673, & \text{if } (r,z) \in \Omega^+ \\ r^2 + z^2 - 3.48673, & \text{if } (r,z) \in \Omega^- \end{cases}, \tag{6.36b}$$

where $\nu^- = 1$, $\nu^+ = 10$ and

$$\boldsymbol{\sigma} = \begin{pmatrix} (e^{r^2+z^2} - r^2 - z^2)n_r \\ (e^{r^2+z^2} - r^2 - z^2)n_z \end{pmatrix}, \quad \text{if } (r,z) \in \Gamma. \tag{6.36c}$$

In order to show the approximation capability of the IFE space $S_h^s(\Omega)$ and the IFE particular functions, we create uniform Cartesian meshes having $N^2 = 5^2, 10^2, 20^2, 40^2, 80^2, 160^2, 320^2, 640^2$ square elements and show the L^2 interpolation errors in Table 6.1. Broken H^1 interpolation errors for the velocity are shown in Table 6.2. Note that we follow the same procedure as in Section 4.2.1 to compute the interpolation errors. Since the IFE shape functions are coupled, the i^{th} component of $\mathbf{I}_h(\mathbf{u},p)$ denoted by $(\mathbf{I}_h(\mathbf{u},p))_i$ is such that $(\mathbf{I}_h(\mathbf{u},p))_1 \approx u_r$, $(\mathbf{I}_h(\mathbf{u},p))_2 \approx u_z$ and $(\mathbf{I}_h(\mathbf{u},p))_3 \approx p$ with errors $\epsilon(u_1) = u_r - (\mathbf{I}_h(\mathbf{u},p))_1$, $\epsilon(u_2) = u_z - (\mathbf{I}_h(\mathbf{u},p))_2$ and $\epsilon(p) = p - (\mathbf{I}_h(\mathbf{u},p))_3$.

Next, we solve Stokes interface problem (6.3a)-(6.3b) using NIPG and SIPG methods. Tables 6.3-6.6 show the L^2 and H_1 errors for the NIPG and SIPG methods using uniform Cartesian meshes having $N^2 = 5^2, 10^2, 20^2, 40^2, 80^2, 160^2$ square elements. The results show optimal convergence rates for both velocity and pressure.

N	$\|\|\epsilon(u_1)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{0,\Omega}$	Order	$\|\|\epsilon(p)\|\|_{0,\Omega}$	Order
5	1.0856379e-01	0	2.3305131e-01	0	4.6908971e-01	0
10	2.7866727e-02	1.9619	5.9355848e-02	1.9732	2.8694522e-01	0.7091
20	7.0753794e-03	1.9777	1.4962605e-02	1.9880	1.5035181e-01	0.9324
40	1.7816441e-03	1.9896	3.7582846e-03	1.9932	7.8725548e-02	0.9334
80	4.4740453e-04	1.9936	9.4230218e-04	1.9958	4.1341933e-02	0.9292
160	1.1215141e-04	1.9961	2.3598395e-04	1.9975	2.1390042e-02	0.9507
320	2.8076001e-05	1.9980	5.9048054e-05	1.9987	1.1115584e-02	0.9444
640	7.0236840e-06	1.9990	1.4768365e-05	1.9994	5.7534917e-03	0.9501

Table 6.1: L^2 interpolation errors for Example 6.3.1.

N	$\|\|\epsilon(u_1)\|\|_{1,\Omega}$	Order	$\|\|\epsilon(u_2)\|\|_{1,\Omega}$	Order
5	8.1106161e-01	0	1.9389213	0
10	4.1144714e-01	0.9791	9.9411494e-01	0.9638
20	2.0794416e-01	0.9845	5.0629661e-01	0.9734
40	1.0437684e-01	0.9944	2.5484515e-01	0.9904
80	5.2440062e-02	0.9931	1.2823624e-01	0.9908
160	2.6279386e-02	0.9967	6.4425896e-02	0.9931
320	1.3155398e-02	0.9983	3.2377497e-02	0.9926
640	6.5802972e-03	0.9994	1.6269287e-02	0.9928

Table 6.2: H^1 interpolation errors for Example 6.3.1.

N	$\|\|u_r - u_{h,r}\|\|_{0,\Omega}$	Order	$\|\|u_z - u_{h,z}\|\|_{0,\Omega}$	Order	$\|\|p - p_h\|\|_{0,\Omega}$	Order
5	9.4319719e-02	0	2.2454486e-01	0	5.8772950	0
10	2.4047769e-02	1.9717	5.8767856e-02	1.9339	1.9042523	1.6259
20	5.4646708e-03	2.1377	1.3572075e-02	2.1144	5.9593073e-01	1.6760
40	1.5234498e-03	1.8428	3.8193315e-03	1.8292	2.2606959e-01	1.3984
80	3.8362178e-04	1.9896	9.5860167e-04	1.9943	8.2944194e-02	1.4466
160	9.6776069e-05	1.9870	2.3759725e-04	2.0124	3.3054746e-02	1.3273

Table 6.3: L^2 IFE errors for NIPG method applied to Example 6.3.1.

N	$\|\|u_r - u_{h,r}\|\|_{1,\Omega}$	Order	$\|\|u_z - u_{h,z}\|\|_{1,\Omega}$	Order
5	8.4612929e-01	0	2.0405984	0
10	4.2492877e-01	0.9937	1.0177753	1.0036
20	2.0172889e-01	1.0748	4.8156827e-01	1.0796
40	1.0540784e-01	0.9364	2.5618594e-01	0.9105
80	5.2750816e-02	0.9987	1.2856579e-01	0.9947
160	2.6422708e-02	0.9974	6.4525575e-02	0.9946

Table 6.4: H^1 IFE errors for NIPG method applied to Example 6.3.1.

N	$\|\|u_r - u_{h,r}\|\|_{0,\Omega}$	Order	$\|\|u_z - u_{h,z}\|\|_{0,\Omega}$	Order	$\|\|p - p_h\|\|_{0,\Omega}$	Order
5	7.6801230e-02	0	1.9014195e-01	0.	2.5120401	0
10	1.6765687e-02	2.1956	4.5489583e-02	2.0635	8.7545953e-01	1.5207
20	3.7367291e-03	2.1657	1.0956793e-02	2.0537	2.8165736e-01	1.6361
40	8.7682076e-04	2.0914	2.6871048e-03	2.0277	1.0518215e-01	1.4211
80	2.1483450e-04	2.0291	6.7097560e-04	2.0017	4.6934253e-02	1.1642
160	5.5364507e-05	1.9562	1.7094399e-04	1.9727	2.2991455e-02	1.0295

Table 6.5: L^2 IFE errors for SIPG method applied to Example 6.3.1.

