UCLA
UCLA Electronic Theses and Dissertations

Title
Virtual Node Algorithms for Stokes Interface Problems

Permalink
https://escholarship.org/uc/item/0m70g8xz

Author
Assencio, Diego

Publication Date
2012

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Los Angeles

Virtual Node Algorithms for Stokes Interface Problems

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Physics

by

Diego C. Assêncio

2012

© Copyright by

Diego C. Assêncio

2012

Abstract of the Dissertation

Virtual Node Algorithms for Stokes Interface Problems

by

Diego C. Assêncio

Doctor of Philosophy in Physics

University of California, Los Angeles, 2012

Professor Giovanni Zocchi, Chair

We present two numerical methods for the solution of the Stokes equations designed to handle both interfacial discontinuities, geometrically irregular flow domains and discontinuous fluid properties such as viscosity and density. The methods are efficient, easy to implement and yield second order accurate, discretely divergence free velocities. We call these methods Virtual Node Algorithms. The first method handles the case in which the fluid viscosity is continuous across the interfaces, while the second method handles the case in which the fluid viscosity is discontinuous across the interfaces. In both cases, we assume the fluid viscosity to be uniformly constant over the spatial extension of each fluid. We discretize the Stokes equations using an embedded approach on a uniform MAC-grid employing virtual nodes at interfaces and boundaries. Interfaces and boundaries are represented with a hybrid Lagrangian/level set method. For the continuous viscosity case, we rewrite the Stokes equations as three Poisson equations and use the techniques developed in Bedrossian et al. (2010) [1] to impose jump and boundary conditions. We also use a final Poisson equation to enforce a discrete divergence-free condition. All four linear systems involved are symmetric positive definite with three of the four having the standard 5-point Laplace stencil everywhere. Numerical results are presented indicating second order accuracy in L^∞ for both velocities and pressure. For the discontinuous viscosity case, we presented a method which requires no knowledge of the jumps on the fluid variables and their derivatives along the interface. The degrees of freedom associated with virtual

nodes allow for accurate resolution of discontinuities in the fluid stress at the interfaces but require a Lagrange multiplier term to enforce continuity of the fluid velocity. We provide a novel discretization of this term that accurately resolves the constant pressure null modes. The discrete coupled equations for the velocity, pressure and Lagrange multipliers are in the form of a symmetric KKT system. Numerical results are presented indicating second order accuracy for the velocities and first order accuracy for the pressure (in L^∞).

The dissertation of Diego C. Assêncio is approved.

Eric D'Hoker

Jianwei Miao

Joseph Teran

Giovanni Zocchi, Committee Chair

University of California, Los Angeles

2012

Table of Contents

1 Introduction **1**

1.1 Existing methods 5

2 Stokes problem: continuous fluid viscosity **10**

2.1 Reformulation of the Stokes equations: Poisson 10

2.1.1 Interface conditions 12

2.1.2 Irregular domain boundary conditions 14

2.2 Description of numerical method 14

2.2.1 Computation of the level set from Γ_h 16

2.2.2 Computation of jump conditions at $\hat{\Gamma}^h$ 20

2.2.3 Construction of ∇p 23

2.2.4 Projection onto divergence-free space 26

2.2.5 Interface advection 28

3 Stokes problem: discontinuous fluid viscosity **29**

3.1 Description of numerical method 29

3.1.1 Computation of the level set from Γ_h 32

3.1.2 Transfer of interfacial forces to Eulerian fluid grid 35

3.1.3 Eulerian discretization details 35

3.1.4 Interface advection 52

4 Numerical results **53**

4.1 Convergence measure 53

4.2 Examples with continuous fluid viscosity 55

4.2.1 Elastic interface discretization 55
4.2.2 Surface tension 57
4.2.3 Irregular domain 58
4.3 Examples with discontinuous fluid viscosity 59
4.3.1 Jacobi preconditioned MINRES 59
4.3.2 Elastic interface discretization 60
4.3.3 Surface tension 60
4.3.4 Pressure null mode test 61
4.3.5 Rising drop 62

5 Summary and discussion 70

A Derivation of pressure boundary conditions 72

B Equivalence between original and modified Stokes problems 74

C Integration of polynomials over polygonal domains/curves 76

References 78

List of Figures

1.1 Fluid domains and interface 3

2.1 Discretization of fluid domains and interface 11

2.2 Construction of the level set from the discrete interface Γ_h 17

2.3 Perturbation of the nodes of Γ_h 18

2.4 Segment-cell collision detection 19

2.5 Computation of the level set on the doubly-fine grid nodes 20

2.6 Setting the jump conditions on $\hat{\Gamma}_h$ 21

2.7 Computation of ∇p 24

2.8 Lagrangian approximation to zero isocontour 26

2.9 Stencil of the divergence matrix $\mathbf{D}$ 28

3.1 Discretization of fluid domains and interface 30

3.2 MAC grid sub-grids 31

3.3 Original MAC grid and computation of the level set values 33

3.4 Determination of cut u and v cells from $\hat{\Gamma}_h$ 37

3.5 Duplication of nodes 39

3.6 Nodes used for computing the elemental stiffness matrix 46

3.7 Construction of positively oriented subcell boundaries 49

3.8 Expressions for computing $K_{w_1}^{c^p}$ 50

3.9 Discrete stencils away from the interface 51

4.1 Evolution of interface for Example 4.2.1 64

4.2 Evolution of interface for Example 4.2.2 65

4.3 Evolution of marker particles for Example 4.2.3 66

4.4 Evolution of interface for Example 4.3.2 67

4.5 Evolution of interface for Example 4.3.3 68

4.6 Evolution of interface for Example 4.3.5 69

A.1 Local orthogonal coordinates on a surface point 72

List of Tables

4.1 Orders of convergence for Example 4.2.1. 57

4.2 Orders of convergence for Example 4.2.2. 58

4.3 Orders of convergence for Example 4.2.3. 59

4.4 Orders of convergence for Example 4.3.2. 61

4.5 Orders of convergence for Example 4.3.3. 61

4.6 Average number of MINRES iterations per time step for Example 4.3.4 . . 62

4.7 Average run time per time step (in seconds) for Example 4.3.4 62

Acknowledgments

I must thank above all my father, Eduardo, my mother, Cristina, my sister, Bruna and everyone else in my family for all the love and emotional support they gave me during my Ph.D. I would like to wholeheartedly thank my wife, Franziska, without whom I would hardly have succeeded in crossing the often cold waters of this academic endeavour. I would also like to thank a great teacher and friend, Lauro, whose help was crucial for the success of my acceptance in the Physics Graduate Program at UCLA. He is undoubtedly one of the best mentors I have ever had.

I would like to thank especially my roommates from college: Thiago and Lucas. The time we spent together left on me an intelectual imprint without which I might not have been capable of tackling the research projects I worked on. Also, I would like to thank all my other friends from college who motivated me to pursue a career abroad and, sometimes through harsh but honest criticism, improved the quality of my character and enlighted me with thoughts and ideas I have found to be of extreme value during my Ph.D. experience.

My thesis adviser Joseph Teran has been a great source of knowledge. I found his ability to quickly develop solutions to difficult problems to be truly impressive. Joining his group was certainly the best decision I made during my graduate studies.

I would also like to thank the members of my committee, Giovanni Zocchi, Jianwei Miao and Eric D'Hoker, for the time and effort they put on reading my thesis.

Finally, I would like to thank James von Brecht and Jacob Bedrossian for teaching me the basic aspects of the Virtual Node Algorithms. They were always very approachable every time I found myself in need of help.

Vita

2002-2003	National Council for Scientific and Technologic Development Scholarship (Conselho Nacional de Desenvolvimento Científico e Tecnológico) University of São Paulo, São Paulo, Brazil
2004-2006	São Paulo Research Foundation Scholarship (Fundação de Amparo à Pesquisa do Estado de São Paulo) University of São Paulo, São Paulo, Brazil
2006	B.S. Physics and Mathematics (minor in Numerical Analysis and Biosciences) University of São Paulo, São Paulo, Brazil
2007-2012	Physics Division Fellowship University of California, Los Angeles, California
2008	Physics and Astronomy Outstanding Teaching Award University of California, Los Angeles, California
2009	Summer Research Program Fellowship University of California, Los Angeles, California
2009	Physics and Astronomy Outstanding Teaching Award University of California, Los Angeles, California
2010	Physics and Astronomy Outstanding Teaching Award University of California, Los Angeles, California

CHAPTER 1

Introduction

Interface problems are present in a vast array of applications in the Natural Sciences and in Engineering. They occur naturally, for instance, when different materials interact through surfaces which are very thin when compared to their other relevant physical dimensions. Due to the often complicated geometry of the contact interfaces and of the material domains, the discontinuity in material properties across the interfaces and the possible freedom of the interface to undergo motion, these problems are very difficult to tackle. Analytic tools are in general insuficient to solve problems of this type, making numerical simulations the best available ways to study and solve them in practice. However, the challenges imposed by this type of problem cannot be dealt with standard numerical methods since they are not designed to account for such interfacial discontinuities. In this dissertation, the author will describe his work with Prof. Joseph Teran on a new class of methods, which will be referred to as Virtual Node Agorithms (VNA), that were designed to numerically solve problems of this type on the context of fluid flow.

We consider the Stokes equations for two-phase, incompressible flow in irregular domains:

$$\nabla \cdot \sigma = \mu \Delta \mathbf{u} - \nabla p = -\mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma \tag{1.1}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega \setminus \Gamma \tag{1.2}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega \tag{1.3}$$

$$[\mathbf{u}](\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in \Gamma \tag{1.4}$$

$$[\sigma \cdot \mathbf{n}](\mathbf{x}) = \mathbf{f}^i, \quad \mathbf{x} \in \Gamma \tag{1.5}$$

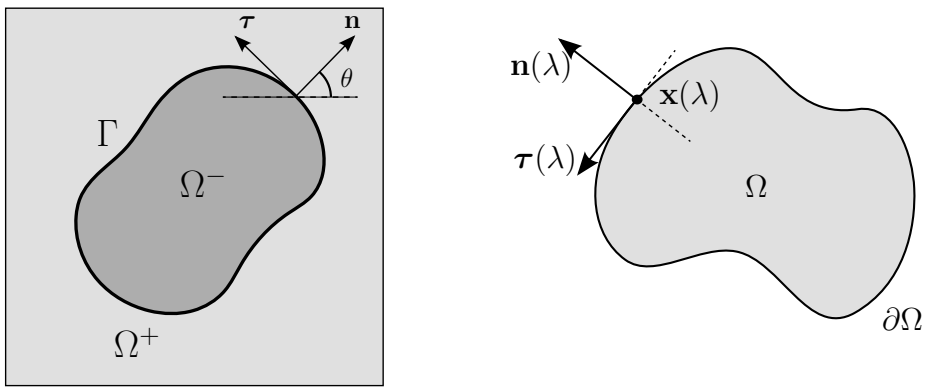
Here p is the pressure, $\mathbf{u} = (u, v)$ is the fluid velocity, μ is the fluid viscosity (assumed uniform over the extension of each fluid), $\sigma = \mu\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}}^T\right) - p\mathbf{I}$ is the fluid stress tensor, Γ is the interface between the two phases, $\mathbf{f}$ is the body force density and $\mathbf{f}^i$ is the force per unit length supported on the interface between the two phases (e.g. surface tension). The interface Γ is a codimension one closed curve that divides the domain into an interior region Ω^- and an exterior region Ω^+ such that $\Omega = \Omega^- \cup \Omega^+ \cup \Gamma$ (see Figure 1.1(a)). We let $\mathbf{n}$ denote the outward unit normal to Ω^- at a point $\mathbf{x} \in \Gamma$ and define $[v](\mathbf{x}) := v^+(\mathbf{x}) - v^-(\mathbf{x}) = \lim_{\epsilon \to 0^+} v(\mathbf{x} + \epsilon\mathbf{n}) - \lim_{\epsilon \to 0^+} v(\mathbf{x} - \epsilon\mathbf{n})$ as the "jump" of the quantity v across the interface Γ. Unless otherwise stated, we assume the curves Γ and $\partial\Omega$ are smooth ($\partial\Omega$ will also be taken to be a rectangular curve for interface problems).

The Stokes equations describe fluid flow with very small Reynolds number $\mathrm{Re} = UL\rho/\mu$, where U is a characteristic velocity of the flow, L is a characteristic length of the system and ρ is the fluid density. As shown in [2], they can be derived directly from the the Navier-Stokes equations,

$$\rho \frac{D\mathbf{u}}{Dt} = \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = \nabla \cdot \sigma + \mathbf{f}, \tag{1.6}$$

when we take the limit $\mathrm{Re} \to 0$. Flows with small Reynolds numbers are characterized by either small velocities, high viscosities or, most commonly, small body sizes. In this type of regime, the advective intertial force term $\rho\frac{D\mathbf{u}}{Dt}$ becomes negligible when compared with the other force terms. Examples of fluid flow of this type include swimming of microorganisms and flow of viscous polymers.

We introduce in this dissertation two second order VNAs for approximating the incompressible two-phase Stokes equations with irregular embedded interfaces and boundaries on a uniform Cartesian MAC-grid. The first algorithm was developed to tackle the problem when the viscosity μ is the same inside and outside the interface Γ. The second algorithm deals with the case in which the fluids inside and outside the interface have different viscosities. In other words, these algorithms were developed to handle the cases $[\mu] = 0$ and $[\mu] \neq 0$ respectively across Γ. In both algorithms, we use regular grids because it simplifies the implementation, permits straightforward numerical linear algebra and achieves higher



(a) (b)

Figure 1.1: (a) Interface Γ separates the fluid domain $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$ into its exterior Ω^+ and interior Ω^-. The figure shows the unit normal and tangent vectors $\mathbf{n}$ and $\boldsymbol{\tau}$. If θ is the angle between $\mathbf{n}$ and the horizontal x axis, then $\mathbf{n} = (\cos\theta, \sin\theta)$ and $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$. (b) Irregular domain Ω. The vectors $\mathbf{n}(\lambda)$ and $\boldsymbol{\tau}(\lambda)$ are the outward normal and the positively oriented tangent vectors of $\partial\Omega$ at the point $\mathbf{x}(\lambda)$.

order accuracy in L^∞. Our approach uses duplicated Cartesian grid cells along the interface to introduce additional "virtual" nodes that accurately account for the lack of regularity.

When the viscosity of both fluid phases is the same, we can formulate the Stokes problem as three Poisson equations with jump conditions. This allows us to directly use the VNAs developed by Bedrossian et al. in [1] (generalized to three dimensions by Hellrung et al. in [3]). Their methods solve Poisson problems with interfaces:

$$-\nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \backslash \Gamma \tag{1.7}$$

$$[\mathbf{u}] = a(\mathbf{x}), \quad \mathbf{x} \in \Gamma \tag{1.8}$$

$$[\beta(\mathbf{x})\nabla u] \cdot \mathbf{n} = b(\mathbf{x}), \quad \mathbf{x} \in \Gamma \tag{1.9}$$

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_d \tag{1.10}$$

$$\beta(\mathbf{x})\nabla u(\mathbf{x}) \cdot \mathbf{n} = h(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_n, \tag{1.11}$$

where $\partial\Omega = \partial\Omega_d \cup \partial\Omega_n$. Their methods are second order accurate, geometrically flexible and easy to implement. Also, they discretize the equations (1.7-1.11) on a uniform Cartesian grid, making the coupling between our methods with theirs straightforward to implement. The spatial uniformity of the fluid viscosity implies, as will be shown in Chapter 2, that the coefficient $\beta(\mathbf{x})$ is a constant for each of the Poisson equations that need to be solved, simplifying the overall work even further.

Our choice of staggering the variables (velocities and pressure) on a MAC-grid requires separate Poisson discretizations for each variable. We also solve a final Poisson equation over the pressure grid to enforce a discrete divergence-free condition yielding a total of four Poisson solves per Stokes solve. All four linear systems involved are symmetric positive definite. In all but the final Poisson equation, our approach yields the standard 5-point stencil almost everywhere (notably, we have the standard 5-point stencil across the interface between the phases). This desirable property is directly associated with our choice of using the methods of [1] to solve them. The VNA which results from this reformulation of the Stokes equations is outlined in Chapter 2. Numerical results indicating second order accuracy in L^∞ for both velocities and the pressure are presented in Chapter 4.

If the fluid viscosities are different, the aforementioned approach fails to be directly

useful since the interfacial discontinuities on the velocities and the pressure become coupled in a complicated manner, making the solutions of the Poisson equations for each of them very difficult to obtain. We describe a VNA developed to tackle this kind of problem in Chapter 3. In this case, the discretization of the Stokes equations as well as the divergence-free and jump constraints are conducted through a variational formulation which yields symmetric numerical stencils. We use Lagrange multipliers to enforce the jump conditions weakly. Results for this algorithm indicating second order accuracy in L^∞ for both velocities and first order accuracy in L^∞ for the pressure are also presented in Chapter 4.

We represent the interfaces by Lagrangian particles for straightforward interface advection, however the techniques described in this dissertation require a level set representation of interfaces and boundaries. We provide methods (which are slightly different for the each of the VNAs presented) for transforming a Lagrangian interface into a level set which is meaningful for both the velocities and pressure grids. This ensures that the discrete interface conditions are enforced consistently for the staggered variables.

1.1 Existing methods

Our methods are second order accurate in L^∞ for both embedded boundary conditions on irregular flow domains and for embedded interfacial discontinuities in two-phase flows. This is achieved with relatively simplistic linear algebra demands. Specifically, for the continuous viscosity case, all linear systems are symmetric positive definite and have discrete stencils equal to that of the standard 5-point Laplacian discretization almost everywhere. The linear systems for the discontinuous viscosity case are symmetric indefinite and can be solved using standard fast Krylov subspace methods such as MINRES. Both VNA described in this dissertation yield discretely divergence-free velocities. Although many researchers have developed embedded methods for the Stokes equations with interfacial discontinuities and irregular domains, to the best of our knowledge our approaches are the first to support this feature set. In our discussion of existing approaches, we will focus only on embedded (or immersed) methods that avoid unstructured meshing when addressing boundary and

interface conditions at irregular geometric boundaries.

Embedded techniques use a computational domain that simply encompasses rather than geometrically adheres to the irregular domain. A good review of embedded methods is given by Lew et al. in [4]. They point out that these techniques originated with the papers of Harlow and Welch [5] and Charles Peskin [6]. Peskin developed the Immersed Boundary Method (IBM) to simulate blood flow in the heart [7, 8, 9, 10], but it has also been applied to problems such as aquatic animal locomotion, [11], platelet adhesion and aggregation during blood clotting [12, 13], swimming of microorganisms in viscoelastic Stokesian fluids [14], peristaltic pumping [15], oscillating filaments on viscous fluids [16] and many others [17, 18, 19, 20, 21]. A summary of the development of the Immersed Boundary Method and its applications can be found in [22]. Despite its vast popularity and considerable ease of implementation, the IBM suffers from its use of regularized delta functions to represent singular forces acting on interfaces. This renders the method first order accurate and implies that the physical characteristics of the flow near the interfacial boundaries are not accurately captured. Singular forces acting on the interface impose discontinuities in the pressure, the velocities and their derivatives which the IBM may fail to accurately resolve [23]. However, for sufficiently smooth problems in which the interfaces are thick instead of infinitesimally thin the IBM can achieve second order accuracy [24, 25]. Adaptive versions of the IBM were developed in [26] to enhance convergence over coarse grids but the results were still only first order accurate. Another deficiency of the IBM is poor conservation of volume near the interface. The seriousness of this problem, especially for the simulation of blood flow in the heart, motivated the development of a better volume conserving version of the IBM in [27]. However, this version of the IBM only works for periodic domains. Also, the stencils of its discrete gradient and divergence operators are much larger than those of the original IBM.

Attempts to address the lower order of accuracy of the IBM resulted in a series of other methods. For instance, the Blob-Projection Method presented in [28] is effectively a higher order version of the IBM. However, methods of the type just mentioned still fail to sharply capture the discontinuities created by the singular forces across interfaces since they yield,

similarly to the IBM, velocities and pressures which are continuously differentiable across them.

Many methods were designed to improve the order of accuracy of the original IBM. The Immersed Interface Method (IIM) is perhaps the most popular example of this. The IIM was first developed for elliptic equations with interfacial discontinuities [29] and later applied to Stokes flow [30]. The IIM achieves second (and higher) order accuracy by capturing interfacial discontinuities in the pressure, the velocities and their derivatives in a sharp manner. The IIM has been used in many fluid flow problems including interface and rigid boundary problems [23, 31, 32, 33, 34, 35], Hele-Shaw flow [36] and also problems in which the viscosity is discontinuous across the interfaces [37]. Arbitrarily high orders of accuracy have been achieved [38, 39]. The method is considerably more difficult to implement than the IBM and most applications are in two space dimensions as a result. However, researchers have applied the IIM to three dimensional flows [40]. For more about the IIM, we refer the reader to a review of its applications in [41].

A limitation of the IIM is the lack of symmetry in discretizations arising from problems with discontinuous coefficients. This imposes an obstacle on the overall speed of these methods since fast linear solvers such as conjugate gradients cannot be used. However, it should be noted that the IIM can yield symmetric matrices for continuous viscosity Stokes flow.

One method that is capable of always guaranteeing a symmetric discretization is the Ghost Fluid Method (GFM). Initially applied to the Poisson equation with interfacial jumps and variable coefficients [42], the GFM was used to simulate multimaterial compressible inviscid flows [43] and also multiphase incompressible flows [44]. Unfortunately, the GFM is only capable of achieving first order results for interface problems.

Some of the first embedded methods were fictitious domain methods by Hyman [45] and Saul'ev [46]. The fictitious domain approach has been used with incompressible materials in a number of works [47, 48, 49, 50, 51, 52, 53, 54, 55]. These approaches embed the irregular geometry in a more simplistic domain for which fast solvers exist (e.g. Fast

Fourier Transforms). The calculations include fictitious material in the complement of the domain of interest. A forcing term (often from a Lagrange multiplier) is used to maintain boundary conditions at the irregular geometry. Although these techniques naturally allow for efficient solution procedures, they depend on a smooth solution across the embedded domain geometry for optimal accuracy, which is not typically possible.

The Extended Finite Element Method (XFEM) and related approaches in the finite element literature also make use of geometry embedded in regular elements. Although originally developed for crack-based field discontinuities in elasticity problems, these techniques are also used with embedded problems in irregular domains. Daux et al. first showed that these techniques can naturally capture embedded Neumann boundary conditions [56, 57]. These approaches are equivalent to the variational cut cell method of Almgren et al. in [58]. Enforcement of Dirichlet constraints is more difficult with variational cut cell approaches [59, 4] and typically involves a Lagrange multiplier or stabilization. Dolbow and Devan recently investigated the convergence of such approaches with incompressible materials and point out that much analysis in this context remains to be completed [60]. Despite the lack of thorough analysis, such XFEM approaches appear to be very accurate and have been used in many applications involving incompressible materials in irregular domains [61, 62, 63, 64, 65].

There are also many finite difference (FDM) and finite volume methods (FVM) that utilize cut uniform grid cells. Many of these methods have been developed in the context of incompressible flow. For example, Almgren et al. use cut uniform bilinear cells to solve the Poisson equation for pressures in incompressible flow calculations [58]. Marelle et al. use collocated grids and define sub cell interface and boundary geometry in cut cells via level sets [66]. Ng et al. also use level set descriptions of the irregular domain and achieve second order accuracy in L^∞ for incompressible flows [67]. The approach of Batty and Bridson is similar, but not as accurate [68]. Cut cell FDM and FVM have also been developed for incompressible and nearly incompressible elastic materials. Bijelonja et al. use cut cell FVM to enforce incompressibility more accurately than is typically seen with FEM [69]. Beirão da Veiga et al. use polygonal FVM cells to avoid remeshing with irregular domains [70].

Barton et al. [71] and Hill et al. [72] use cut cells with Eulerian elastic/plastic flows. Other examples of FVM can be found in [73] and [74].

A series of other second order cartesian grid methods have also been developed to tackle interface and irregular boundary problems. For instance, as examples of other methods which capture interfaces and boundaries in a sharp manner we should mention ghost cell methods [75] and boundary integral methods [76], the last one being based on a Poisson solver for irregular domain problems developed in [77]. Some methods are capable of producing results which are only partially second order accurate for fluid flow problems. As an example, see [78] for a method which was used to simulate two dimensional fluid flow using second-order FDM/FVM discretizations for the streamfunction-vorticity equations.

CHAPTER 2

Stokes problem: continuous fluid viscosity

2.1 Reformulation of the Stokes equations: Poisson

As mentioned in Chapter 1, our method for the Stokes problem with continuous fluid viscosity is designed to leverage the advances in a recent paper by Bedrossian et al [1]. We facilitate this by formulating our discretization in terms of three Poisson equations. We will first cover the derivation of these Poisson equations from the two-phase Stokes equations in irregular domains (1.1). The solutions $\mathbf{u}$ and p may have discontinuities across the interface Γ [30]. We therefore first consider each subdomain Ω^+ and Ω^- separately to avoid complications associated with these discontinuities. Taking the divergence of both sides of equation (1.1) and noting that $\nabla \cdot \mathbf{u} = 0$ from the incompressibility condition, we get the following Poisson equation:

$$\nabla \cdot (\mu \Delta \mathbf{u} - \nabla p + \mathbf{f}) = 0 \implies \Delta p = \nabla \cdot \mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma \tag{2.1}$$

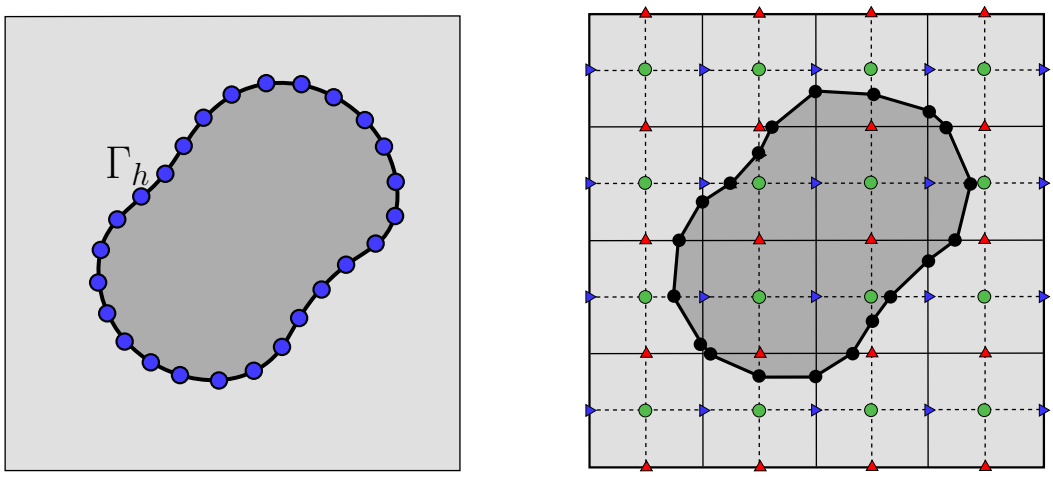
This divergence is rigorously defined for all points $\mathbf{x} \in \Omega$ on either side of the interface ($\mathbf{x} \in \Omega \setminus \Gamma = \Omega^- \cup \Omega^+$) but not for points $\mathbf{x}$ on the interface. After solving this Poisson equation for the pressure p, we can solve another two Poisson equations for the velocity components u and v using the solution for obtained for p:

$$\Delta p = \nabla \cdot \mathbf{f}, \quad \mathbf{x} \in \Omega \setminus \Gamma \tag{2.2}$$

$$\mu \Delta u = p_x - f^1, \quad \mathbf{x} \in \Omega \setminus \Gamma \tag{2.3}$$

$$\mu \Delta v = p_y - f^2, \quad \mathbf{x} \in \Omega \setminus \Gamma. \tag{2.4}$$

Here, f^1 and f^2 are the horizontal and vertical components of the body force density field $\mathbf{f}$. These equations hold on the interior of the domain and away from the interface. We



(a) (b)

Figure 2.1: (a) Discrete Lagrangian interface Γ_h. (b) Fluid variables are discretized over an Eulerian MAC grid. A level set representation of the interface is defined over the nodes of a "doubly-fine" sub grid. We approximate the zero isocontour of the level set with piecewise linear $\hat{\Gamma}_h$. The black dots represent the intersection of the interface $\hat{\Gamma}_h$ and the edges of each doubly-fine grid cell.

must therefore derive boundary and interface conditions in terms of the body forces $\mathbf{f}$ and the interfacial forces $\mathbf{f}^i$ to solve these equations in practice.

2.1.1 Interface conditions

We assume that all fluids have the same viscosity μ. In this case, the discontinuities in u, v and p are decoupled from each other [37] and the three Poisson equations above can truly be solved separately. Let $\mathbf{x}(\lambda, t)$ be an arbitrary parametrization of the curve Γ. We can express the jump conditions in u, v and p in terms of the force per unit length of the parametrization parameter λ. We will use $\mathbf{F}(\lambda, t)$ to denote this parameterization dependent force density. Recall that the jump in a quantity w at a point $\mathbf{x}(\lambda, t)$ of the interface is expressed as:

$$[w](\mathbf{x}) := w^+(\mathbf{x}) - w^-(\mathbf{x}) = \lim_{\epsilon \to 0^+} w(\mathbf{x} + \epsilon \mathbf{n}) - \lim_{\epsilon \to 0^+} w(\mathbf{x} - \epsilon \mathbf{n}) \tag{2.5}$$

where $\mathbf{n}$ is the outward unit normal of Ω^- at the point $\mathbf{x}$. If we let $\boldsymbol{\tau}(\lambda, t)$ denote the positively oriented unit tangent to the interface Γ at the point $\mathbf{x}$, then

$$\begin{aligned} \mathbf{n} &= (\cos\theta, \sin\theta) \\ \boldsymbol{\tau} &= (-\sin\theta, \cos\theta), \end{aligned} \tag{2.6}$$

where θ is the positively oriented angle between $\mathbf{n}$ and the x axis (see Figure 1.1(a)). The normal and tangential components of the force density $\mathbf{F}(\lambda, t)$ are then:

$$\begin{aligned} F_n(\lambda, t) &:= \mathbf{F}(\lambda, t) \cdot \mathbf{n}(\lambda, t) \\ F_\tau(\lambda, t) &:= \mathbf{F}(\lambda, t) \cdot \boldsymbol{\tau}(\lambda, t). \end{aligned} \tag{2.7}$$

We can use these conventions to express the jump conditions on u, v, p across the interface Γ [23, 34, 79]:

$$[p](\mathbf{x}(\lambda,t)) = F_n(\lambda,t)\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|^{-1} \tag{2.8}$$

$$[u](\mathbf{x}(\lambda,t)) = 0 \tag{2.9}$$

$$[v](\mathbf{x}(\lambda,t)) = 0 \tag{2.10}$$

$$\left[\frac{\partial p}{\partial n}\right](\mathbf{x}(\lambda,t)) = \frac{\partial}{\partial \lambda}\left(F_\tau(\lambda,t)\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|^{-1}\right)\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|^{-1} \tag{2.11}$$

$$\mu\left[\frac{\partial u}{\partial n}\right](\mathbf{x}(\lambda,t)) = F_\tau(\lambda,t)\sin\theta\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|^{-1} \tag{2.12}$$

$$\mu\left[\frac{\partial v}{\partial n}\right](\mathbf{x}(\lambda,t)) = -F_\tau(\lambda,t)\cos\theta\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|^{-1}, \tag{2.13}$$

Despite the explicit appearance of the parametrization function $\mathbf{x}(\lambda,t)$, the jumps defined above are independent of the chosen parametrization in the sense that, if $\tilde{\mathbf{x}}(\tilde{\lambda},t)$ also parametrizes Γ, then all the jumps above are the same at each point $\tilde{\mathbf{x}}(\tilde{\lambda},t) = \mathbf{x}(\lambda,t)$.

Equations (2.9) and (2.10) can together be interpreted as expressing the no-slip and no-penetration conditions that make the fluid surrounding the interface stick to but not travel through it [80]. The interface points move then with the local fluid velocities (which are uniquely defined since the no-slip and no-penetration conditions make u and v continuous along Γ):

$$\frac{\partial \mathbf{x}}{\partial t} = (u(\mathbf{x},t), v(\mathbf{x},t)). \tag{2.14}$$

Equation (2.8) states that the normal component of the interfacial force is balanced by a pressure jump across the interface, while Equations (2.12) and (2.13) state that the tangential component of the interfacial force is balanced by a jump on the fluid shear forces acting on the interface.

2.1.2 Irregular domain boundary conditions

We must also know appropriate boundary conditions for u, v and p to solve Poisson equations (2.2), (2.3) and (2.4) in practice. We will assume that Dirichlet boundary conditions are know for the velocities along $\partial\Omega$. Thus for all $\mathbf{x} \in \partial\Omega$ we will have $u(\mathbf{x}) = U(\mathbf{x})$ and $v(\mathbf{x}) = V(\mathbf{x})$ for some functions $U(\mathbf{x})$ and $V(\mathbf{x})$ defined only over $\partial\Omega$. Let $\mathbf{x}(\lambda)$ be a parameterization of $\partial\Omega$, let $\mathbf{n}(\lambda)$ be the unit outward normal vector to $\partial\Omega$ at the point $\mathbf{x}(\lambda)$ and let $\boldsymbol{\tau}(\lambda)$ be the unit tangential vector to $\partial\Omega$ at that same point (see Figure 1.1(b)). The Neumann boundary conditions for the pressure are then (a derivation of this fact is given in Appendix A):

$$\nabla p \cdot \mathbf{n} = \mathbf{f} \cdot \mathbf{n} - \frac{\partial \omega}{\partial \tau} \quad \text{for } \mathbf{x} \in \partial\Omega, \tag{2.15}$$

where $\omega = v_y - u_x$ is the fluid vorticity and $\partial\omega/\partial\tau = \nabla\omega \cdot \boldsymbol{\tau}$ is the tangential component of the gradient of ω at $\mathbf{x} \in \partial\Omega$. It is shown in Appendix B that the solutions of the original and the reformulated versions of the Stokes problem are the same if the boundary conditions are of the Dirichlet type for the velocities and of the Neumann type for the pressure as in equation (2.15). Naturally, we need to have Dirichlet boundary conditions for $\mathbf{u} = (u, v)$ which preserve the total fluid volume inside of the domain Ω:

$$\int_{\partial\Omega} (U, V) \cdot \mathbf{n} \, dl = 0 \tag{2.16}$$

2.2 Description of numerical method

The method couples a discrete Lagrangian representation of the interface (Γ_h) with a background Eulerian representation of fluid velocities and pressures. The Lagrangian interface moves with the local fluid velocity (as in equation (2.14)) and the temporal discretization is explicit. The fluid variables are stored on a staggered MAC grid [81] to facilitate enforcement of a discrete divergence-free condition. We use the VNA presented in [1] to solve the Poisson interface problems (2.2), (2.3) and (2.4) over the respective sub-grids in the MAC grid. The interface is represented with Lagrangian particles and interfacial forces are naturally defined at the particle locations. However, the VNA was designed for a level set

representation of the interface geometry and jump conditions. We provide a procedure for defining a level set representation of the interface geometry and boundary conditions from the Lagrangian particles. The overall procedure for advancing one time step is:

1. Compute interface forces (from surface tension or elasticity) at all particles in Γ_h; details in Sections 4.2.1 and 4.2.2.