N	$\|\|u_r - u_{h,r}\|\|_{1,\Omega}$	Order	$\|\|u_z - u_{h,z}\|\|_{1,\Omega}$	Order
5	9.1150148e-01	0	2.0019068e+00	0
10	4.3985982e-01	1.0512	1.0086862e+00	0.9889
20	2.1509537e-01	1.0321	5.1022581e-01	0.9833
40	1.0633369e-01	1.0164	2.5588889e-01	0.9956
80	5.3161310e-02	1.0002	1.2853372e-01	0.9934
160	2.6774145e-02	0.9895	6.4523829e-02	0.9942

Table 6.6: H^1 IFE errors for SIPG method applied to Example 6.3.1.

6.4 Axisymmetric Stokes problem with a moving interface

In this section, we present the algorithm used to solve the axisymmetric Stokes problem with moving interfaces. This includes determining the surface force, solving the Stokes interface problem, tracking the interface and determining the position of the mesh nodes with respect to the interface. We note that, due to the interface motion, the IFE space and particular functions evolve in time. Therefore, the operations described below must be carried out at every time step with the corresponding IFE space and particular functions. However, since the three-dimensional Stokes problem is reduced to a two-dimensional problem, most of the operations remain the same as the ones described in section 5.2. In fact, determining the position of the mesh nodes with respect to the interface can be done using the even-odd rule described in section 5.2.1 and the normal vector to the interface can be computed using the procedure introduced in section 5.2.2. However, to compute the surface force, we need to determine the curvature of the three-dimensional interface, otherwise, the computed surface force will be underestimated.

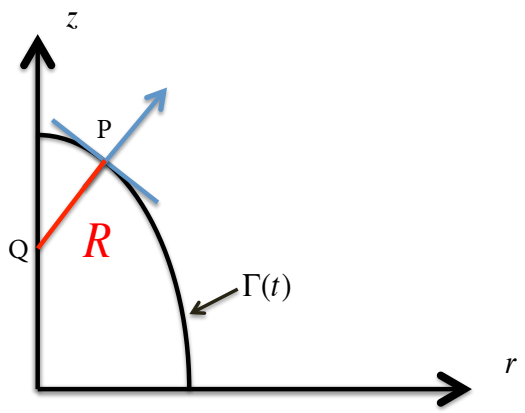


Figure 6.3: The distance R used to compute the curvature κ_2.

6.4.1 Surface force

The surface force is given by the expression

$$\boldsymbol{\sigma}(r,z,t) = -\sigma\kappa(r,z,t)\mathbf{n}(r,z,t), \tag{6.37}$$

where $\sigma \geq 0$ is the surface tension, $\kappa(r,z,t)$ is the curvature of the three-dimensional interface at the point (r,z) and $\mathbf{n}(r,z,t) = (n_r, n_z)^T$ is the unit normal outward vector to the interface at the point (r,z). As mentioned above, the normal vector is computed using the procedure presented in section 5.2.2. On the other hand, we note that the curvature κ at a given point $P(r,z)$ can be written as $\kappa = \kappa_1 + \kappa_2$ (see [132]), where κ_1 is the curvature computed in section 5.2.2 and κ_2 is defined as

$$\kappa_2 = \begin{cases} \frac{1}{R}, & \text{if} \quad r \neq 0 \\ \kappa_1, & \text{if} \quad r = 0 \end{cases}, \tag{6.38}$$

where $R = |PQ|$ and Q is the intersection point of the line normal to the interface at P with the z-axis as illustrated in Figure 6.3.

6.4.2 Proposed algorithm

To solve the axisymmetric Stokes problem with a moving interface, we need to implement a time integrator to solve the following ODEs:

$$\begin{cases} \frac{d}{dt}\mathcal{X}^{(i)}(t) & = \mathbf{u}_h(\mathcal{X}^{(i)}(t), t), \quad t \in [0, T] \\ \mathcal{X}^{(i)}(0) & = \mathcal{X}_0^{(i)} \end{cases}, \quad i = 1, 2, ..., N \tag{6.39}$$

where $\mathbf{u}_h(\mathcal{X}^{(i)}(t), t) = (u_{h,r}(\mathcal{X}^{(i)}(t), t), u_{h,z}(\mathcal{X}^{(i)}(t), t))^T$ is the velocity of the fluid at the point $\mathcal{X}^{(i)}(t)$, N is the number of control points and $\mathcal{X}_0$ is the set of the initial control points obtained by uniformly sampling the projected interface Γ. We use the algorithm described in Figure 5.4 to solve the axisymmetric problem (6.3a)-(6.3i) with a moving interface using the DG-IFE formulation (6.34) and ODEs (6.39).

In this section we use the trapezoidal method given by:

$$\mathcal{X}_{k+1}^{(i)} = \mathcal{X}_k^{(i)} + \frac{\Delta t}{2}\left(\mathbf{u}_h(\mathcal{X}_k^{(i)}, t_k) + \mathbf{u}_h(\mathcal{X}_{k+1}^{(i)}, t_{k+1})\right), \quad i = 1, 2, ..., N,$$

where N is the number of control points and $\mathcal{X}_k^{(i)}$ is the approximation of $\mathcal{X}^{(i)}(t_k)$ at time t_k.

This implicit time integrator is coupled with Newton's method, which yields the following procedure:

$$
\begin{cases}
\textit{Set up the initial time } t_0 = 0 \textit{ and select the time step } \Delta t,\ tol_1 \textit{ and } tol_2 : \\
\textit{For k=0,1,2,...} \\
\quad 1.\ \textit{Set } n = 0 \textit{ and } \epsilon_1 = 1 \\
\quad 2.\ \textit{Compute the initial guess } \mathcal{X}_{k+1}^{(i,n)} \textit{ using Euler method as:} \\
\qquad \mathcal{X}_{k+1}^{(i,n)} = \mathcal{X}_k^{(i)} + \Delta t \mathbf{u}_h \left(\mathcal{X}_k^{(i)}, t_k \right), \qquad i = 1, 2, ...N. \\
\quad 3.\ \textit{While } \epsilon_1 > tol_1 \\
\qquad \textit{- Determine the Jacobian of } \mathbf{u}_h(\mathcal{X}_{k+1}^{(i,n)}, t_{k+1}) \textit{ at the control points:} \\
\qquad\quad \mathbf{J}(\mathcal{X}_{k+1}^{(i,n)}) = I - \frac{1}{2}\Delta t \left(\begin{array}{cc} \frac{\partial}{\partial r}\mathbf{u}_h(\mathcal{X}_{k+1}^{(i,n)}, t_{k+1}) & \frac{\partial}{\partial z}\mathbf{u}_h(\mathcal{X}_{k+1}^{(i,n)}, t_{k+1}) \end{array} \right), \quad i = 1, 2, ...N. \\
\qquad \textit{- Evaluate } \mathbf{F}(\mathcal{X}_{k+1}^{(i,n+1)}) \textit{:} \\
\qquad\quad \mathbf{F}(\mathcal{X}_{k+1}^{(i,n)}) = \mathcal{X}_{k+1}^{(i,n)} - \mathcal{X}_k^{(i)} - \frac{1}{2}\Delta t \left(\mathbf{u}_h(\mathcal{X}_k^{(i)}, t_k) + \mathbf{u}_h(\mathcal{X}_{k+1}^{(i,n)}, t_{k+1}) \right), \quad i = 1, 2, ..., N. \\
\qquad \textit{- Update } \mathcal{X}_{k+1}^{(i,n+1)} \textit{:} \\
\qquad\quad \mathcal{X}_{k+1}^{(i,n+1)} = \mathcal{X}_{k+1}^{(i,n)} - \left(\mathbf{J}(\mathcal{X}_{k+1}^{(i,n)}) \right)^{-1} \mathbf{F}(\mathcal{X}_{k+1}^{(i,n)}), \quad i = 1, 2, ..., N. \\
\qquad \textit{- } \epsilon_1 = \max_{1 \le i \le N} ||\mathcal{X}_{k+1}^{(i,n+1)} - \mathcal{X}_{k+1}^{(i,n)})||,\ n = n + 1 \textit{:} \\
\quad 4.\ \mathcal{X}_{k+1}^{(i)} = \mathcal{X}_{k+1}^{(i,n+1)},\ i = 1, 2, ..., N. \\
\quad 5.\ \textit{Stop when } \max_{1 \le i \le N} ||\mathbf{u}_h(\mathcal{X}_{k+1}^{(i)}, t_{k+1}) - \mathbf{u}_h(\mathcal{X}^{(i)}(t_k), t_k)|| < tol_2.
\end{cases}
\tag{6.40}
$$

6.4.3 Numerical experiments

Example 6.4.1.

We validate our method by studying the behavior of a drop subjected to an extensional flow. We assume that the system is symmetric with respect to the z-axis and the xy-plane. Therefore, we can reduce it to a two-dimensional problem in the upper half of the meridian plane as shown in Figure 6.4.

We use the same parameters as in [132] with an initial drop of radius $R_0 = H/10$ and $H = W = 1$. The boundary conditions are $\mathbf{u}(r, z) = (u_r, u_z) = (-0.5\epsilon r, \epsilon z)$ if $r = W$ or $z = H$ such that $\epsilon > 0$ is the extension rate. On the boundaries defined by $r = 0$ and $z = 0$,

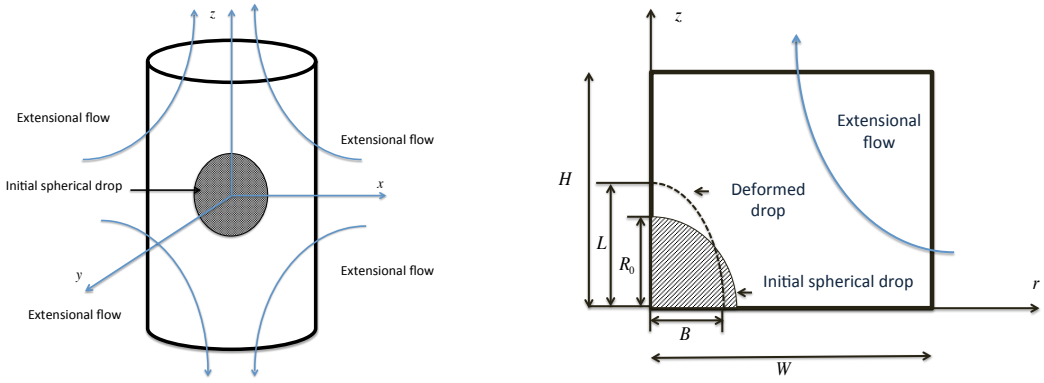
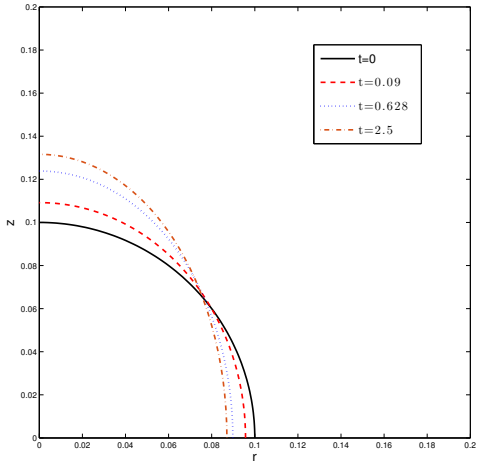


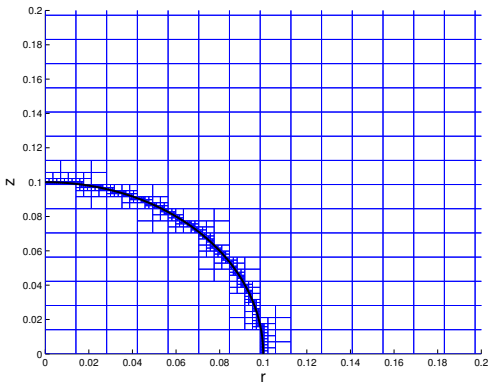
Figure 6.4: The computational domain for transient drop deformation under extensional flow in three-dimensional domain (left) and its projection onto the meridian plane (right).

we use the symmetry of the system to impose the following boundary conditions:

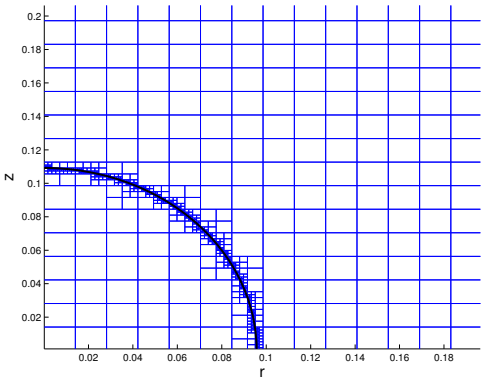
$$\begin{cases} u_r = 0, \ \frac{\partial}{\partial r} u_z = 0, & \text{if } r = 0, \\ u_z = 0, \ \frac{\partial}{\partial z} u_r = 0, & \text{if } z = 0. \end{cases}$$

We denote by ν^+ and ν^- the viscosity of the matrix fluid (*i.e.* the fluid outside the drop) and the viscosity of the drop, respectively. The capillary number is then defined as $Ca = \frac{\nu^+ \epsilon R_0}{\sigma}$ and the viscosity ratio is defined as $\beta = \frac{\nu^-}{\nu^+}$. The simulation is carried out using the SIPG method with $Ca = 0.1$, $\beta = 0.5$, the trapezoidal method (6.40) with uniform time step $\Delta t = 10^{-3}$ and meshes having at most 6000 elements obtained by locally refining a background 71×71 uniform mesh four times. We plot the interfaces and the meshes used to partition the domain at $t = 0, 0.09, 0.628, 2.5$ in Figure 6.5, and we plot the corresponding three-dimensional drops in Figure 6.6 .