2. Compute the level set and transfer jump conditions from the Lagrangian interface Γ_h; details in Section 2.2.1.

3. Use VNA to solve $\Delta p = \nabla \cdot \mathbf{f}$ with interface conditions (2.8, 2.11) and Neumann boundary conditions (2.15); details in Section 2.2.2.

4. Use VNA to solve $\mu \Delta u = -p_x + f^1$ with interface conditions (2.9, 2.12); details in Sections 2.2.2 and 2.2.3.

5. Use VNA to solve $\mu \Delta v = -p_y + f^2$ with interface conditions (2.10, 2.13); details in Sections 2.2.2 and 2.2.3.

6. Project the velocity field $\mathbf{u} = (u, v)$ to enforce the divergence-free condition; details in Section 2.2.4.

7. Interpolate velocities from the MAC grid to the Lagrangian interface and update particle positions using forward Euler; details in Section 2.2.5.

In the last step, the x and y components of the velocity on a given interface node $\mathbf{x}_i$ are interpolated from the u and v velocities respectively determined in steps 4 and 5. This interpolation is done using the conventions presented in [1] for defining quantities on the interface.

We will now describe each of these steps in detail. We give only limited discussion of the computation of the Lagrangian interface forces $\mathbf{f}^i$ (Sections 4.2.1 and 4.2.2) and the interpolation from the MAC grid to the Lagrangian interface (Section 2.2.5) as these aspects are relatively straightforward.

2.2.1 Computation of the level set from Γ_h

A Lagrangian representation of the interface is convenient for computing interfacial forces and also for discretizing the motion of the interface. We represent the discrete interface Γ_h with a sequence of points $\mathbf{x}_i$ where $i \in \{0, 1, 2, \ldots M-1\}$. The points are connected by segments and form a closed curve as shown in Figure 2.1(a) (we also support multiple closed curves). This curve is used to divide the domain into the discrete interior and exterior domains Ω_h^- and Ω_h^+ respectively (see Figures 2.1(b) and 2.2). In order to use the VNA, we need to define a level set over the MAC grid that corresponds with Γ_h. Let N denote the number of MAC grid cells per dimension and let h be the length of a cell edge (we assume x and y edges have the same length, see Figure 2.1(b)). The level set is defined on the nodes of a doubly-fine grid with $2N \times 2N$ cells per direction (see Figure 2.1(b)). We use the doubly-fine grid because it naturally defines a discrete level set on the u, v and p subgrids. An alternative is to compute a level set with N cells per dimension on each of the respective subgrids, however such a choice would complicate the accurate computation of p_x and p_y needed for the u and v Poisson solves (see Section 2.2.3 for a more detailed discussion of these issues). The level set is computed as a signed distance function on the nodes of cells intersecting Γ_h. Nodes that are not incident on a cell intersecting Γ_h are set to positive or negative $h/2$ depending on whether the node is inside Ω_h^+ or Ω_h^- respectively. The computation of the narrow band level set is negligible compared to the time taken to solve the linear systems with the VNA.

In order to simplify the process of determining which interface segments in Γ_h intersect which grid cell edges, we first ensure that each segment on the interface is smaller than the cell width $h/2$ of the doubly-fine grid. This can be done by adding nodes $\tilde{\mathbf{x}}_i$ to subdivide its segments (see Figure 2.2). This is only for the computation of nodal level set values; the computation of the interfacial forces uses the original set of segments Γ_h. We will denote this subdivided discrete interface as $\tilde{\Gamma}_h$. This subdivision simplifies the determination of which doubly-fine grid cells intersect which segments in Γ_h. We also perturb the nodes $\tilde{\mathbf{x}}_i$ of $\tilde{\Gamma}_h$ to prevent them from falling directly on the edges of any cell (see figure 2.3). Specifically,

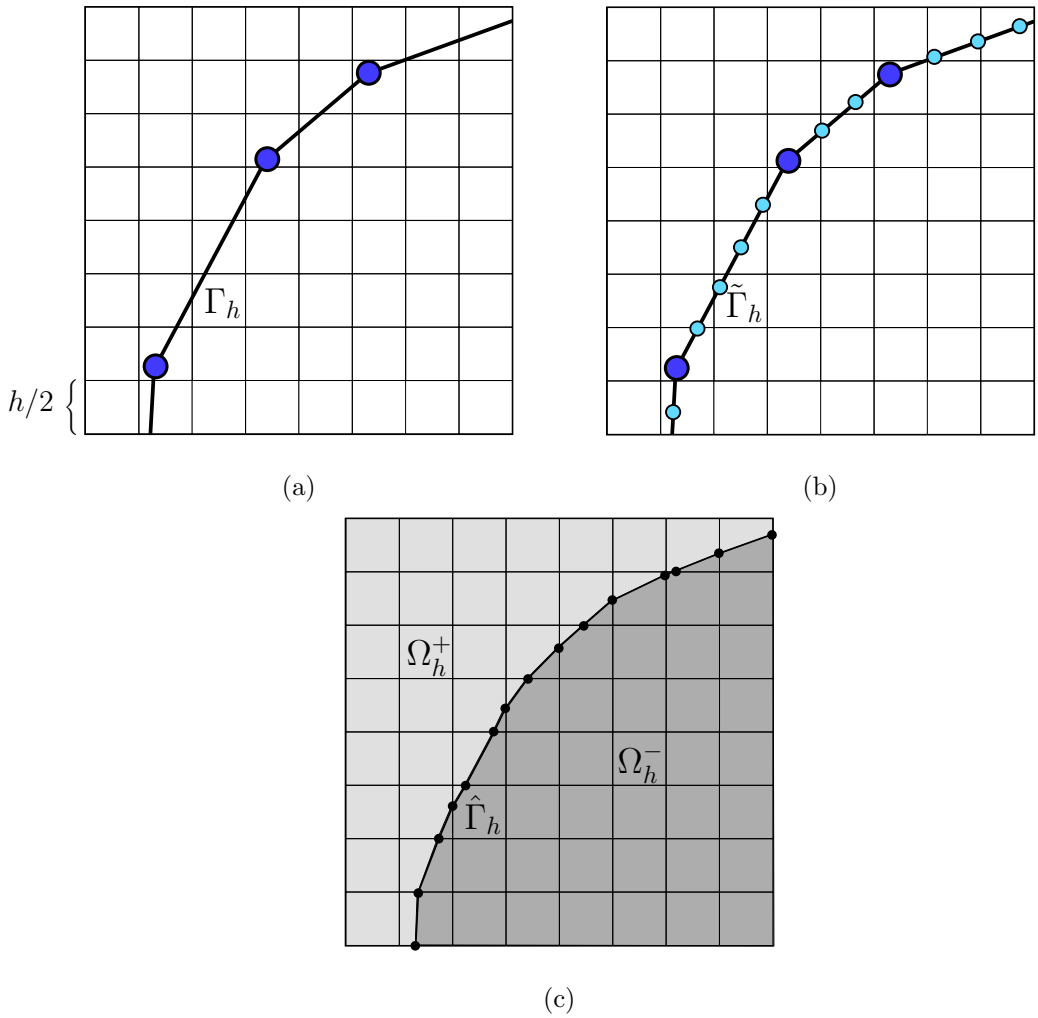


Figure 2.2: Figure (a) shows the original discrete Lagrangian interface Γ_h with nodes $\mathbf{x}_i$ (blue circles). Figure (b) shows the subdivided $\tilde{\Gamma}_h$. The added nodes ($\tilde{\mathbf{x}}_i$) are indicated with smaller light blue circles. Nodes are added until all segments of $\tilde{\Gamma}_h$ are shorter than the cell width $h/2$ of the doubly-fine grid. Figure (c) shows the Lagrangian approximation $\hat{\Gamma}_h$ of the discrete interface Γ_h which is generated by the level set computed from Γ_h. The black circles denote the positions at which $\hat{\Gamma}_h$ crosses the edges of each cut cell. The domains Ω_h^- and Ω_h^+ are defined from the level set and $\hat{\Gamma}_h$. The Lagrangian curve $\hat{\Gamma}_h$ is used in the VNA [1] for quadrature purposes and interface conditions must be transferred from Γ_h (where they are naturally defined) to $\hat{\Gamma}_h$.

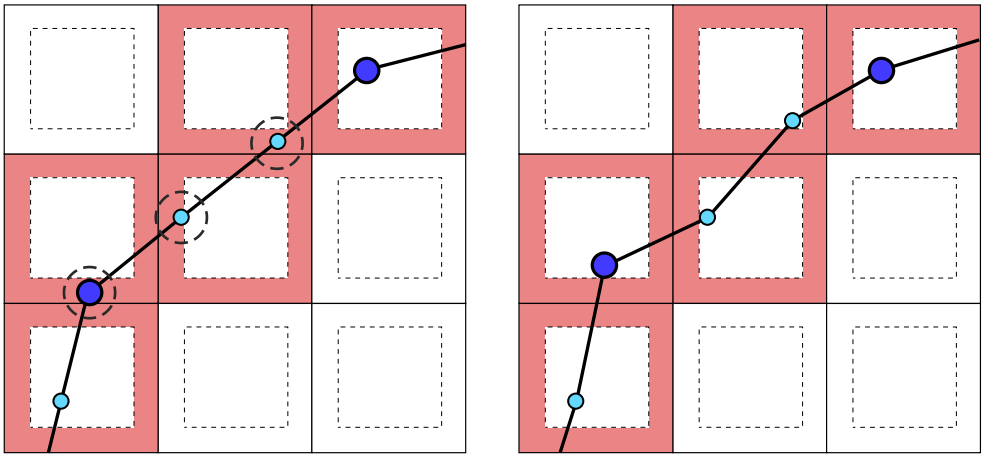


Figure 2.3: The picture on the left shows a piece of the interface $\tilde{\Gamma}_h$ with three (circled) nodes lying too close to the edges of the cells they fall into. The picture on the right shows the perturbation applied to these nodes. The shaded regions depict the thresholds used for determining if a node is too close to a grid cell edge. Note that the size of these regions is exaggerated for ease of visualization. In practice they are set to have a width proportionate to $h/2$.

for a given $\tilde{\mathbf{x}}_i$, we determine which cell in the doubly-fine grid contains it. We then clamp this node toward the cell center in a dimension by dimension fashion until it is at least as far as some tolerance αh away from the edges of the cell (typically we use $\alpha = 1\mathrm{e}^{-6}$). Once the nodes are perturbed, we can unambiguously determine which segments in $\tilde{\Gamma}_h$ intersect which edges in the doubly-fine grid. Cell edges cut an even number of times are considered to be uncut. All possible cases are illustrated in Figure 2.4(a). In order to avoid ambiguities, cases (4) and (7) require special treatment. We treat case (4) as a subcase of case (3) and also case (7) as a subcase of case (6). For example, consider Figure 2.4(b). In case (4) the node A is within a small tolerance of the segment PQ, P is in the lower-left cell and Q is in the upper-right cell. Here, we automatically consider edges AC and AB as intersecting edge PQ. Otherwise if node A is sufficiently far from edge PQ we explicitly compute intersections between edge PQ and edges AC, AB, AE and AD to determine if we are in case (5) or case (3).

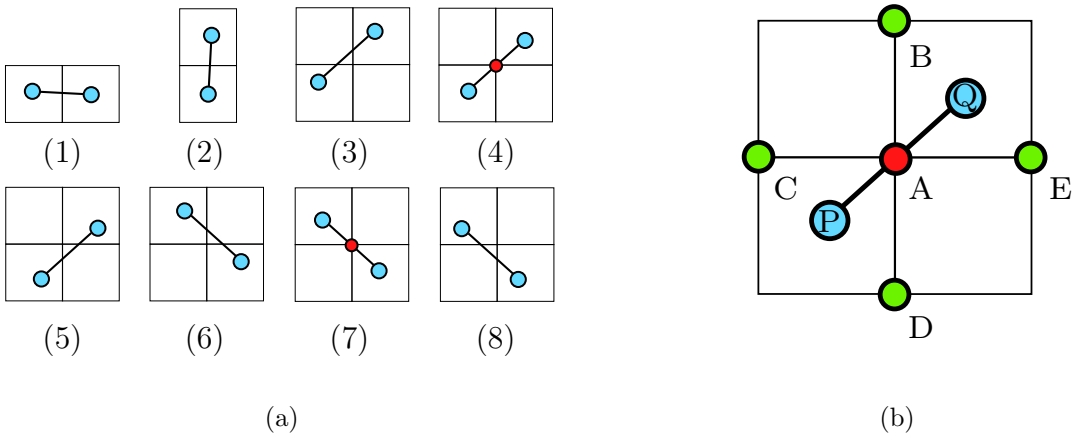


Figure 2.4: Segment-cell collision detection. Figure (a) shows the eight cases to be considered when determining which cell edges are cut by a given segment of the discrete interface $\tilde{\Gamma}_h$. Cases (4) and (7) happen when a segment intersects four cells simultaneously by crossing a corner node (shown as small red circles in the figure). Figure (b) shows a case which requires special attention.

Once we have determined which subdivided interface segments in $\tilde{\Gamma}_h$ intersect which grid cell edges, we compute the exact distance to $\tilde{\Gamma}_h$ for all grid nodes incident on a cut cell. That is, for all grid nodes with at least one of its four incident cells cut by a segment in $\tilde{\Gamma}_h$, we compute the analytic distance from that node to each of the segments that intersect any of its incident cells and define the minimum as the distance from the grid node to $\tilde{\Gamma}_h$. We set the distance to be a large positive constant for all grid nodes not incident on a cut cell. Finally, we determine the sign of the level set value depending on whether or not the nodes they correspond to are inside or outside $\tilde{\Gamma}_h$ (see figure 2.5). This is done with a flood-fill approach. The lower left node of the domain is defined to be outside the domain. We then sweep through the nodes in a dimension-by-dimension fashion. The sign of the next node in the sweep is the same as the previous node if the edge connecting them is crossed an even number of times, otherwise it is given the opposite sign.

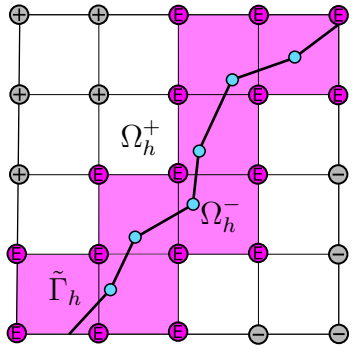


Figure 2.5: The level set values are computed exactly on the corner nodes of doubly-fine cells which are crossed by the interface $\tilde{\Gamma}_h$. These cut cells are shaded in pink, their nodes are labeled with an "E". The level set values are set to $h/2$ on the nodes labeled with a "+" and to $-h/2$ on the nodes labeled with a "−". The "+" and "−" nodes lie on the side of Ω_h^+ and Ω_h^- respectively.

2.2.2 Computation of jump conditions at $\hat{\Gamma}^h$

The VNA requires a description of the jump conditions along the embedded interface. It suffices to define these conditions at the center of each segment on a cell-wise Lagrangian approximation to the zero isocontour of the level set. We call this additional Lagrangian curve $\hat{\Gamma}_h$. $\hat{\Gamma}_h$ is an approximation to the zero isocontour of the level set (see Figure 2.2). It is determined by connecting points on the doubly-fine grid cell edges where the level set interpolates to zero. The VNA uses $\hat{\Gamma}_h$ for quadrature purposes and the accuracy required in the definition of the jump conditions is related to this. Unfortunately, the jump and boundary conditions needed for equations (2.2), (2.3) and (2.4) are naturally defined at the nodes of the original Lagrangian interface Γ_h. We provide a procedure to transfer these conditions from the Lagrangian nodes of Γ_h to the centers of the segments on $\hat{\Gamma}_h$ (see Figure 2.6).