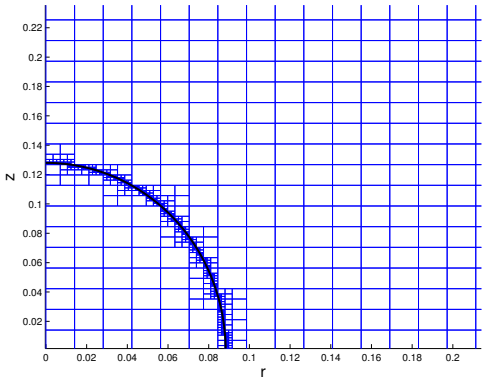




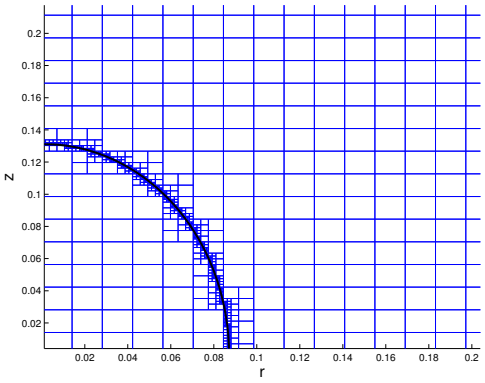
(a) t=0



(b) t=0.09

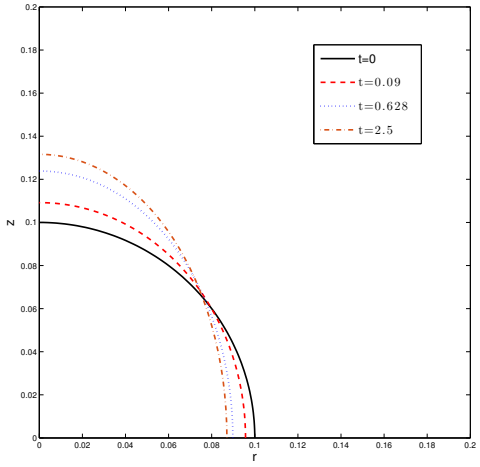


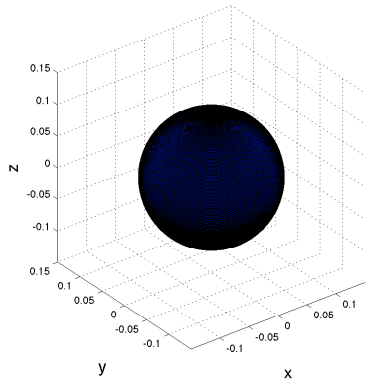
(c) t=0.628



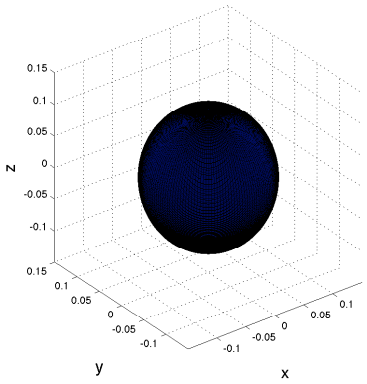
(d) t=2.5

Figure 6.5: Drop deformation in extensional flow versus time (top) and blown-ups, near the interface, of the meshes used to partition the domain Ω (bottom) for Example 6.4.1.

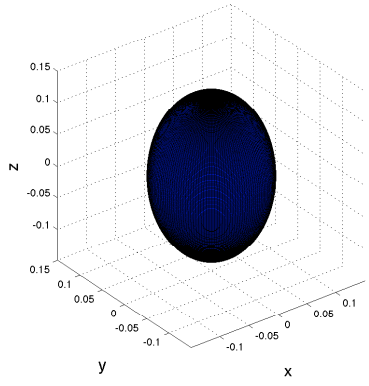




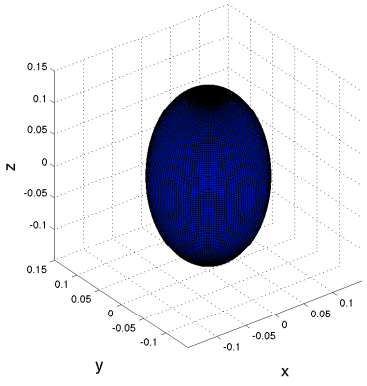
(a) t=0



(b) t=0.09



(c) t=0.628



(d) t=2.5

Figure 6.6: Drop deformation in extensional flow versus time (top) and blown-ups of the three-dimensional drops (bottom) for Example 6.4.1.

Next, we define the deformation $D = \frac{L}{R_0}$ (L is the length of the deformed drop shown in Figure 6.4), and plot it versus time in Figure 6.7, together with the results obtained by Yue *et al.* in [132]. We observe good agreement between our results and those obtained in [132] for $t < 1.5$ with spurious oscillations appearing near the equilibrium state.

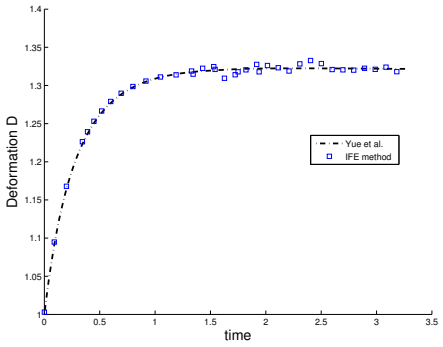


Figure 6.7: Drop deformation in extensional flow versus time for Example 6.4.1.

These spurious oscillations are caused by the clustering of the control points near the upper tip of the drop as shown in Figure 6.8, which makes the curvature computation very sensitive to errors in the control points locations.

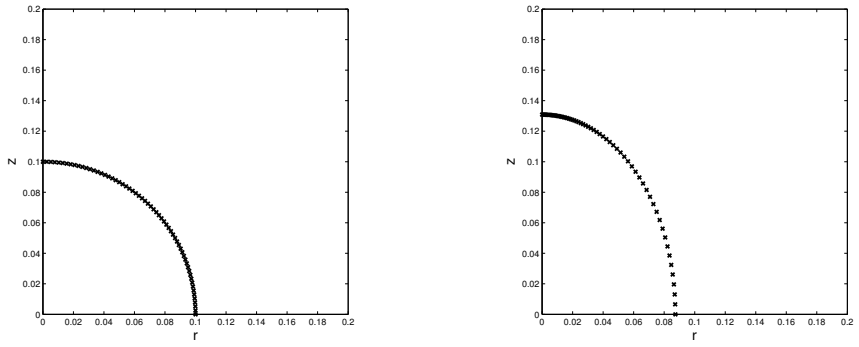


Figure 6.8: Control points at $t = 0$ (left) and $t = 1$ (right) for Example 6.4.1 without re-sampling.

6.4.4 Re-sampling of the interface

In order to reduce this effect, we assume that the interface can be represented as a function of z and propose to re-sample the interface by computing a cubic spline interpolation of the current control points at every time step. This procedure is not ideal but it works well for modeling simple interfaces. Other re-sampling procedures such as [127] will be investigated in future work.