Recall that we denote the points on the ends of the segments in the curve Γ_h as $\mathbf{x}_i$ with $i = 0, 1, \ldots, M-1$. If there are m doubly-fine grid cells cut by the zero isocontour of the levelset, let $\mathcal{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{m-1}\}$ denote the set of segment centers on $\hat{\Gamma}_h$ (see Figure 2.6). Assume we have computed the jump conditions at the points $\mathbf{x}_i$ of Γ_h. For each

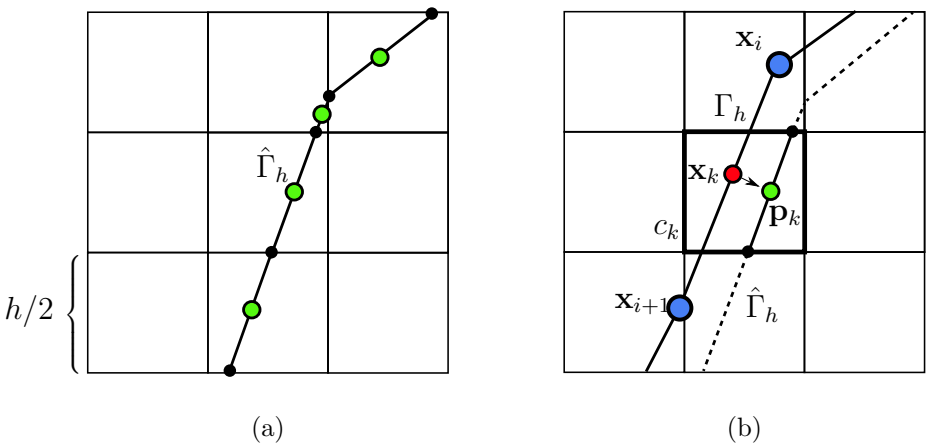


Figure 2.6: Setting the jump conditions. (a): In the VNA, the jump conditions (Equations (2.8 -2.13)) must be defined at the centers of the segments on the cell-wise Lagrangian approximation to the zero isocontour of the level set. These cell-wise segment centers are shown in light-green large circles above. The black smaller circles represent points were the level set interpolates to zero for cut cells (these points define the cell-wise Lagrangian approximation to the zero-isocontour). Figure (b): Jump conditions are naturally computed at the nodes $\mathbf{x}_i$ of the discrete interface Γ_h and then transferred to the points $\mathbf{p}_k$ through the points $\hat{\mathbf{x}}_k$. Each point $\mathbf{p}_k$ lies at the center of the $\hat{\Gamma}_h$ segment on the doubly-fine cut cell c_k. The jump at $\hat{\mathbf{x}}_k$ is computed by linearly interpolating the values at the nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Note that the above separation between the level set generated interface $\hat{\Gamma}_h$ and the discrete interface Γ_h is exaggerated for illustrative purposes.

$k = 0, 1, \ldots, m-1$, let $\hat{\mathbf{x}}_k$ be the point on Γ_h closest to $\mathbf{p}_k$ (see Figure 2.6). We linearly interpolate the jump condition at $\hat{\mathbf{x}}_k$ from the values at the ends of the segment to which it belongs. This condition is then defined to be the jump at $\mathbf{p}_k$.

In the VNA [1], we impose the jump conditions for u, v and p over each cut cell of their respective grids in an integral average sense. For that, we need to represent the discrete velocities and the pressure in terms of the bilinear functions over each grid node of their grids. Each node i on the u, v and p grids (including duplicated ones) have bilinear functions $N_i^u(\mathbf{x})$, $N_i^v(\mathbf{x})$ and $N_i^p(\mathbf{x})$ associated to them. Notice, however, that the total number of nodes after the duplication is performed might be different for each grid (we duplicate the nodes of each grid independently as described in Section 2.2.1), meaning the index i has different ranges for u and v nodes. The discrete versions of u, v and p are then:

$$u_h(\mathbf{x}) = \sum_{i=1}^{n_u} u_i N_i^u(\mathbf{x}), \tag{2.17}$$

$$v_h(\mathbf{x}) = \sum_{i=1}^{n_v} v_i N_i^v(\mathbf{x}), \tag{2.18}$$

$$p_h(\mathbf{x}) = \sum_{i=1}^{n_p} p_i N_i^p(\mathbf{x}), \tag{2.19}$$

for all $\mathbf{x} \in \Omega_h$. Above n_u, n_v and n_p are the total number of nodes (including duplicates ones) on the u, v and p grids respectively. The discrete jump conditions are expressed as below:

$$\int_{c_l^u \cap \hat{\Gamma}_h} [u_h(\mathbf{x})] \; dl = \sum_{i=1}^{n_u} u_i \Theta_i^u \int_{c_l^u \cap \hat{\Gamma}_h} N_i^u(\mathbf{x}) \; dl = \int_{c_l^u \cap \hat{\Gamma}_h} a^u(\mathbf{x}) \; dl \tag{2.20}$$

$$\int_{c_l^u \cap \hat{\Gamma}_h} [\nabla u_h(\mathbf{x}) \cdot \mathbf{n}] \; dl = \sum_{i=1}^{n_u} u_i \Theta_i^u \int_{c_l^u \cap \hat{\Gamma}_h} \nabla N_i^u(\mathbf{x}) \cdot \mathbf{n} \; dl = \int_{c_l^u \cap \hat{\Gamma}_h} b^u(\mathbf{x}) \; dl \tag{2.21}$$

where $c_l^u \cap \hat{\Gamma}_h$ is the portion of $\hat{\Gamma}_h$ on the lth cut cell (c_l^u) of the u grid, $a^u(\mathbf{x})$ is the jump in $u(\mathbf{x})$ and $b^u(\mathbf{x})$ is the jump in the normal derivative of $u(\mathbf{x})$. Similar expressions for $[v_h(\mathbf{x})]$ and $[p_h(\mathbf{x})]$ enforce the discrete jumps on v_h and p_h on the cut cells of the v and p grids respectively. Above $\Theta_i^u = +1$ if the ith node on the u grid is associated with a cut cell of Ω_h^+, $\Theta_i^u = -1$ if it is associated with a cut cell of Ω_h^- and $\Theta_i^u = 0$ for all other nodes; Θ_i^v and Θ_i^p are defined in an equivalent manner for the nodes of the v and p grids respectively.

Expressions for line integrals over polygonal curves needed to compute the integrals of the bilinear functions on equations (2.20) and (2.21) ($\nabla N_i^u(\mathbf{x}) \cdot \mathbf{n}$ is a linear function, hence also bilinear) are given in Appendix C.

Let L_k denote the length of the segment of $\hat{\Gamma}_h$ on the doubly-fine cut cell c_k. We can approximate the integrals of the jump conditions on the right hand side of equations (2.20) and (2.21) using the values at the segment centers ($a(\mathbf{p}_k)$ and $b(\mathbf{p}_k)$) and the segment lengths (L_k):

$$\int_{c_l^u \cap \hat{\Gamma}_h} a(\mathbf{x}) dS \approx \sum_{c_k \cap c_l^u} a(\mathbf{p}_k) L_k \tag{2.22}$$

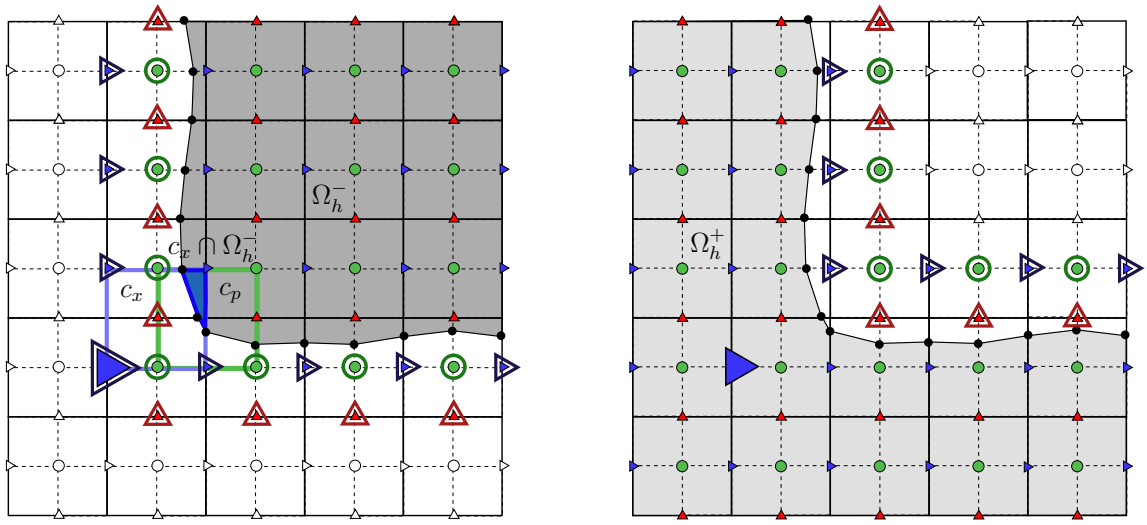
$$\int_{c_l^u \cap \hat{\Gamma}_h} b(\mathbf{x}) dS \approx \sum_{c_k \cap c_l^u} b(\mathbf{p}_k) L_k, \tag{2.23}$$

where the sum takes into account only the doubly-fine cells c_k which intersect the cell c_l^u.

2.2.3 Construction of ∇p

The VNA uses duplicated grid cells along the interface to introduce additional nodal degrees of freedom. These degrees of freedom allow for the more accurate solution of the Poisson interface problems. The MAC grid structure we employ therefore requires this duplication procedure on each of the u, v and p subgrids. See Figure 2.7 for an illustration of this process. The duplication must be done in a manner that coordinates duplication of the three subgrids. Specifically, the u, v and p degrees of freedom interact through the right hand side terms in Poisson equations for u and v (where p_x and p_y respectively appear). Therefore the duplication process must admit a procedure for the computation of the respective components of ∇p on each of the duplicated grids for u and v. This is complicated by the use of virtual p degrees of freedom in the ∇p stencils on cut cells. Fortunately, we will show here that the definition of the level set over the doubly-fine grid admits a consistent duplication and subsequent ∇p transfer procedure.

The duplication procedure we advocate is trivially defined by inheriting the doubly-fine level set to each of the u, v and p subgrids. We first define the Lagrangian zero isocontour on the doubly-fine grid ($\hat{\Gamma}_h$) as described in Section 2.2.2. This gives a piecewise linear



(a) (b)

Figure 2.7: A typical case where the computation of ∇p is not trivial. The colored nodes in figure (a) are either virtual (shown with an extra border outline) or material nodes of the u, v and p grids associated with Ω_h^-, while the colored nodes on figure (b) are the equivalent ones for Ω_h^+. The large blue triangle corresponds to a virtual node on the Ω_h^- u grid. The original node is shown enlarged on the right in Ω_h^+. Notice that simple methods such as centered differences cannot be used to compute p_x numerically on the node on the Ω_h^- side since there is no p node of Ω_h^- on its left. Note that we highlight the cells c_x and c_p as well as the region $c_x \cap \Omega_h^-$ in the image at the left to aid in the discussion from Section 2.2.3.

approximation to the zero isocontour over each cut cell in the doubly-fine grid. Then, for each subgrid we define the Lagranagian cell-wise approximation to the zero isocontour as the union of the piecewise linear approximations over the four grid cells in the doubly-fine grid that are contained in the subgrid (see Figure 2.8 for illustration). Also, a u, v or p cell is considered cut (and then subsequently duplicated) if any of its four sub doubly-fine grid cells are cut. Note that this is a slightly different criteria than that used in the original VNA [1]. There, a grid cell was considered cut if any of its four nodal level set values were of differing sign.

The staggering of the variables on the MAC grid naturally facilitates a second order centered finite difference stencil for p_x at each u degree of freedom and for p_y at each v

degree of freedom. Not coincidentally, these are precisely the source terms needed in the Poisson equations for u and v respectively. Unfortunately, these stencils are not always well defined for interfacial u and v degrees of freedom. For example consider the left subfigure in Figure 2.7. Here, a centered difference computation of p_x at the virtual u node (drawn as a large blue triangle pointing to the right) is not possible because there is no p degree of to the left.

Fortunately, the VNA requires only cell-wise averages of the right had side terms of the Poisson equations (see Section 4 in [1] for more details). Notably we will never explicitly require components of ∇p at virtual u and v nodes like the one node highlighted in the left subfigure in Figure 2.7. The required cell-wise averages are:

$$\overline{p_{h,x}} = \int_{c_x \cap \Omega_h^{\pm}} p_{h,x} \, dA \quad \text{and} \quad \overline{p_{h,y}} = \int_{c_y \cap \Omega_h^{\pm}} p_{h,y} \, dA. \tag{2.24}$$

Here, c_x and c_y are cells in the duplicated u and v grids respectively. $c_x \cap \Omega_h^{\pm}$ is the intersection of the fluid domain with c_x (similar for $c_y \cap \Omega_h^{\pm}$). Note that these regions are non-trivial for cells in the duplicated grids that intersect the interface. That is, these regions correspond to cut cells. An example of such a cut cell is shown incident to the enlarged u node in left subfigure of Figure 2.7. While we do not have a p degree of freedom to the left of this node, we do have enough information to compute $\overline{p_{h,x}}$ over the cut cell. Specifically, we have enough information to determine a piecewise bilinear approximation to p over $c_x \cap \Omega_h^{-}$. Note that there is a p cell (c_p) containing region $c_x \cap \Omega_h^{-}$ in Figure 2.7. This implies that we can approximate p as bilinear over the required region. In general, there will be at most two p cells overlapping any $c_x \cap \Omega_h^{\pm}$ or $c_y \cap \Omega_h^{\pm}$ region and we can always use this to generate a piecewise bilinear approximation to p wherever needed.

The aforementioned piecewise bilinear representation of p is merely the one given in equation (2.19) restricted to points $\mathbf{x} \in c_x \cap \Omega_h^{\pm}$ or $c_y \cap \Omega_h^{\pm}$. It can be directly used to compute the derivatives of p with respect to x and y:

$$p_{h,x}(\mathbf{x}) = \sum_{i=1}^{N_p} p_i N_{i,x}^p(\mathbf{x}) \quad \text{and} \quad p_y(\mathbf{x}) = \sum_{i=1}^{N_p} p_i N_{i,y}^p(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega_h. \tag{2.25}$$

The cell-wise averages ($\overline{p_{h,x}}$ and $\overline{p_{h,y}}$) of $p_{h,x}$ and $p_{h,y}$ can then be computed exactly as in

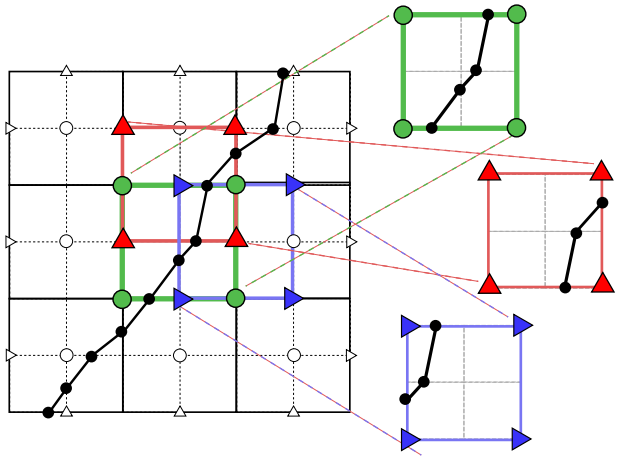


Figure 2.8: Inheriting the Lagrangian zero isocontour from the doubly-fine grid. The image depicts the definition of the piecewise linear approximation to the zero isocontour over u, v and p grid cells. For each case, the grid cell is considered cut if any of the four sub grid cells are cut. u grid cells are shown in blue with triangular nodes pointing to the right, the v grid cells are show in red with triangular nodes pointing upwards and the p grid cells are shown in green with circular nodes.

[1] (see also Appendix C). This is easily done because the integrands in

$$\overline{p_{h,x}} = \sum_{i=1}^{n_p} p_i \int_{c_x \cap \Omega_h^{\pm}} N_{i,x}^p \, dA \quad \text{and} \quad \overline{p_{h,y}} = \sum_{i=1}^{n_p} p_i \int_{c_y \cap \Omega_h^{\pm}} N_{i,y}^p \, dA \tag{2.26}$$

are piecewise bilinear. That is, we need only integrate a low order polynomial in x and y over the polygonal cut cell regions $c_x \cap \Omega_h^{\pm}$ and $c_y \cap \Omega_h^{\pm}$.

2.2.4 Projection onto divergence-free space

When we use the Poisson formulation (equations (2.3) and (2.4)) for the discretization of u and v, we get second order (in L^∞) accurate velocities. However, they do not in general satisfy a discrete divergence-free condition. While any consistent approximation of the divergence will converge to zero under refinement with the u and v we generate from the Poisson equations, it is often advantageous to enforce a discrete divergence-free condition. We satisfy this divergence-free condition via projection and our numerical experiments

suggest that this process does not degrade the L^∞ accuracy.

The divergence-free condition is defined over each cell in a p-node centered grid. For every p node (including virtual p nodes created in the duplication procedure), we define the p-centered grid cell ($\mathcal{C}_p$) to consist of the four surrounding sub cells in the doubly-fine grid (see Figure 2.9). These cells are also duplicated whenever the p node at its center is virtual. We enforce the following conditions at all p-centered grid cells $\mathcal{C}_p$:

$$\begin{aligned}\int_{\mathcal{C}_p \cap \Omega_h^\pm} \nabla \cdot \mathbf{u}_h \; dA &= \int_{\mathcal{C}_p \cap \Omega_h^\pm} (u_{h,x} + v_{h,y}) \; dA \\ &= \int_{\mathcal{C}_p \cap \Omega_h^\pm} \left[\sum_{i=1}^{n_u} u_i N_{i,x}^u + \sum_{i=1}^{n_v} v_i N_{i,y}^v \right] \; dA = 0.\end{aligned} \tag{2.27}$$

Note that this defines a linear set of constraints on the u_h and v_h degrees of freedom. If we define $\mathbf{U}$ to be the vector of degrees of freedom of u_h, $\mathbf{V}$ to be the vector of degrees of freedom of v_u, and $\mathbf{W}$ to be the vector of all nodal velocity degrees of freedom:

$$\mathbf{W} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \end{pmatrix}, \tag{2.28}$$

then we can symbolically represent the linear constraints in equations (2.27) as $\mathbf{DW} = \mathbf{0}$. The limited support of the bilinear functions N_i^u and N_i^v implies that $\mathbf{D}$ is a sparse matrix (see Figure 2.9). Since the number of cells $\mathcal{C}_p$ with pressure nodes in their centers is the same as the total number of pressure nodes n_p, then $\mathbf{D} \in \mathbb{R}^{n_p \times (n_u + n_v)}$. Note that the evaluation of the entries in $\mathbf{D}$ must be done using the cut-cell integration of low order polynomials over polygons as discussed in Section 2.2.3. The projection of the $\mathbf{W}$ obtained by solving the discrete equations (2.3) and (2.4) can then be performed as:

1. Solve $\mathbf{DD}^T\hat{\mathbf{P}} = \mathbf{DW}$, where $\hat{\mathbf{P}} \in \mathbb{R}^{n_p}$
2. $\mathbf{W} \leftarrow \mathbf{W} - \mathbf{D}^T\hat{\mathbf{P}}$.

The linear system $\mathbf{DD}^T\hat{\mathbf{P}} = \mathbf{DW}$ is symmetric positive definite and can be solved with the conjugate gradient method.

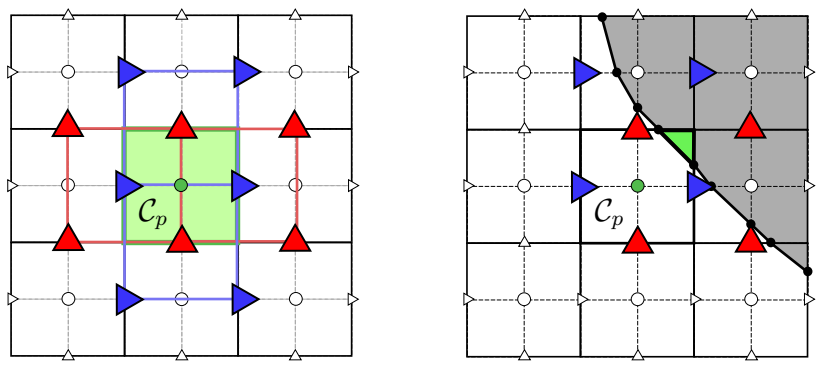


Figure 2.9: The stencil of the matrix $\mathbf{D}$ for the divergence-free projection algorithm proposed involves only the colored nodes on the figures above (blue triangles pointing to the right are nodes of the u grid; red triangles pointing up are nodes of the v grid). Far away from the interface (left figure), the stencil encompasses a maximum of 12 points (6 from the u grid and 6 from the v grid). For cut cells, fewer u and v values are necessary to represent the stencil (figure on the right).

2.2.5 Interface advection

At the end of each time step, we advect the nodes of the interface Γ_h using forward Euler to compute the interfacial forces at the next time step. To accomplish this we must interpolate u and v values from the MAC grid to each node $\mathbf{x}_i$ of Γ_h. This must be computed using virtual and material degrees of freedom associated with either Ω_h^+ or Ω_h^- but not both as in [1]. For that, we determine the original cells on the u and v grids at which $\mathbf{x}_i$ lies and use bilinear interpolation to compute the velocity values at $\mathbf{x}_i$ using only degrees of freedom from Ω_h^+ or Ω_h^-. The choice of region is in principle arbitrary since Γ_h and $\hat{\Gamma}_h$ approximate each other. In practice, however, we verify explicitly if $\mathbf{x}_i$ falls into Ω_h^+ or Ω_h^- by directly verifying whether or not it is in the interior of $\hat{\Gamma}_h$. Using degrees of freedom of u and v associated with both sides of $\hat{\Gamma}_h$ would decrease the order of accuracy of our results since these quantities are not continuously differentiable across $\hat{\Gamma}_h$.

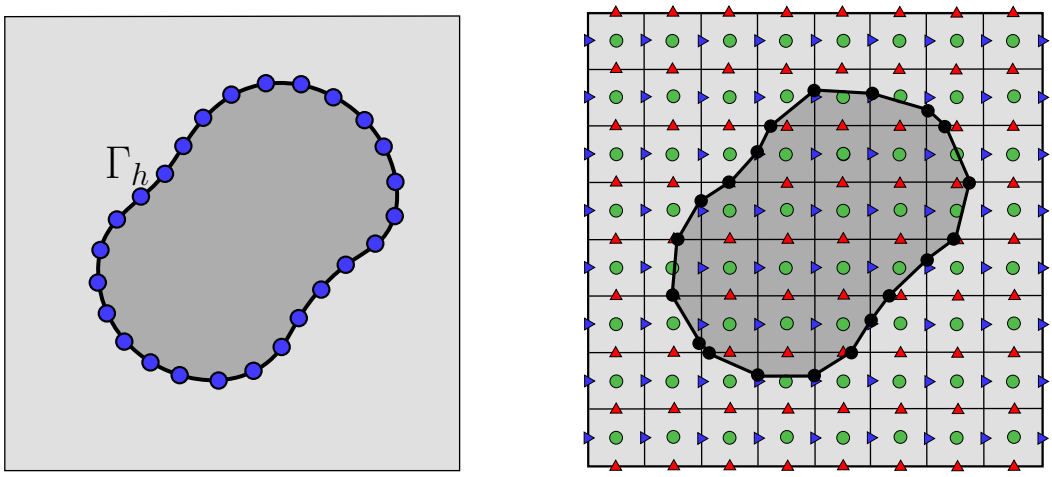
CHAPTER 3

Stokes problem: discontinuous fluid viscosity

For two-phase incompressible Stokes problems in which the fluid viscosities are different, the approach of Chapter 2 is not useful since the jumps on the velocities, the pressure and their normal derivatives across the interface given in Equations (2.8- 2.13) are no longer valid. Instead, the jumps across the interface are coupled in a complicated way [37]. This led us to developed a VNA to tackle this kind of problem through a variational formulation of the Stokes equations (1.1- 1.5). The jump and divergence-free conditions are enforced weakly.

3.1 Description of numerical method

As in the previous VNA, we couple a Lagangian representation of the interface (Γ_h) with an Eulerian representation of the fluid velocity and pressure (see Figure 3.1). The Eulerian discretization is defined over a staggered MAC grid [81] with piecewise bilinear velocity and piecewise constant pressure as in [82] (see Figure 3.2). We use a weak form of the Stokes interface problem and simultaneously solve for the pressure, velocity and Lagrange multipliers needed to ensure continuity of the velocity across duplicated cells at the interface. We create a level set defined over the pressure grid from the Lagrangian curve Γ_h. This level set naturally defines a consistent representation of the interface on each of the staggered velocity and pressure grids. Interface cells in each grid can then be duplicated consistently. Our cutting procedure for interface cells is specifically designed to resolve the constant pressure null modes of the weak formulation. We show that this property of the discretization significantly improves the performance of our solver. Advection for the par-



(a) (b)

Figure 3.1: (a) Discrete Lagrangian interface Γ_h. (b) Fluid variables are discretized over an Eulerian MAC grid. A level set representation of the interface is defined over the nodes of the pressure sub grid. We approximate the zero isocontour of the level set with piecewise linear $\hat{\Gamma}_h$. The black dots represent the intersection of the interface $\hat{\Gamma}_h$ and the edges of each pressure grid cells.

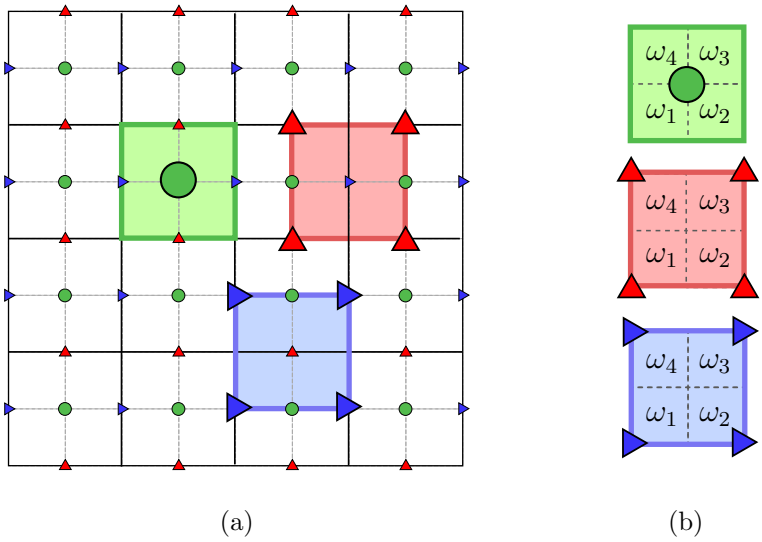


Figure 3.2: (a) Cells on the u, v and p sub-grids of the MAC grid. Triangles and circles indicate where the velocity and pressure degrees of freedom are located. The blue cell with corner triangles pointing to the right is a u grid cell; the red cell with corner triangles pointing up is a v grid cell and the green cell with a single center circle is a p cell. Notice that while u and v cells have four degrees of freedom per cell, p cells have only a single degree of freedom. (b) Cells from Figure (a) with their four subcells indexed as ω_1, ω_2, ω_3, and ω_4.

ticles in the Lagrangian Γ_h is explicit and is done with the interpolated local fluid velocity. Interfacial forces are defined on the Lagrangian particles and then transferred to the cut cells to define stress jump conditions for the Eulerian variables. The complete procedure for advancing one time step is:

1. Compute interface forces (from surface tension or elasticity) at all particles in Γ_h; details in Sections 4.2.1 and 4.2.2.

2. Compute the level set from Γ_h and transfer stress jump conditions; details in Sections 3.1.1 and 3.1.2.

3. Construct discrete stencils that respect the cut grid cells; details in Section 3.1.3.

4. Solve the symmetric system for the coupled velocity, pressure and Lagrange multipliers; details in Section 4.3.1.

5. Interpolate velocities from the MAC grid to the Lagrangian interface and update particle positions using forward Euler; details in Section 3.1.4.

In what follows, we describe each of these steps in detail.

3.1.1 Computation of the level set from Γ_h

Similarly to the VNA presented in the previous chapter, a Lagrangian representation of the interface is again convenient for computing interfacial forces and also for explicit Euler update of the interface geometry. We therefore represent the discrete interface Γ_h again as a sequence of Lagrangian points $\mathbf{x}_i$ where $i \in \{0, 1, 2, \ldots M-1\}$. The points are connected by segments and form a closed curve as shown in Figure 3.1(a) An Eulerian representation of the interface for the discretization of the fluid and pressure variables is defined through a level set which is constructed from Γ_h. The level set values are defined at the corner nodes of the pressure grid cells (see Figure 3.3). This level set can naturally be interpreted as having a constant normal over pressure cells. The discrete Eulerian interface is perturbed to prevent it from intersecting any of the velocity nodes on the MAC grid.

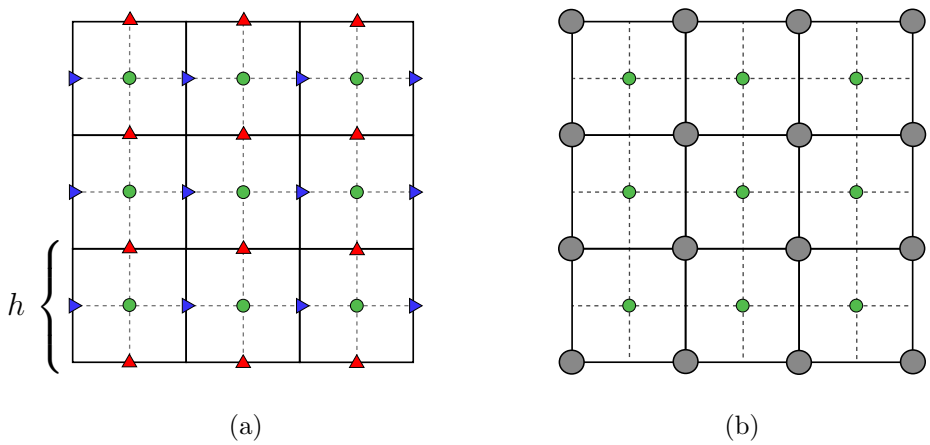


Figure 3.3: Figure (a) shows the original MAC grid with the pressure cells drawn. The edges of the subcells are drawn with dashed lines. Figure (b) shows the nodes where the level set is computed drawn as large circles. These nodes lie on the corners of each pressure grid cell (notice the pressure degrees of freedom are not stored there, but instead at the small green nodes at the center of each p cell); u and v grid nodes are omitted for clarity.

We duplicate cells in the MAC grid that are crossed by the interface to introduce additional degrees of freedom that allow us to accurately capture discontinuities present at the interface. In order to complete this duplication procedure, we must know which cells in the MAC grid intersect the interface and we must know which velocity and pressure grid nodes are inside or outside the interface. We determine this with a level set approximation to Γ_h defined over the pressure grid (see Figures 3.1(b) and 3.3). We define the pressure grid to consist of cells centered around pressure degrees of freedom (see Figures 3.2 and 3.3). These choices are motivated by the constraint that the normal to the interface must be approximated as constant over each cell on the pressure grid. The necessity of this constraint is outlined in Section 3.1.3.5.

3.1.1.1 Level set definition over the p grid

We define u, v and p sub-grids of the MAC grid. These are shown in Figure 3.2. The u and v grids are node-centric in that the degrees of freedom are located at grid nodes. The

p grid however is cell-centric in that p degrees of freedom are located at cell centers. We use h to denote the grid cell size (which we assume is the same in both x and y directions). We define the level set representation of the grid over the p grid. However unlike the pressure degrees of freedom, we store the signed distance values at the nodes of the grid (see Figure 3.3(b)).

The computation of the level set which implicitly defines the discrete Eulerian interface is identical to the computation of the level set outlined in Section 2.2.1, the exception being we compute level set values on the corners of the pressure grid cells instead of on the corners of the doubly-fine grid cells. We refer the reader to that section for more details.

3.1.1.2 Definition of u and v interface cells

The level set on the pressure grid naturally defines interface pressure cells as those with any vertices having signed distance values with opposite signs. However, we also need to know which u and v cells cross the interface. The velocity sub-grids of the MAC grid are defined node-wise (see Figure 3.2). That is, velocity grid cells are defined to have velocity degrees of freedom on their vertices, as opposed to on their centers as with the pressure grid. We will later duplicate all interface pressure and velocity cells to introduce virtual degrees of freedom, however first we must define the interface velocity cells in a manner consistent with the definition of interface pressure cells. This is done by first creating a single segment approximation to the zero isocontour of the level set over each pressure grid cell. Using linear interpolation, we determine the approximate intersection of the level set isocontour at each edge of each interface pressure cell. If this intersection is too close to a u or v node, we perturb it slightly towards the end node of that cell edge lying closer to the intersection point (see Figure 3.4). This modification is enough to move the intersection away from the relevant u or v node. Typically we perturb the position of the intersection point by αh with $\alpha = 10^{-6}$. This perturbation is performed because our discretization of the u and v fluid equations will require that this interface does not cross the velocity grid nodes (see Section 3.1.3.3). Connecting the segments in each pressure cell together forms

a closed curve that we denote as $\hat{\Gamma}_h$ (see Figure 3.1(b)). We use this curve to define which cells in the u and v grids intersect the interface. Each velocity grid cell intersects two pressure cells. A u or v velocity cell is defined to intersect the interface whenever either of its two incident pressure cells has an interface segment that intersects the velocity cell (see Figure 3.4).

3.1.2 Transfer of interfacial forces to Eulerian fluid grid

We require a description of the interface forces $\mathbf{f}^i$ along the embedded interface $\hat{\Gamma}_h$ when discretizing the Eulerian fluid variables (see Section 3.1.3.3). It suffices to define values for $\mathbf{f}^i$ at the center of each segment of $\hat{\Gamma}_h$. Unfortunately, the interface forces $\mathbf{f}^i$ are naturally defined at the nodes of the original Lagrangian interface Γ_h. We provide a procedure to transfer these conditions from the Lagrangian nodes of Γ_h to the centers of the segments on $\hat{\Gamma}_h$. This procedure is very similar to the one outlined in Section 2.2.2, the only difference being $\hat{\Gamma}_h$ is now a segment over each pressure cut cell, while before it was a segment over each doubly-fine grid cell.

Recall that we denote the points on the curve Γ_h as $\mathbf{x}_i$ with $i = 0, 1, \ldots, M-1$. We denote the number of pressure cells that intersect the interface as n_q. Let $\mathcal{P} = \left\{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n_q-1}\right\}$ denote the set of segment centers on $\hat{\Gamma}_h$. As in Chapter 2, we compute the forces $\mathbf{f}^i$ at the points $\mathbf{x}_i$ of Γ_h. For each $k = 0, 1, \ldots, m-1$, let $\hat{\mathbf{x}}_k$ be the point on Γ_h closest to $\mathbf{p}_k$. We linearly interpolate $\mathbf{f}^i$ at $\hat{\mathbf{x}}_k$ from the values at the ends of the segment to which it belongs. This interpolated value is then defined to be the interfacial force at $\mathbf{p}_k$.

3.1.3 Eulerian discretization details

With the Eulerian interface geometry and jump conditions defined from the Lagrangian interface representation, we will now describe the discretization of the Eulerian fluid variables over the MAC grid. Our discrete stencils for the velocity and fluid variables are defined from the weak formulation of the Stokes equations. We will first derive this weak form

including the effects of the discontinuity in the fluid stresses arising from the interfacial forces. We will use a cell duplication procedure to introduce virtual nodes that accurately capture the discontinuities in the stress, but this process requires a Lagrange multiplier term to guarantee continuity in the fluid velocity across the interface. The procedure is designed to exactly capture the null modes in the variational formulation. The pressure can only be determined up to a constant, and the Lagrange multipliers that guarantee continuity can only be determined up to the same constant times the x and y components of the interface normal. We show that failure to capture these modes exactly leads to significantly inferior performance. Furthermore, we show that our definition of the Eulerian interface $\hat{\Gamma}_h$ is designed to facilitate discrete resolution of these modes.