At time t_k, our re-sampling algorithm performs the following steps:

- Use the current control points to construct the not-a-knot cubic spline $r = cs(z)$.
- Define the partition $z_i = i\frac{z_{\max}}{N}$, $i = 0, 1, ..., N$, where N is the number of control points and $z_{\max}$ is the maximum z-value reached by the control points.
- Construct the new control points $(cs(z_i), z_i)$, $i = 0, 1, ..., N$.

Example 6.4.2.

We run the simulation for $\beta = 0.5, 1$ using the same parameters as Example 6.4.1 with a coarser mesh having at most 3200 elements (obtained by locally refining a 51×51 mesh four times) and a uniform time step $\Delta t = 7 \times 10^{-3}$. We plot the deformation versus time together with results obtained by Yue *et al.* [132] in Figure 6.9 to observe excellent agreement. We also note that, despite using a much larger time step and a coarser mesh, the method does not exhibit the spurious oscillations seen in Figure 6.7. We further verify the conservation of mass property in Figure 6.10 by plotting the volume of the drop versus time to observe that the volume remains constant.

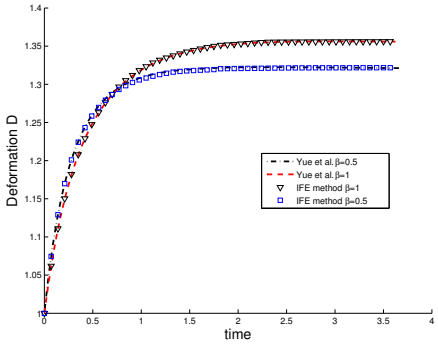


Figure 6.9: Drop deformation in extensional flow versus time for Example 6.4.2 using re-sampling.

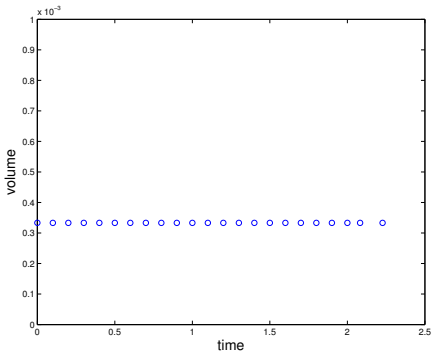


Figure 6.10: Drop volume versus time for Example 6.4.2.

Chapter 7

Conclusion and future work

7.1 Contribution

This dissertation covered two main topics: the LDG method applied to a second order elliptic problem and the immersed finite element method applied to the Stokes interface problem.

In the first part, we showed that by integrating the Dirichlet boundary conditions at Radau points the previous estimate of the velocity gradient can be improved and the optimal order of convergence is reached. If polynomials of degree at most k are used to approximate the solution gradient $\mathbf{q} = \nabla u$, the new error estimate of the LDG solution $\mathbf{q}_h$ satisfies

$$||\mathbf{q} - \mathbf{q}_h||_{0,\Omega} \leq Ch^{k+1},$$

which is an improvement of the existing bound

$$||\mathbf{q} - \mathbf{q}_h||_{0,\Omega} \leq Ch^{k+1/2}.$$

In the second part of the dissertation, we constructed IFE methods to solve the Stokes interface problem. One of the main advantages of the IFE method is that it does not require the mesh to be aligned with interfaces. Other methods that have this feature are at most second order convergent and have failed to reach a higher order approximation and some of them utilizes non-conservative schemes. On the other hand, the IFE method is conservative; and it has been extended to higher degree approximation for second order elliptic problems and following the same procedure, IFE spaces with higher approximation capabilities may be developed for the Stokes interface problem. Our work mainly consisted of developing IFE spaces and particular functions with optimal approximation capability to produce accurate numerical solutions to the Stokes interface problem. Then, we derived a discontinuous Galerkin finite element method that uses the IFE spaces and particular functions developed in Chapter 3. Numerical results for several interface problems with different interface geometries suggest that our IFE method is optimal.

Based on two different sets of jump conditions on the interface, we developed two IFE spaces $S_h(\Omega)$ and $\tilde{S}_h(\Omega)$ for the two-dimensional Stokes interface problem where we constructed local IFE spaces on arbitrary interface elements. We constructed Lagrange and null space based shape functions on interface elements and presented a general procedure to construct functions belonging to both spaces $S_h(\Omega)$ and $\tilde{S}_h(\Omega)$. Using extensive numerical simulations, we have shown that these IFE spaces have optimal approximation capabilities.

The basic properties of the space $S_h(\Omega)$ were studied and we showed the existence and uniqueness of the space for any mesh-interface configuration. We further showed that the IFE basis functions form a partition of unity and are physically consistent with the standard Lagrange IFE basis functions (*i.e.,* the two sets of basis functions are the same if $\nu^- \to \nu^+$). In an effort to investigate the global *a priori* error estimate, we also established modified trace and inverse inequalities.

We derived interior penalty IFE methods which relies on the proposed IFE spaces $S_h(\Omega)$ and $\tilde{S}_h(\Omega)$ and based on the NIPG and SIPG methods [120]. We presented several numerical results and computational examples which showed that this method leads to $O(h^2)$ and $O(h)$ optimal convergence rates, respectively, in the L^2 and broken H^1 norms. These schemes were further used to solve the Stokes interface problem with a moving interface. Two benchmark problems were treated and our results were validated by comparing them with results obtained using different methods. Our method showed that it can solve the problems using fewer degrees of freedom. We finally solved the axisymmetric Stokes interface problem by developing a new IFE space and deriving a new numerical scheme. Again, a benchmark problem with moving interface was solved and our results were validated.

7.2 Future work

There are several long term goals and future work that can follow this dissertation. Here, we present some possible extensions to our work and give ideas and perspectives on how these problems may be treated.

7.2.1 High order IFE spaces

We have numerically shown that Q_1/Q_0 IFE spaces exhibit optimal convergence rates when an arbitrary interface is interpolated using piecewise linear approximation. Developing IFE spaces that exhibit higher approximation capabilities will require:

- extending Q_k/Q_{k-1} elements to IFE elements
- additional jump constraints across the interface

• high degree approximation of the interface.

The first and second constraints might be circumvented by developing IFE spaces based on null spaces following the same procedure as in Section 3.2, and the third constraint can be satisfied by using higher degree polynomials to approximate the interface.

7.2.2 *A priori* error analysis

We have shown, numerically, that the interpolation error in the IFE space $S_h(\Omega)$ and the projection error in the IFE space $\tilde{S}_h(\Omega)$ for functions which are not smooth across the interface reach optimal order of convergence. We have further established few preliminary results which will be needed to prove *a priori* error bounds for the interpolation and projection onto the IFE spaces.

Next, we will develop *a priori* IFE error estimates for NIPG and SIPG methods using the proposed Q_1/Q_0 IFE spaces.

7.2.3 Three-dimensional interface problems

Three-dimensional interface problems provide a powerful tool to understand the fluid dynamics of complex systems. In this dissertation, we only treated the case where the system is symmetric with respect to the z-axis. A future work consists of developing IFE spaces for the three-dimensional Stokes interface problem. The main difficulty that will arise when developing these spaces is to design a procedure to approximate the three-dimensional interface, which is not a trivial task.

Furthermore, implementing a more sophisticated interface tracking procedure such as the level set and the front tracking methods will allow us to solve more complex problems.

Bibliography

[1] S. Adjerid and M. Baccouch. The discontinuous Galerkin method for two-dimensional hyperbolic problems. Part I: Superconvergence error analysis. *Journal of Scientific Computing*, 33:75–113, 2007.

[2] S. Adjerid and M. Baccouch. Asymptotically exact *a posteriori* error estimates for a one-dimensional linear hyperbolic problem. *Applied Numerical Mathematics*, 60:903–914, 2010.

[3] S. Adjerid and M. Baccouch. A superconvergent local discontinuous Galerkin method for elliptic problems. *Journal of Scientific Computing*, 52:113–152, 2012.

[4] S. Adjerid, M. Ben-Romdhane, and T. Lin. High-order interior penalty immersed finite element method for second-order elliptic interface problems. *International Journal of Numerical Analysis & Modeling*, 11:541–566, 2014.

[5] S. Adjerid, K. D. Devine, J. E. Flaherty, and L. Krivodonova. *A posteriori* error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 191:1097–1112, 2002.

[6] S. Adjerid and A. Klauser. Superconvergence of discontinuous finite element solutions for convection-diffusion problems. *Journal of Scientific Computing*, 22-23:5–24, 2005.

[7] S. Adjerid and T. Lin. A p^{th}-degree immersed finite element method for boundary value problems with discontinuous coefficients. *Applied Numerical Mathematics*, 59:1303–1321, 2009.

[8] S. Adjerid and T. C. Massey. *A posteriori* discontinuous finite element error estimation for two-dimensional hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 191:5877–5897, 2002.

[9] S. Adjerid and T. C. Massey. Superconvergence of discontinuous finite element solutions for nonlinear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 195:3331–3346, 2006.

[10] D. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21:337–344, 1984.

[11] Douglas N. Arnold and Jinshui Qin. Quadratic velocity/linear pressure stokes elements. In *Advances in Computer Methods for Partial Differential Equations VII*, pages 28–34. IMACS, 1992.

[12] C. Attanayake and D. Senaratne. Convergence of an immersed finite element method for semilinear parabolic interface problems. *Applied Mathematical Sciences*, 5(1-4):135–147, 2011.

[13] R. Ausas, F. Sousa, and G. Buscaglia. An improved finite element space for discontinuous pressures. *Computer Methods in Applied Mechanics and Engineering*, 199:1019–1031, 2010.

[14] I. Babuska. Error-bounds for finite element method. *Numerische Mathematik*, 16:322–333, 1971.

[15] I. Babuska and J. E. Osborn. Can a finite element method perform arbitrarily badly? *Mathematics of Computation*, 69:443–462, 2000.

[16] M. Baccouch. *Superconvergence and A posteriori Error Estimation for the Discontinuous Galerkin Method Applied to Hyperbolic Problems on Triangular Meshes.* Ph.D dissertation, Virginia Tech, 2008.

[17] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[18] R. Becker, E. Burman, and P. Hansbo. A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity. *Computer Methods in Applied Mechanics and Engineering*, 198:3352–3360, 2009.

[19] R. Becker, P. Hansbo, and M. Larson. Energy norm *a posteriori* error estimation for discontinuous Galerkin methods. *Computer Methods in Applied Mechanics and Engineering*, 192:723–733, 2003.

[20] M. Ben-Romdhane. *Higher-degree immersed finite elements for second-order elliptic interface problems.* Ph.D dissertation, Virginia Tech, 2011.

[21] K. S. Bey, A. Patra, and J. T. Oden. *hp*-version discontinuous Galerkin method for hyperbolic conservation laws: A parallel adaptive strategy. *International Journal of Numerical Methods in Engineering*, 38:3889–3908, 1995.

[22] R. Biswas, K. D. Devine, and J. E. Flaherty. Parallel adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.

[23] P. Bochev, C. Dohrmann, and M. Gunzberger. Stabilization of low-order mixed finite elements for the Stokes equations. *SIAM Journal on Numerical Analysis*, 44(1):82–101, 2006.

[24] J. Boland and R. Nicolaides. On the stability of bilinear-constant velocity pressure finite elements. *Numerische Mathematik*, 44(2):219–222, 1984.

[25] K. Bottcher and R. Rannache. Adaptive error control in solving ordinary differential equations by the discontinuous Galerkin method. *Tech. report, University of Heidelberg*, 1996.

[26] J. Brackbill, D. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100:335–354, 1992.

[27] J. H. Bramble and J. T. King. A finite element method for interface problems in domains with smooth boundary and interfaces. *Advances in Computational Mathematics*, 6:109–138, 1997.

[28] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, New York, 2008.

[29] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique*, 8(R2):129–151, 1974.

[30] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks/Cole, 9 edition, 2010.

[31] Z. Cai, T.A. Manteuffel, and S.F McCormick. First-order system least squares for the Stokes equations, with application to linear elasticity. *SIAM Journal on Numerical Analysis*, 34(5):1727–1741, 1997.

[32] B. Camp, T. Lin, Y. Lin, and W. Sun. Quadratic immersed finite element spaces and their approximation capabilities. *Advances in Computational Mathematics*, 24(1-4):81–112, 2006.

[33] Y. Cao and M. Gunzburger. Least-squares finite element approximations to solutions of interface problems. *SIAM Journal on Numerical Analysis*, 35(1):393–405, 1998.

[34] P. Castillo. A superconvergence result for discontinuous Galerkin methods applied to elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 192:4675–4685, 2003.

[35] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau. An *a priori* error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM Journal on Numerical Analysis*, 38:1676–1706, 2000.

[36] P. Castillo, B. Cockburn, D. Schötzau, and C. Schwab. Optimal *a priori* error estimates for the *hp*-version of the local discontinuous Galerkin method for convection-diffusion problems. *Mathematics of Computation*, 71:455–478, 2002.

[37] F. Celiker and B. Cockburn. Superconvergence of the numerical traces of discontinuous Galerkin and hybridized mixed methods for convection-diffusion problems in one space dimension. *Mathematics of computation*, 76:67–96, 2007.

[38] Y. Chang, T. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.

[39] L. Changzhi and G. Liejin. Experimental study of drop deformation and breakup in simple shear flows. *Chin. J. Chern. En*, 15:1–5, 2007.

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

[41] S. Chou, D. Kwak, and K. Wee. Optimal convergence analysis of an immersed interface finite element method. *Advances in Computational Mathematics*, 33:149–168, 2010.

[42] B. Cockburn and B. Dong. An analysis of the minimal dissipation local discontinuous Galerkin method for convection-diffusion problems. *Journal of Scientific Computing*, 32:233–262, 2007.

[43] B. Cockburn, J. Gopalakhirishnan, N. Nguyen, J. Peraire, and F. Sayas. Analysis of HDG methods for Stokes flow. *Mathematics of Computation*, 80:723–760, 2011.