3.1.3.1 Variational formulation

Here we derive the weak form of the Stokes equations with interfacial discontinuities needed for our discrete Eulerian fluid equations. For simplicity of exposition we assume that the domain Ω is rectangular and periodic. Taking the dot product of the force balance portion of the Stokes interface problem (Equation (1.1)) with a function $\mathbf{w}(\mathbf{x}) \in \mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma)$ and integrating over $\Omega \setminus \Gamma$ yields:

$$\begin{aligned} -\int_{\Omega\setminus\Gamma} \mathbf{w} \cdot (\nabla \cdot \sigma)\, dA &= -\int_{\Omega\setminus\Gamma} \nabla \cdot (\mathbf{w} \cdot \sigma) - \nabla \mathbf{w} : \sigma \, dA \\ &= \int_{\Gamma} [\mathbf{w} \cdot \sigma] \cdot \mathbf{n} \, dl + \int_{\Omega\setminus\Gamma} \nabla \mathbf{w} : \sigma \, dA \\ &= \int_{\Omega\setminus\Gamma} \mathbf{w} \cdot \mathbf{f} \, dA, \end{aligned} \tag{3.1}$$

where the divergence theorem was used with $\mathbf{n}$ being the outward unit normal to Ω^- at a point $\mathbf{x} \in \Gamma$ as in Chapter 1. Using the equalities

$$[ab] = a^+ b^+ - a^- b^- = [a]\,\overline{b} + \overline{a}\,[b]\,, \quad \overline{a} = \frac{a^+ + a^-}{2} \quad \text{and} \quad \overline{b} = \frac{b^+ + b^-}{2}, \tag{3.2}$$

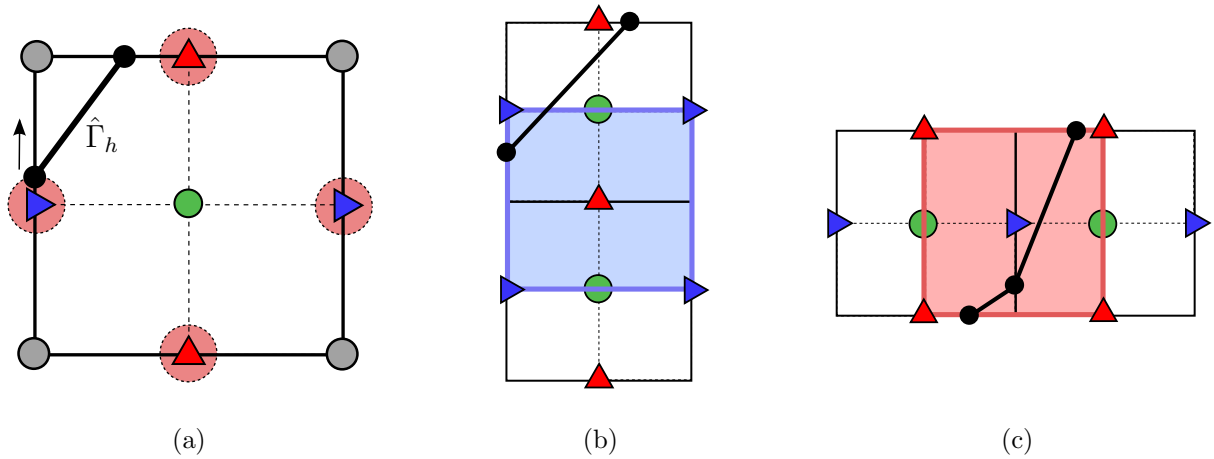


Figure 3.4: Determination of cut u and v cells from $\hat{\Gamma}_h$: velocity cells are defined to be interface cells if either of the two incident pressure cells have a segment in $\hat{\Gamma}_h$ that intersects the cell. Figure (a) shows a typical case of interface perturbation to prevent u and v nodes from intersecting the segments. The dashed red circles indicate the threshold distance from the interface to a given u or v node. The upward arrow indicates the perturbation of the intersection point away from that u node. Figure (b) shows a u cell and the two p cells it intersects. This u cell is considered cut since it is intersected by the interface segment at one of these pressure cells. Figure (c) shows a similar example for a cut cell on the v grid.

and the symmetry of σ, we can rewrite these equations as:

$$
\begin{aligned}
-\int_{\Omega\setminus\Gamma} \mathbf{w}\cdot(\nabla\cdot\sigma)\,dA &= \int_{\Gamma} [\mathbf{w}]\cdot\overline{\sigma}\cdot\mathbf{n}\,dl + \int_{\Gamma} \overline{\mathbf{w}}\cdot[\sigma]\cdot\mathbf{n}\,dl + \int_{\Omega\setminus\Gamma} \nabla\mathbf{w} : \sigma\,dA \\
&= \int_{\Omega\setminus\Gamma} \nabla\mathbf{w} : \left(\mu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^T\right) - p\mathbf{I}\right)\,dA \\
&\quad + \int_{\Gamma} [\mathbf{w}]\cdot\overline{\sigma}\cdot\mathbf{n}\,dl + \int_{\Gamma} \overline{\mathbf{w}}\cdot[\sigma]\cdot\mathbf{n}\,dl \\
&= \int_{\Omega\setminus\Gamma} \frac{\mu}{2}\left(\nabla\mathbf{w}+\nabla\mathbf{w}^T\right) : \left(\nabla\mathbf{u}+\nabla\mathbf{u}^T\right)\,dA \\
&\quad - \int_{\Omega\setminus\Gamma} p(\nabla\cdot\mathbf{w})\,dA + \int_{\Gamma} [\mathbf{w}]\cdot\mathbf{q}\,dl + \int_{\Gamma} \overline{\mathbf{w}}\cdot\mathbf{f}^i\,dl \\
&= \int_{\Omega\setminus\Gamma} \mathbf{w}\cdot\mathbf{f}\,dA,
\end{aligned}
\tag{3.3}
$$

where $\mathbf{q}(\mathbf{x}) = (q^1, q^2)^T = \overline{\sigma}\cdot\mathbf{n}$ is defined only along Γ. Therefore, we can express the weak

form of the Stokes interface problem as:

$$\text{Find } (\mathbf{u}, p, \mathbf{q}) \in \mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{L}^2(\Omega \setminus \Gamma) \times \mathbf{L}^2(\partial\Omega) \times \mathbf{L}^2(\partial\Omega) \text{ such that}$$

$$\int_{\Omega\setminus\Gamma} \frac{\mu}{2} \left(\nabla \mathbf{w} + \nabla \mathbf{w}^T\right) : \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \, dA - \int_{\Omega\setminus\Gamma} p(\nabla \cdot \mathbf{w}) \, dA + \int_{\Gamma} [\mathbf{w}] \cdot \mathbf{q} \, dl \tag{3.4}$$

$$= -\int_{\Gamma} \overline{\mathbf{w}} \cdot \mathbf{f}^i \, dl + \int_{\Omega\setminus\Gamma} \mathbf{w} \cdot \mathbf{f} \, dA, \quad \forall \mathbf{w} \in \mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma)$$

$$\int_{\Omega\setminus\Gamma} \lambda \nabla \cdot \mathbf{u} \, dA = 0, \quad \forall \lambda \in \mathbf{L}^2(\Omega \setminus \Gamma), \tag{3.5}$$

$$\int_{\Gamma} \boldsymbol{\psi}(\mathbf{x}) \cdot [\mathbf{u}(\mathbf{x})] \, dl = \mathbf{0}, \quad \forall \boldsymbol{\psi} \in \mathbf{L}^2(\Gamma) \times \mathbf{L}^2(\Gamma) \tag{3.6}$$

Note that we must treat $\mathbf{q}$ as an additional unknown since $\overline{\sigma} \cdot \mathbf{n}$ will not generally be known. This can be viewed as the Lagrange multiplier associated with the constraint that the velocity must be continuous (Equation (3.6)).

3.1.3.2 Null modes

The null modes of the weak formulation play an important role in our discretization. Specifically, we found that failure to exactly resolve discrete counterparts to the continuous null modes lead to drastically inferior performance. This is not uncommon for simple discrete systems like the Poisson equation with periodic or Neumann boundary conditions in which the rows of the discrete system must sum to zero. In the case of the weak form of the interfacial Stokes problem, there are three null modes. The first two modes arise from our assumption of periodicity in the domain Ω. These two modes are simply constant x or constant y velocities (with pressures p and Lagrange multipliers $\mathbf{q}$ equal to zero). The third mode is a constant pressure mode with Lagrange multipliers equal to the constant pressure times the x or y component of normal at the interfacial discontinuity (and with zero velocities). We can derive these modes by first noting that all integral operators on the left of Equations (3.4-3.6) are bilinear in $\mathbf{u}$, p, $\mathbf{q}$ and $\mathbf{w}$, λ, $\boldsymbol{\psi}$. Hence, with some abuse of notation, we can write the left hand side of these equations as:

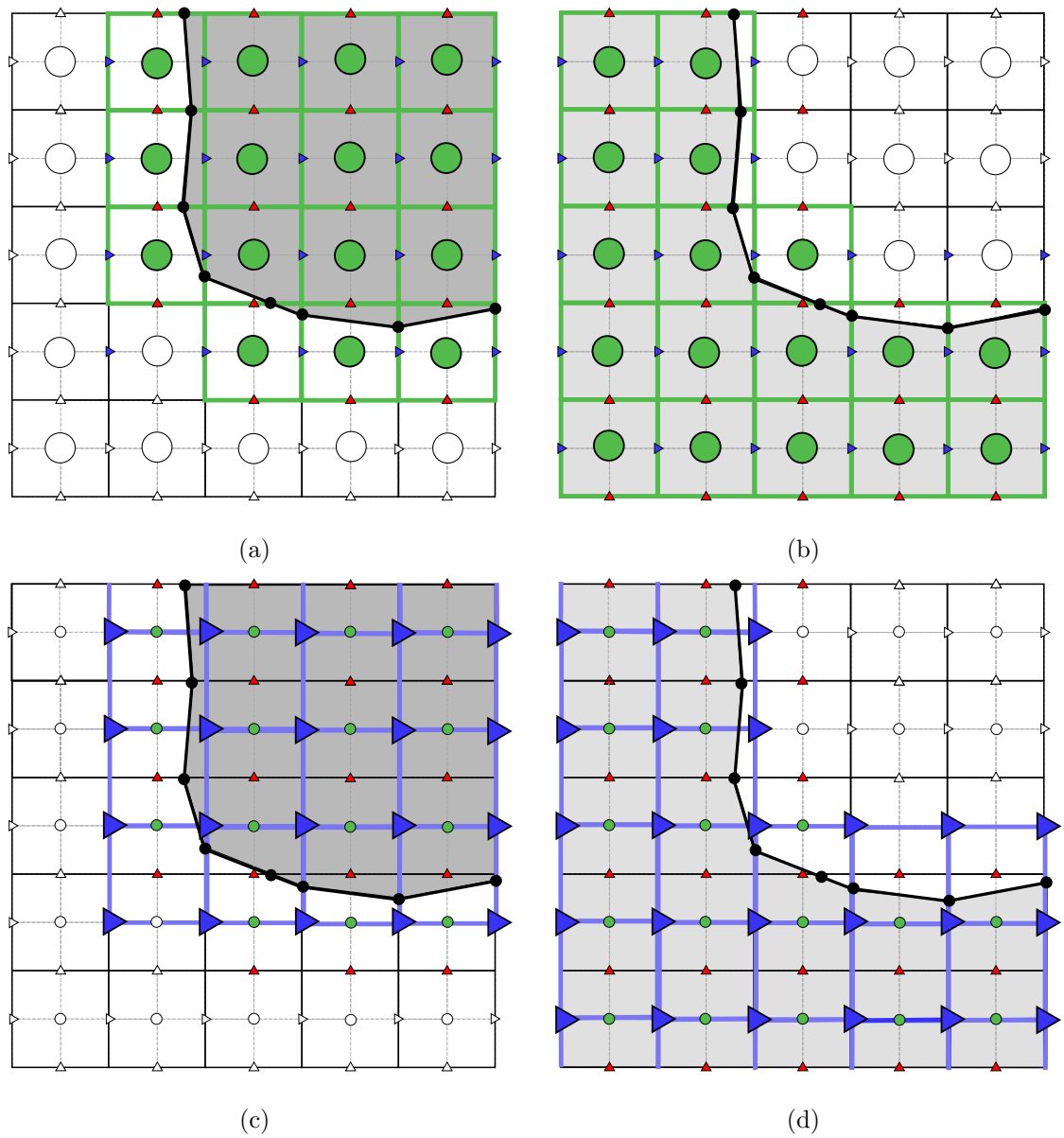


Figure 3.5: All interface velocity and pressure cells are duplicated to introduce virtual degrees of freedom for capturing discontinuities. Figure (a) shows the material and virtual nodes on the pressure grid associated with Ω_h^- while Figure (b) shows the material and virtual nodes of that grid associated with Ω_h^+. The virtual nodes associated with Ω_h^- are the duplicated versions of the material nodes associated with Ω_h^+ which lie at the same positions. Figures (c) and (d) show the nodes of the u grid associated with Ω_h^- and Ω_h^+ respectively.

$$(\mathbf{w},\ \lambda,\ \boldsymbol{\psi})\,\mathcal{K}\begin{pmatrix}\mathbf{u}\\ p\\ \mathbf{q}\end{pmatrix} = (\mathbf{w},\ \lambda,\ \boldsymbol{\psi})\begin{pmatrix}\mathcal{A} & -\mathcal{G} & \mathcal{C}\\ -\mathcal{D} & 0 & 0\\ \mathcal{B} & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}\\ p\\ \mathbf{q}\end{pmatrix}, \tag{3.7}$$

where:

$$\mathbf{w}\mathcal{A}\mathbf{u} = \int_{\Omega\setminus\Gamma} \frac{\mu}{2}\left(\nabla\mathbf{w} + \nabla\mathbf{w}^T\right) : \left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\ dA, \tag{3.8}$$

$$\mathbf{w}\mathcal{G}p = \int_{\Omega\setminus\Gamma} p(\nabla\cdot\mathbf{w})\ dA, \tag{3.9}$$

$$\mathbf{w}\mathcal{C}\mathbf{q} = \int_{\Gamma} [\mathbf{w}]\cdot\mathbf{q}\ dl, \tag{3.10}$$

$$\lambda\mathcal{D}\mathbf{u} = \int_{\Omega\setminus\Gamma} \lambda\nabla\cdot\mathbf{u}\ dA = 0, \tag{3.11}$$

$$\boldsymbol{\psi}\mathcal{B}\mathbf{u} = \int_{\Gamma} \boldsymbol{\psi}(\mathbf{x})\cdot[\mathbf{u}(\mathbf{x})]\ dl = \mathbf{0}. \tag{3.12}$$

Setting $\mathbf{u} = (c^1, c^2)^T$ with $p = 0$ and $\mathbf{q} = \mathbf{0}$ for arbitrary constants c^1 and c^2, it is trivial to see that $\mathbf{w}\mathcal{A}\mathbf{u} = 0$ since $\nabla(c^1, c^2)^T = \mathbf{0}$, $\lambda\mathcal{D}\mathbf{u} = 0$ since $\nabla\cdot(c^1, c^2)^T = 0$ and $\boldsymbol{\psi}\mathcal{B}\mathbf{u} = 0$ since $\mathbf{u} = (c^1, c^2)^T$ is continuous across the interface. Therefore, $\mathbf{u} = (c^1, c^2)^T$ with $p = 0$ and $\mathbf{q} = \mathbf{0}$ are clearly two of the null modes. We can derive the third mode by setting $\mathbf{u} = \mathbf{0}$ with $p = c$ and applying the divergence theorem:

$$\int_{\Omega\setminus\Gamma} p(\nabla\cdot\mathbf{w})\ dA = c\int_{\Omega\setminus\Gamma}(\nabla\cdot\mathbf{w})\ dA = -c\int_{\Gamma}[\mathbf{w}]\cdot\mathbf{n}\ dl = \int_{\Gamma}[\mathbf{w}]\cdot\mathbf{q}\ dl. \tag{3.13}$$

Therefore in order for $-\mathcal{G}p + C\mathbf{q} = \mathbf{0}$, we must have $\mathbf{q} = -c\mathbf{n}$. We will discuss the discrete versions of these kernel modes in Section 3.1.3.5.

3.1.3.3 Discretization of fluid variables

We duplicate all u, v and p interface grid cells to introduce virtual nodes that capture the discontinuities in the fluid stress. That is, for each interface cell, we create a positive and negative version of the cell associated with Ω_h^+ and Ω_h^- respectively. These new cells introduce four new degrees of freedom for the u and v grids and one new degree of freedom for the p grid. As before, we refer to these newly introduced degrees of freedom as "virtual". For the x and y grids, the virtual degrees of freedom are on the vertices of the newly created

Ω_h^+ and Ω_h^- cells that lie outside the respective subdomains. There is only one virtual pressure degree of freedom since the p degrees of freedom are on cell centers. This process is illustrated in Figure 3.5.