[44] B. Cockburn, G. Kanschat, I. Perugia, and D. Schötzau. Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM Journal on Numerical Analysis*, 39:264–285, 2001.

[45] B. Cockburn, G. Kanshat, and D. Schotzau. A locally conservative LDG method for the incompressible Navier-Stokes equations. *Mathematics of computation*, 74(251):1067–1095, 2004.

[46] B. Cockburn, M. Luskin, C.-W. Shu, and E. Suli. Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Mathematics of Computation*, 72:577–606, 2003.

[47] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin methods for scalar conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[48] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35:2240–2463, 1998.

[49] V. Cristini, J. Blawzdziewicz, and M. Loewenberg. Drop breakup in three-dimensional viscous flows. *Physics of Fluids*, 10(8):1781–1783, 1998.

[50] L. Djamel. Advanced simulation of transient multiphase flow and flow assurance in the oil and gas industry. *The Canadian Journal of Chemical Engineering*, 91(7):1201–1214, 2013.

[51] J. Douglas and J. Wang. An absolutely stabilized finite element method for the Stokes problem. *Mathematics of Computation*, 52(186):495–508, 1989.

[52] K. Ericksson and C. Johnson. Adaptive finite element methods for parabolic problems I: A linear model problem. *SIAM Journal on Numerical Analysis*, 28:43–77, 1991.

[53] K. Ericksson and C. Johnson. Adaptive finite element methods for parabolic problems II: Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$. *SIAM Journal on Numerical Analysis*, 32:706–740, 1995.

[54] R. Falk and G. Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM Journal on Numerical Analysis*, 36:935–952, 1999.

[55] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.

[56] J. Flaherty. Finite element analysis lecture notes. *Renssealer Polythechnique Institute*.

[57] T. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84:253–304, 2010.

[58] E. Georgoulis, O. Lakkis, and J. Virtanen. *A posteriori* error control for discontinuous Galerkin methods for parabolic problems. *SIAM Journal on Numerical Analysis*, 49(2):427–458, 2011.

[59] V. Girault, B. Riviere, and M. Wheeler. A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier-Stokes problems. *Mathematics of computation*, 74:53–84, 2004.

[60] Y. Gong, B. Li, and Z. Li. Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions. *SIAM Journal on Numerical Analysis*, 46(1):472–495, 2008.

[61] K. Gupta and J. Meek. A brief history of the beginning of the finite element method. *International journal for numerical methods in engineering*, 39:3761–3774, 1996.

[62] P. Hansbo and M. Larson. Discontinuous Galerkin and the Crouzeix-Raviart element: Application to elasticity. *Mathematical modelling and numerical analysis*, 37(1):63–72, 2003.

[63] P. Hansbo, M. Larson, and S. Zahedi. A cut finite element method for a Stokes interface problem. *Applied Numerical Mathematics*, 85:90–114, 2014.

[64] X. He. *Bilinear Immersed Finite Elements For Interface Problems.* Ph.D dissertation, Virginia Tech, 2009.

[65] X. He, T. Lin, and Y. Lin. Approximation capability of a bilinear immersed finite element space. *Numerical Methods for Partial Differential Equations*, 24:1265–1300, 2008.

[66] X. He, T. Lin, and Y. Lin. A bilinear immersed finite volume element method for the diffusion equation with discontinuous coefficient. *Communications in Computational Physics*, 6(1):185–202, 2009.

[67] X. He, T. Lin, and Y. Lin. Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *International Journal of numerical analysis and modeling*, 8(2):284–301, 2011.

[68] X. He, T. Lin, Y. Lin, and X. Zhang. Immersed finite element methods for parabolic equations with moving interface. *Numerical Methods for Partial Differential Equations*, 29:619–646, 2013.

[69] P. Hessari. First-order system least squares method for the interface problem of the Stokes equations. *Computers and Mathematics with Applications.*, 68(3):309–324, 2014.

[70] R. Hooper, M. Toose, C. Macosko, and J. Derby. A comparison of boundary element and finite element methods for modeling axisymmetric polymeric drop deformation. *Intl. J. Numer. Meth. Fluids*, 37:837–864, 2001.

[71] H. Hu, N. Patankar, and M. Zhu. Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique. *Journal of Computational Physics*, 169:427–462, 2001.

[72] T. Hughes and L. Franca. A new finite element formulation for computational fluid dynamics: VII. the Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces. *Computer methods in applied mechanics and engineering*, 65:85–96, 1987.

[73] V. Isailovic, M. Obradovic, D. Nikolic, I. Saveljic, and N. Filipovic. SIFEM project: Finite element modeling of the cochlea. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pages 1–4, Nov 2013.

[74] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press New York, 1987.

[75] C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46:1–26, 1986.

[76] M. Kang, R. Fedkiw, and X. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15:323–360, 2000.

[77] M. Kang, R. Fedkiw, and D. Nguyen. A boundary condition capturing method for incompressible flame discontinuities. *Journal of Computational Physics*, 172:71–98, 2001.

[78] O. Karakashian and F. Pascal. *A posteriori* error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. *SIAM Journal on Numerical Analysis*, 41(6):2374–2399, 2003.

[79] N. Kechar and D. Silvester. Analysis of locally stabilized mixed finite element methods for the Stokes problem. *Mathematics of computation*, 58(197):1–10, 1992.

[80] R. Khayat. Three-dimensional boundary-element analysis of drop deformation for Newtonian and viscoelastic systems. *Intl. J. Numer. Meth. Fluids*, 34(3):241–275, 2000.

[81] L. Krivodonova and J. E. Flaherty. Error estimation for discontinuous Galerkin solutions of two-dimensional hyperbolic problems. *Advances in Computational Mathematics*, 19:57–71, 2003.

[82] D. Kwak and K. Wee. An analysis of a broken P_1-nonconforming finite element method for interface problems. *SIAM Journal on Numerical Analysis*, 48(6):2117–2134, 2010.

[83] M. Larson and T. Barth. *A posteriori* error estimation for adaptive discontinuous Galerkin approximations of hyperbolic systems. *Lecture Notes in Computational Science and Engineering*, 11:363–368, 2000.

[84] A. Layton and J. Beale. On the accuracy of finite difference methods for elliptic problems with interfaces. *Communications in Applied Mathematics and Computational Science*, 1:91–119, 2006.

[85] R.D. Lazrov and X. Ye. Stabilized discontinuous finite element approximations for Stokes equations. *Journal of computational and applied mathematics*, 198:236–252, 2007.

[86] P. Lesaint. *Sur la resolution des systemes hyperboliques du premier ordre par la methode des elements finis*. Ph.D dissertation, Universite Pierre et Marie Curie, Paris, 1975.

[87] P. Lesaint and P. Raviart. On a finite element method for solving the neutron transport equations. *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–123, 1974.

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

[89] R. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18:709–735, 1997.

[90] J. Li and Y. Renardy. Numerical study of flows of two immiscible liquids at low reynolds number. *SIAM Rev*, 42(3):417–439, 2000.

[91] J. Li and Y. Renardy. Shear-induced rupturing of a viscous drop in a Bingham liquid. *J. Non-Newtonian Fluid Mech.*, 95:235–251, 2000.

[92] Z. Li. *The Immersed interface method a numerical approach for partial differential equations with interfaces*. Ph.D dissertation, University of Washington, 1994.

[93] Z. Li. The immersed interface method using a finite element formulation. *Applied Numerical Mathematics*, 27:253–267, 1998.

[94] Z. Li, K. Ito, and M. Lai. An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. *Computers and fluids*, 36:622–635, 2007.

[95] Z. Li, T. Lin, Y. Lin, and R. Rogers. An immersed finite element space and its approximation capability. *Numerical Methods for Partial Differential Equations*, 20:338–367, 2004.

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

[97] Z. Li, W. Wang, I. Chern, and M. Lai. New formulations for interface problems in polar coordinates. *SIAM Journal on Scientific Computing*, 25:224–245, 2003.

[98] Q. Lin, N. Yan, and A. Zhou. An optimal error estimate of the discontinuous Galerkin method. *Journal of Engineering Mathematics*, 13:101–105, 1996.

[99] Q. Lin and A. Zhou. Convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *Acta Math. Sci.*, 13:207–210, 1993.

[100] T. Lin, Y. Lin, R. Rogers, and M. Ryan. A rectangular immersed finite element space for interface problems. *Advances in Computation: Theory and Practice*, 7:107–114, 2001.

[101] T. Lin, Y. Lin, and W. Sun. Error estimation of a class of quadratic immersed finite element methods for elliptic interface problems. *Discrete and Continuous Dynamical Systems - Series B*, 7(4):807–823, 2007.

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

[103] 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.

[104] T. Lin, D. Sheen, and X. Zhang. A locking-free immersed finite element method for planar elasticity interface problems. *Journal of Computational Physics*, 247:228–247, 2013.

[105] T. Lin and X. Zhang. Linear and bilinear immersed finite elements for planar elasticity interface problems. *Journal of Computation and Applied Mathematics*, 236:4681–4699, 2012.

[106] T. Mason and J. Bibette. Emulsification in viscoelastic media. *Phys. Rev. Lett.*, 77:3481–3484, 1996.

[107] N. Nassif and D. Khuwayri-Fayyad. *Introduction to Numerical Analysis and Scientific Computing*. Chapman and Hall/CRC, 2013.

[108] J.T. Oden, W. Wu, and M. Ainsworth. Three-step $h-p$ adaptive strategy for the incompressible Navier-Stokes equation. In I. Babuska, J.E. Flaherty, W.D. Henshaw, J.E. Hopcroft, J.E. Oliger, and T. Tezduyar, editors, *Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations, The IMA Volumes in Mathematics and its Applications*, volume 75, pages 347–366, New York, 1995. Springer.

[109] K. Ohmori and N. Saito. On the convergence of finite element solutions to the interface problem for the Stokes system. *Journal of computational and applied mathematics*, 198:116–128, 2007.

[110] J. Ottino, P. DeRoussel, S. Hansen, and D. Khakhar. Mixing and dispersion of viscous liquids and powdered solids. *Advances in Chemical Engineering*, 25:105–204, 2000.

[111] C. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.

[112] C. Peskin. Lectures on mathematical aspects of physiology. *Lectures in Appl. Math.*, 107:19–69, 1981.

[113] C. Peskin. The immersed boundary method. *Acta Numer.*, 11:479–517, 2002.

[114] C. Peskin and D. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 37:113–132, 1980.

[115] T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis*, 28:133–140, 1991.

[116] S. Ramaswamy and L. Leal. The deformation of a viscoelastic drop subjected to steady uniaxial extensional flow of a Newtonian fluid. *J. Non-Newtonian Fluid Mechanics*, 85:127–163, 1999.

[117] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. technical report la-ur-73-479. *Los Alamos Scientific Laboratory, Los Alamos*, 3, 1973.

[118] A. Reusken. Analysis of an extended pressure finite element space for two-phase incompressible flows. *Comput. Visual sci.*, 11:293–305, 2008.

[119] G. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Mathematics of Computation*, 50(181):75–88, 1988.

[120] B. Riviere. *Discontinuous Galerkin methods for solving Elliptic and parabolic equations*. Frontiers in applied mathematics, Philadelphia, 2008.

[121] B. Riviere and M. Wheeler. *A posteriori* error estimates and mesh adaptation strategy for discontinuous Galerkin methods applied to diffusion problems. *TICAM Report 00-10, University of Texas at Austin, USA*, 2000.

[122] D. Schotzau and C. Schwab. An *hp a-priori* error analysis of the DG time-stepping method for initial value problems. *Calcolo*, 37:207–232, 2000.

[123] D. Schotzau and C. Schwab. Time discretization of parabolic problems by the *hp*- version of the discontinuous Galerkin finite element method. *SIAM Journal on Numerical Analysis*, 38:837–875, 2000.

[124] M. Shimrat. Algorithm 112: Position of point relative to polygon. *Communications of the ACM*, 5:5877–5897, 1962.

[125] E. Suli and P. Houston. Finite element methods for hyperbolic problems: *A posteriori* error analysis and adaptivity. *In I. Duff and G. Watson, editors, The State of the Art in Numerical Analysis*, pages 441–471, 1997.

[126] E. Toose, B. Geurts, and J. Kuerten. A boundary integral method for two-dimensional (non)-Newtonian drops in slow viscous flow. *Non-Newtonian Fluid Mech.*, 60:129–154, 1995.

[127] H. Udaykumar, R. Mittal, and W. Shyy. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. *Journal of Computational Physics*, 153:535–574, 1999.

[128] S. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[129] L. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*, volume 1605 of *Lecture Notes in Mathematics*. Springer, 1995.

[130] C. Wu, Z. Li, and M. Lai. Adaptive mesh refinement for elliptic interface problems using the non-conforming immersed finite element method. *International Journal of Numerical Analysis & Modeling*, 8(3):466–483, 2011.

[131] P. Yue, J. Feng, C. Liu, and J. Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. *Journal of Fluid Mechanics*, 515:293–317, 2004.

[132] P. Yue, C. Zhou, J. Feng, C. Ollivier-Gooch, and H. Hu. Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing. *Journal of Computational Physics*, 219:47–67, 2006.

[133] X. Zhang. *Nonconforming Immersed Finite Element Methods for Interface Problems*. Ph.D dissertation, Virginia Tech, 2013.

[134] A. Zhou and Q. Lin. Optimal and superconvergence estimates of the finite element method for a scalar hyperbolic equation. *Acta Math. Sci.*, 14:90–94, 1994.

[135] H. Zhou and C. Pozrikidis. The flow suspension in channels: Single files of drops. *Physics of Fluids*, A(5):311–324, 1993.

[136] O. Zienkiewicz and R. Taylor. *The finite element method: Fluid mechanics.*, volume 3. Butterworth Heinemann Oxford, 2000.