This duplication process provides a classification of all grid cells as either interior (Ω_h^-) or exterior (Ω_h^+). Furthermore, all degrees of freedom can also be classified as either interior or exterior. We use this classification to define interior and exterior piecewise bilinear approximations to u and v and interior and exterior piecewise constant approximations to p. Specifically, we assume u and v are bilinear over interior and exterior cells, and we assume p is constant over each exterior and interior cell. Note that our duplication of grid cells at the interface means that these fields are discontinuous at the interface. We can describe these piecewise bilinear fields as

$$u_h(\mathbf{x}) = \sum_{i=1}^{n_u} u_i N_i^u(\mathbf{x}) \quad \text{and} \quad v_h(\mathbf{x}) = \sum_{j=1}^{n_v} v_j N_j^v(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega_h, \tag{3.14}$$

where n_u and n_v are the number of x and y nodes, respectively, and N_i^u and N_j^v are the standard piecewise bilinear interpolating functions associated with x node i and y node j, respectively. We will henceforth use $\mathbf{U} \in \mathbb{R}^{n_u+n_v}$ to denote the vector of velocity degrees of freedom. Similarly, the pressure field is described as

$$p_h(\mathbf{x}) = \sum_{k=1}^{n_p} p_k \phi_k^p(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \Omega_h \setminus \hat{\Gamma}_h, \tag{3.15}$$

where n_p is the number of cells in the duplicated p grid and ϕ_k^p is the characteristic function of the kth pressure cell (equal to one for $\mathbf{x}$ in the cell and zero otherwise). We will henceforth use $\mathbf{P} \in \mathbb{R}^{n_p}$ to denote the vector of pressure degrees of freedom.

Notice that each of the functions $N_i^u(\mathbf{x})$, $N_i^v(\mathbf{x})$ and $\phi_i^p(\mathbf{x})$ are defined on one side of the interface. Hence, if $\mathbf{x} \in \Omega_h^+$, then $N_i^u(\mathbf{x}) = 0$ for all the nodes i on the u grid associated with Ω_h^- and vice versa. The same applies to the basis functions on the v and p grids. Finally, since $\mathbf{q}$ is defined only along Γ, we build its discrete counterpart only along $\hat{\Gamma}_h$. Specifically, we assume that $\mathbf{q}(\mathbf{x})$ is piecewise constant over each segment in $\hat{\Gamma}_h$. It is therefore piecewise constant over each pressure cell on the interface. If there are n_q pressure cells cut by $\hat{\Gamma}_h$,

then:

$$\mathbf{q}_h(\mathbf{x}) = \sum_{l=1}^{n_q} \mathbf{q}_l \phi_l^q(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \hat{\Gamma}_h. \tag{3.16}$$

$\phi_l^q(\mathbf{x})$ is the characteristic function of the lth pressure cell that is cut by the interface. We will henceforth use $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^1 \\ \mathbf{Q}^2 \end{pmatrix} \in \mathbb{R}^{2n_q}$ to denote the vector of Largange multiplier degrees of freedom ($\mathbf{Q}^1$ and $\mathbf{Q}^2$ contain all x and y degrees of freedom of $\mathbf{q}_h$ respectively).

The body force density is discretized also by assuming it is piecewise constant over each pressure cell. Hence:

$$\mathbf{f}_h = \sum_{k=1}^{n_p} \overline{\mathbf{f}}_k \phi_k^p(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega_h \setminus \hat{\Gamma}_h, \tag{3.17}$$

where $\overline{\mathbf{f}}_k$ is the average body force density on the kth pressure cell. In all examples considered, the body forces are spatially constant over each fluid domain and hence the cells averages are constants over Ω_h^+ and Ω_h^-. Similarly, as discussed in Section 3.1.2, we define the jump conditions in the stress to be piecewise constant over each segment in $\hat{\Gamma}_h$, so we can represent the discrete Eulerian stress jump $\mathbf{f}^i$ as

$$\mathbf{f}_h^i(\mathbf{x}) = \sum_{i=1}^{n_q} \mathbf{f}^i(\mathbf{p}_i) \phi_i^q(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \hat{\Gamma}_h. \tag{3.18}$$

Here, $\mathbf{f}^i(\mathbf{p}_i)$ are the values of the interfacial forces transferred from the discrete Lagrangian interface Γ_h to the segment centers $\mathbf{p}_i$ in the discrete Eulerian interface $\hat{\Gamma}_h$ as described on Section 3.1.2.

We now define our discretization of the Stokes interface problem (3.4), (3.5) and (3.6) using the Eulerian approximations u_h, v_h, p_h, $\mathbf{q}_h$, $\mathbf{f}_h^i$ and $\mathbf{f}_h$ just discussed. That is, we approximate the space $\mathbf{H}^1(\Omega \setminus \Gamma) \times \mathbf{H}^1(\Omega \setminus \Gamma)$ by the space $V_h^u(\Omega_h) \times V_h^v(\Omega_h)$ of piecewise bilinear functions over the u and v grids, the space $L^2(\Omega_h)$ by the space $V_h^p(\Omega_h)$ of piecewise constant functions over each pressure grid cell and the space $L^2(\Gamma)$ by the space $V_h^q(\hat{\Gamma}_h)$ of piecewise constant functions over the intersection of $\hat{\Gamma}_h$ with each pressure cell. We obtain a linear system by assuming that $\mathbf{w}_h$ is approximated as with $\mathbf{u}_h$, that λ_h is approximated as with p_h and that $\boldsymbol{\psi}_h$ is approximated as with $\mathbf{q}_h$. Specifically, the discrete variational problem can be written as:

Find $\mathbf{U} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{P} \in \mathbb{R}^{n_p}$, $\mathbf{Q} \in \mathbb{R}^{2n_q}$ such that

$$(\mathbf{W}^T, \mathbf{\Lambda}^T, \mathbf{\Psi}^T)\mathbf{K}\begin{pmatrix}\mathbf{U}\\ \mathbf{P}\\ \mathbf{Q}\end{pmatrix} = (\mathbf{W}^T, \mathbf{\Lambda}^T, \mathbf{\Psi}^T)\begin{pmatrix}\mathbf{F}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix}, \ \forall \begin{pmatrix}\mathbf{W}\\ \mathbf{\Lambda}\\ \mathbf{\Psi}\end{pmatrix} \in \mathbb{R}^{n_u+n_v+n_p+2n_q}, \quad (3.19)$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{A}^{uu} & \mathbf{A}^{uv} & -\mathbf{G}^u & \mathbf{C}^u & \mathbf{0} \\ \mathbf{A}^{vu} & \mathbf{A}^{vv} & -\mathbf{G}^v & \mathbf{0} & \mathbf{C}^v \\ -\mathbf{D}^u & -\mathbf{D}^v & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}^u & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^v & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathbf{F} = \begin{pmatrix}\mathbf{F^1}\\ \mathbf{F^2}\end{pmatrix} = \begin{pmatrix}\hat{\mathbf{F}}^{\mathbf{1}}\\ \hat{\mathbf{F}}^{\mathbf{2}}\end{pmatrix} + \begin{pmatrix}\hat{\mathbf{f}}^{\mathbf{1}}\\ \hat{\mathbf{f}}^{\mathbf{2}}\end{pmatrix}. \quad (3.20)$$

Here, $\mathbf{W} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{\Lambda} \in \mathbb{R}^{n_p}$, and $\mathbf{\Psi} \in \mathbb{R}^{2n_q}$ are the discrete degrees of freedom representing $\mathbf{w}_h$, λ_h and $\boldsymbol{\psi}_h$, respectively, and $\mathbf{U} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{P} \in \mathbb{R}^{n_p}$, and $\mathbf{Q} \in \mathbb{R}^{2n_q}$ are the discrete degrees of freedom representing $\mathbf{u}_h$, p_h and $\mathbf{q}_h$, respectively. $\hat{\mathbf{F}}^1 \in \mathbb{R}^{n_u}$ and $\hat{\mathbf{F}}^2 \in \mathbb{R}^{n_v}$ are associated with the interfacial forces along the x and y directions respectively while $\hat{\mathbf{f}}^1 \in \mathbb{R}^{n_u}$ and $\hat{\mathbf{f}}^2 \in \mathbb{R}^{n_v}$ are associated with the body forces along the x and y directions respectively. We can derive the entries in the matrix $\mathbf{K}$ from Equation (3.4) by assuming that only one entry in any of $\mathbf{W} \in \mathbb{R}^{n_u+n_v}$, $\mathbf{\Lambda} \in \mathbb{R}^{n_p}$, $\mathbf{\Psi} \in \mathbb{R}^{2n_q}$ is equal to one with all other entries zero. The system can then be defined in terms of x equations, y equations, p equations and $\mathbf{q}$ equations as:

x equations ($i = 1, \ldots, n_u$):

$$\begin{aligned}
&\sum_{j=1}^{n_u}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} 2N^u_{i,x}N^u_{j,x} + N^u_{i,y}N^u_{j,y}\,dA\right)u_j + \sum_{j=1}^{n_v}\left(\mu\int_{\Omega_h\setminus\hat{\Gamma}_h} N^u_{i,y}N^v_{j,x}\,dA\right)v_j \\
&\quad - \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N^u_{i,x}\phi^p_j\,dA\right)p_j + \sum_{j=1}^{n_q}\left(\Theta^u_i\int_{\hat{\Gamma}_h} N^u_i\phi^q_j\,dl\right)q^1_j \\
&= -\sum_{j=1}^{n_q}\left(\int_{\hat{\Gamma}_h}\frac{1}{2}N^u_i\phi^q_j\,dl\right)f^{i1}(\mathbf{p}_j) + \sum_{j=1}^{n_p}\left(\int_{\Omega_h\setminus\hat{\Gamma}_h} N^u_i\phi^p_j\,dA\right)\overline{f}^1_j \quad \text{for all } i = 1, \ldots, n_u,
\end{aligned} \tag{3.21}$$

y equations ($i = 1, \ldots, n_v$):

$$\sum_{j=1}^{n_u} \left(\mu \int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_{i,x} N^u_{j,y} \, dA \right) u_j + \sum_{j=1}^{n_v} \left(\mu \int_{\Omega_h \setminus \hat{\Gamma}_h} 2N^v_{i,y} N^v_{j,y} + N^v_{i,x} N^v_{j,x} \, dA \right) v_j$$
$$- \sum_{j=1}^{n_p} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_{i,y} \phi^p_j \, dA \right) p_j + \sum_{j=1}^{n_q} \left(\Theta^v_i \int_{\hat{\Gamma}_h} N^v_i \phi^q_j \, dl \right) q^2_j$$
$$= - \sum_{j=1}^{n_q} \left(\int_{\hat{\Gamma}_h} \frac{1}{2} N^v_i \phi^q_j \, dl \right) f^{i2}(\mathbf{p}_j) + \sum_{j=1}^{n_p} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_i \phi^p_j \, dA \right) \overline{f}^2_j \quad \text{for all } i = 1, \ldots, n_v, \tag{3.22}$$

p equations ($i = 1, \ldots, n_p$):

$$- \sum_{j=1}^{n_u} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} \phi^p_i N^u_{j,x} \, dA \right) u_j - \sum_{j=1}^{n_v} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} \phi^p_i N^v_{j,y} \, dA \right) v_j = 0, \quad \text{for all } i = 1, \ldots, n_p \ , \tag{3.23}$$

q_x equations ($i = 1, \ldots, n_q$):

$$\sum_{j=1}^{n_u} \left(\Theta^u_j \int_{\hat{\Gamma}_h} \phi^q_i N^u_j \, dl \right) u_j = 0, \quad \text{for all } i = 1, \ldots, n_q \ , \tag{3.24}$$

q_y equations ($i = 1, \ldots, n_q$):

$$\sum_{j=1}^{n_v} \left(\Theta^v_j \int_{\hat{\Gamma}_h} \phi^q_i N^v_j \, dl \right) v_j = 0, \quad \text{for all } i = 1, \ldots, n_q. \tag{3.25}$$

Note that in the x and y equations, μ takes the value μ^+ if the node i is associated with Ω^+_h and μ^- if it is associated with Ω^-_h. Also, $\Theta^u_j = 1$ if the node j on the u grid is associated with a cut cell of Ω^+_h, $\Theta^u_j = -1$ if it is associated with a cut cell of Ω^-_h and $\Theta^u_j = 0$ for all other nodes; Θ^v_j is defined analogously for the nodes of the v grid. We can rewrite the equations above using the notation introduced in Equation (3.20) as:

$$A^{uu}_{ij} = \mu \int_{\Omega_h \setminus \hat{\Gamma}_h} 2N^u_{i,x} N^u_{j,x} + N^u_{i,y} N^u_{j,y} \, dA \qquad G^u_{ij} = \int_{\Omega_h \setminus \hat{\Gamma}_h} N^u_{i,x} \phi^p_j \, dA$$
$$A^{uv}_{ij} = \mu \int_{\Omega_h \setminus \hat{\Gamma}_h} N^u_{i,y} N^v_{j,x} \, dA \qquad G^v_{ij} = \int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_{i,y} \phi^p_j \, dA$$
$$A^{vu}_{ij} = \mu \int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_{i,x} N^u_{j,y} \, dA \qquad D^u_{ij} = \int_{\Omega_h \setminus \hat{\Gamma}_h} \phi^p_i N^u_{j,x} \, dA$$
$$A^{vv}_{ij} = \mu \int_{\Omega_h \setminus \hat{\Gamma}_h} 2N^v_{i,y} N^v_{j,y} + N^v_{i,x} N^v_{j,x} \, dA \qquad D^v_{ij} = \int_{\Omega_h \setminus \hat{\Gamma}_h} \phi^p_i N^v_{j,y} \, dA$$

$$C^u_{ij} = \Theta^u_i \int_{\hat{\Gamma}_h} N^u_i \phi^q_j \, dl \qquad B^u_{ij} = \Theta^u_j \int_{\hat{\Gamma}_h} \phi^q_i N^u_j \, dl$$

$$C^v_{ij} = \Theta^v_i \int_{\hat{\Gamma}_h} N^v_i \phi^q_j \, dl \qquad B^v_{ij} = \Theta^v_j \int_{\hat{\Gamma}_h} \phi^q_i N^v_j \, dl,$$

$$\hat{F}^1_i = -\sum_{j=1}^{n_q} \left(\int_{\hat{\Gamma}_h} \frac{1}{2} N^u_i \phi^q_j \, dl \right) f^{i1}(\mathbf{p}_j) \qquad \hat{f}^1_i = \sum_{j=1}^{n_p} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} N^u_i \phi^p_j \, dA \right) \overline{f}^1_j$$

$$\hat{F}^2_i = -\sum_{j=1}^{n_q} \left(\int_{\hat{\Gamma}_h} \frac{1}{2} N^v_i \phi^q_j \, dl \right) f^{i2}(\mathbf{p}_j) \qquad \hat{f}^2_i = \sum_{j=1}^{n_p} \left(\int_{\Omega_h \setminus \hat{\Gamma}_h} N^v_i \phi^p_j \, dA \right) \overline{f}^2_j$$

Note that $\mathbf{A}^{uu}$ and $\mathbf{A}^{vv}$ are symmetric, while $\mathbf{A}^{uv} = (\mathbf{A}^{vu})^T$. Also, we have $\mathbf{G}^u_{ij} = \mathbf{D}^u_{ji}$, $\mathbf{G}^v_{ij} = \mathbf{D}^v_{ji}$, $\mathbf{C}^u_{ij} = \mathbf{B}^u_{ji}$ and $\mathbf{C}^v_{ij} = \mathbf{B}^v_{ji}$, so $\mathbf{G}^u = (\mathbf{D}^u)^T$, $\mathbf{G}^v = (\mathbf{D}^v)^T$, $\mathbf{C}^u = (\mathbf{B}^u)^T$ and $\mathbf{C}^v = (\mathbf{B}^v)^T$. Therefore, the system matrix $\mathbf{K}$ is symmetric.

3.1.3.4 Computation of the matrix K elements

There are many area and line integrals involved in the definition of the discrete system. We will show here that they can all be computed with modest implementation complexity and computational cost. We perform all area integrals by dividing them into sums of integrals over cells in the pressure grid. That is, each area integral is represented as a sum of integrals over the material region of the spatially disjoint pressure cells (whose union is the entire fluid domain). This pressure-cell-wise view of the integration allows us to naturally evaluate these integrals near the geometrically elaborate interface. We use c^p_i to denote the ith pressure cell for $i = 1, 2, \ldots, n_p$. To further facilitate the computation of the integrals, we divide each pressure cell c^p_i into four subcells ω_1, ω_2, ω_3, and ω_4, as shown in Figure 3.6(a). Also, for reasons which will be outlined below, we construct positively oriented boundaries for each of these subcells intersecting the material region as in Figure 3.7. With this convention, the area integrals are expressed as

$$\int_{\Omega_h \setminus \hat{\Gamma}_h} f(x,y) \, dA = \int_{\Omega^+_h} f(x,y) \, dA + \int_{\Omega^-_h} f(x,y) \, dA \tag{3.26}$$

$$= \sum_{i=1}^{n_p} \sum_{j=1}^{4} \int_{w_j \cap \Omega^+_h} f(x,y) \, dA + \sum_{i=1}^{n_p} \sum_{j=1}^{4} \int_{w_j \cap \Omega^-_h} f(x,y) \, dA. \tag{3.27}$$

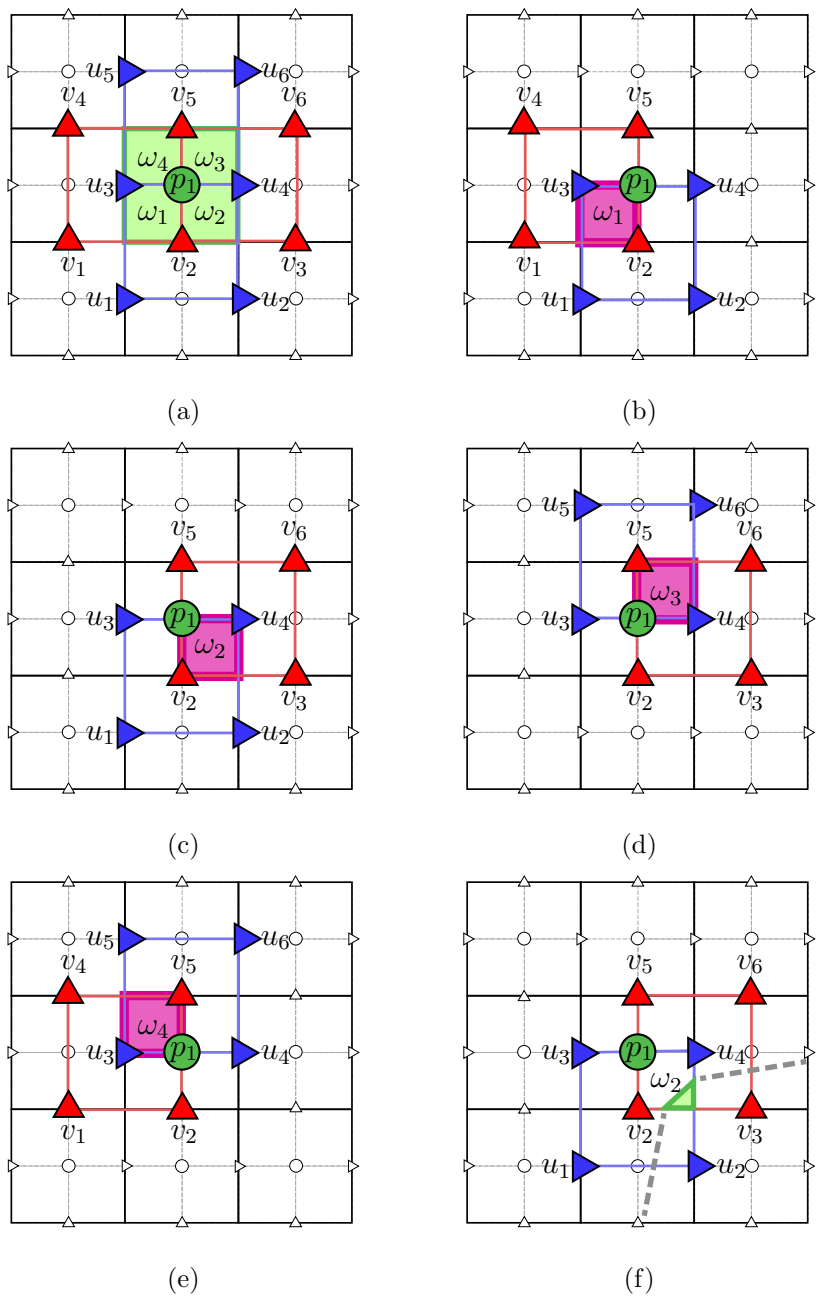


Figure 3.6: Nodes involved in the contribution to the elements of **K** from a given (shaded) pressure cell. Figure (a) shows all contributing nodes on a given material pressure cell, while figures (b)-(e) show the contributing nodes for each subcell (marked in pink) lying on the interior of the pressure cell from Figure (a). Figure (f) shows an example in which the pressure cell is cut; in the example shown fewer u and v nodes are needed to compute the matrix elements associated with that cell in comparison to the cell of Figure (a) (in this case, only subcell ω_2 has a nonvanishing material region).

Note that the second sum in each term of the right hand side is over the subcells ω_j of the pressure cell c_i^p and that some of these subregions may be empty in the case of cut cells, as outlined in Figure 3.7.

We use our pressure-cell-wise integration convention to express the discrete system matrix $\mathbf{K}$ as the sum of the pressure-cell element matrices $\mathbf{K}^{c_i^p} \in \mathbb{R}^{n_i^p \times n_i^p}$ for cells c_i^p. Here we slightly abuse the term "sum" since n_i^p will be between 7 and 13. We simply mean that the entries in $\mathbf{K}$ can be written as sums of entries in these cell-wise matrices. The size of each matrix $\mathbf{K}^{c_i^p}$ is determined by the number of interpolating functions supported over pressure cell c_i^p. The staggering of variables leads to 13 interpolating functions supported over a given interior/uncut pressure cell (6 x-components, 6 y-components and one pressure). There may be fewer than 12 velocity nodes involved in the case of cut cells (see Figure 3.6). The number is determined by the subregions $\omega_1 - \omega_4$ that intersect the domain. This is illustrated in Figure 3.6(f)).

As mentioned above, we further divide the cell-wise $\mathbf{K}^{c_i^p}$ as a sum of matrices defined over the subregions $\omega_1 - \omega_4$ as $\mathbf{K}^{c^p} = \mathbf{K}^{c_i^p}_{\omega_1} + \mathbf{K}^{c_i^p}_{\omega_2} + \mathbf{K}^{c_i^p}_{\omega_3} + \mathbf{K}^{c_i^p}_{\omega_4}$. This is done because the integrands are all smooth over the $\omega_1 - \omega_4$ subregions (with kinks at the boundaries of these regions). Specifically, the integrands are at most quadratic over the subregions and we preform these integrations analytically following the ideas developed in [1]. For example, the entries in $\mathbf{K}^{c_i^p}_{\omega_1}$ involve u_1, u_2, u_3, u_4, v_1, v_2, v_4 and v_5 as demonstrated in Figure 3.6(b); therefore, it only has non-zero values on rows and columns involving these degrees of freedom. The resulting equations based on those degrees of freedom are shown in Figure 3.8. If we order the 13 nodes with indices shown in Figure 3.6(a), then on the interior of the domain, where $c_i^p \cap \Omega_h = c_i^p$, the sum of these four subintegrals is always the same:

$$
\mathbf{K}_{\omega_1}^{c_p} = \frac{1}{64}\begin{pmatrix}
16\mu & 0 & 0 & -16\mu & 0 & 0 & 9\mu & -6\mu & -3\mu & 3\mu & -2\mu & -\mu & 8h \\
0 & 16\mu & -16\mu & 0 & 0 & 0 & 3\mu & 6\mu & -9\mu & \mu & 2\mu & -3\mu & -8h \\
0 & -16\mu & 96\mu & -\mu & 0 & -16\mu & -6\mu & 4\mu & 2\mu & 6\mu & -4\mu & -2\mu & 48h \\
-16\mu & 0 & -\mu & 96\mu & -16\mu & 0 & -2\mu & -4\mu & 6\mu & 2\mu & 4\mu & -6\mu & -48h \\
0 & 0 & 0 & -16\mu & 16\mu & 0 & -3\mu & 2\mu & \mu & -9\mu & 6\mu & 3\mu & 8h \\
0 & 0 & -16\mu & 0 & 0 & 16\mu & -\mu & -2\mu & 3\mu & -3\mu & -6\mu & 9\mu & -8h \\
9\mu & 3\mu & -6\mu & -2\mu & -3\mu & -\mu & 16\mu & 0 & 0 & 0 & -16\mu & 0 & 8h \\
-6\mu & 6\mu & 4\mu & -4\mu & 2\mu & -2\mu & 0 & 96\mu & 0 & -16\mu & -\mu & -16\mu & 48h \\
-3\mu & -9\mu & 2\mu & 6\mu & \mu & 3\mu & 0 & 0 & 16\mu & 0 & -16\mu & 0 & 8h \\
3\mu & \mu & 6\mu & 2\mu & -9\mu & -3\mu & 0 & -16\mu & 0 & 16\mu & 0 & 0 & -8h \\
-2\mu & 2\mu & -4\mu & 4\mu & 6\mu & -6\mu & -16\mu & -\mu & -16\mu & 0 & 96\mu & 0 & -48h \\
-\mu & -3\mu & -2\mu & -6\mu & 3\mu & 9\mu & 0 & -16\mu & 0 & 0 & 0 & 16\mu & -8h \\
8h & -8h & 48h & -48h & 8h & -8h & 8h & 48h & 8h & -8h & -48h & -8h & 0
\end{pmatrix} \tag{3.28}
$$

We illustrate the stencil sparsity in Figure 3.9 for equations sufficiently far from the interface.

For boundary cells where $c^p \cap \Omega_h \neq c^p$, we have to perform the integrations involved in each of $\mathbf{K}_{\omega_j}^{c_p}$ carefully, taking into account the boundary geometry. These integrals can be computed analytically in a straightforward manner following the approach of [1]. Let $\mathcal{P}$ be the polygonal cut-cell geometry consisting of d boundary segments s_i. All pressure subcells cells fit into this category since their boundaries are a set of segments which we orient as in Figure 3.7. All the functions that need to be integrated when computing the elements of $\mathbf{K}$ are at most quadratic over the pressure subcells. We use the procedure described in Appendix C to exactly compute these integrals.

3.1.3.5 Kernel modes of the matrix K

As mentioned in Section 3.1.3.2, there are three kernel modes for the continuous problem. We found that failure to discretely resolve these modes resulted in significantly inferior performance of the method. Specifically, failure to discretize the interface geometry in a manner consistent with the discrete interface jump conditions leads to a matrix $\mathbf{K}$ that does not capture the constant pressure mode. The smallest mode of $\mathbf{K}$ is then numerically similar to the constant pressure mode, but with non-zero eigenvalue. This numerical error

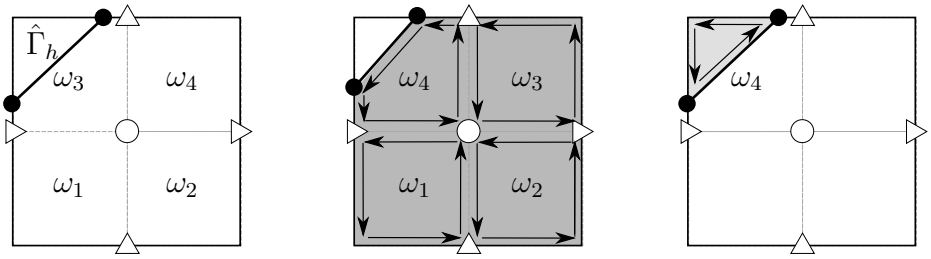


Figure 3.7: Construction of the positively oriented boundaries of each subcell in a given pressure cell. The figure on the left shows the pressure cell and its four subcells ω_1, ω_2, ω_3 and ω_4. The subcells are duplicated since the original pressure cell is cut. The figure in the middle shows the orientation of the boundaries of the subcells associated with the copy of the pressure cell on Ω_h^-, while the figure on the right shows the same for the single subcell (ω_4) associated with the copy of the pressure cell on Ω_h^+.

in the rank of the matrix causes considerable degradation in the performance of the iterative solver for the discrete systems. We demonstrate this with explicit examples in Section 4.3.4.

The first two kernel modes are a consequence of the periodicity of the domain and correspond to constant x and y velocities. These modes will always be captured when the rows of $\mathbf{K}$ associated with x and y velocity equations sum to zero. This constraint is satisfied for nearly all choices of interface geometry and jump condition discretization. As discussed in Section 3.1.3.2, the third mode has zero velocity, constant pressure and Lagrange multiplier $\mathbf{q}$ equal to the constant pressure times the outward normal to the interface. Our choice of interface geometry discretization yields a constant normal to the interface on each pressure cell. We also set the Lagrange multiplier space to be piecewise constant wherever the pressure and the normals are piecewise constant. This combination of choices allows us to have a discrete mode that corresponds exactly to the continuous mode over a piecewise linear interface. Using the notation for our discrete variables, this mode has zero velocities $\mathbf{U}$ with any scalar multiple of $\mathbf{P} = (1, \ldots, 1)^T$ and the same scalar multiple of $q_i^l = -n_i^l$ for $l = 1, 2$ for each entry in $\mathbf{Q}$. Here, $\mathbf{n}_i = (n_i^1, n_i^2)$ is the ith outward normal of the discrete interface $\hat{\Gamma}_h$. As discussed in Section 3.1.3.3, we have a different value of the index i for each cut pressure cell. To verify that this is a kernel mode of $\mathbf{K}$,

i/j	u_1 u_2 u_3 u_4	v_1 v_2 v_3 v_4	p_1
u_1 u_2 u_3 u_4	$\mu \int_{\omega_1} 2N^u_{i,x}N^u_{j,x} + N^u_{i,y}N^u_{j,y}dA$	$\mu \int_{\omega_1} N^u_{i,y}N^v_{j,x}dA$	$-\int_{\omega_1} N^u_{i,x}dA$
v_1 v_2 v_3 v_4	$\mu \int_{\omega_1} N^v_{i,x}N^u_{j,y}dA$	$\mu \int_{\omega_1} 2N^v_{i,y}N^v_{j,y} + N^v_{i,x}N^v_{j,x}dA$	$-\int_{\omega_1} N^v_{i,y}dA$
p_1	$-\int_{\omega_1} N^u_{j,x}dA$	$-\int_{\omega_1} N^v_{j,y}dA$	0

Figure 3.8: Symbolic formulas for the entries in the pressure-cell-wise discretization: $K^{c^p}_{w_1}$.

note that:

$$\mathbf{K}\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{P} \\ \mathbf{Q}^1 \\ \mathbf{Q}^2 \end{pmatrix} = \begin{pmatrix} -(\mathbf{D}^u)^T\mathbf{P} + (\mathbf{B}^u)^T\mathbf{Q}^1 \\ -(\mathbf{D}^v)^T\mathbf{P} + (\mathbf{B}^u)^T\mathbf{Q}^2 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{3.29}$$

Since the velocities are zero, all $\mathbf{q}$ and p equations will be zero. Now, consider the ith x velocity equation. This is the equation associated with x velocity node $i = 1, 2, \ldots, n_u$. Assume that the node is associated with Ω_h^+. Then, if $\mathbf{P} = (1, \ldots, 1)^T$,

$$\begin{aligned} -\left((\mathbf{D}^u)^T\mathbf{P}\right)_i &= -\sum_{j=1}^{n_p} D^u_{ji}p_j = -\sum_{j=1}^{n_p} \int_{\Omega_h \setminus \hat{\Gamma}_h} \phi^p_j N^u_{i,x}\, dA = -\int_{\Omega_h^+} \nabla \cdot (N^u_i, 0)\, dA \\ &= \int_{\hat{\Gamma}_h} N^u_i n^1\, dl = \sum_{k=1}^{n_q} n^1_k \int_{\hat{\Gamma}_h \cap c^p_{j(k)}} N^u_i\, dl. \end{aligned} \tag{3.30}$$

Here we use the fact that the interpolating function N^u_i is only supported on Ω_h^+ since i is assumed to be a positive node. Also, we use the fact that n^1_k is constant over $\hat{\Gamma}_h \cap c^p_{j(k)}$ where $c^p_{j(k)}$ is the pressure cell that intersects the kth segment in $\hat{\Gamma}_h$.

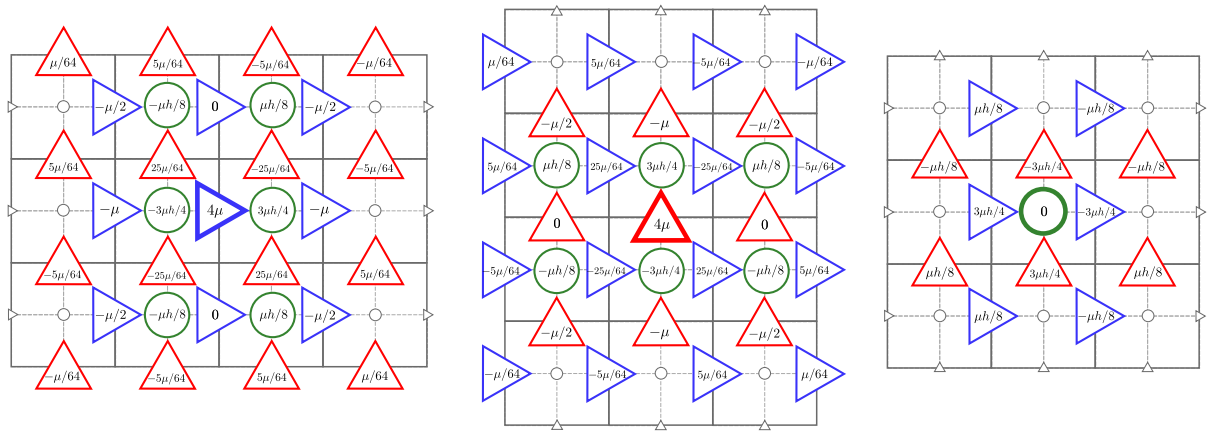


Figure 3.9: Visualization of discrete stencils away from the interface in the matrix $\mathbf{K}$. u node stencils are shown at the left, v node stencils are in the middle and p node stencils are shown at the right.

Now, if we set $q_k^1 = -n_k^1$ for $k = 1, 2, \ldots, n_q$ then,

$$\left((\mathbf{B}^u)^T \mathbf{Q}^1\right)_i = \sum_{k=1}^{n_q} B_{ki}^u q_k^1 = -\sum_{k=1}^{n_q} \Theta_i^u \int_{\hat{\Gamma}_h} N_i^u \phi_k^q n_k^1 \, dl = -\sum_{k=1}^{n_q} n_k^1 \int_{\hat{\Gamma}_h \cap c_{j(k)}^p} N_i^u \, dl \tag{3.31}$$

since i is a positive node implies that $\Theta_i^u = 1$. Therefore $-\left((\mathbf{D}^u)^T \mathbf{P}\right)_i + \left((\mathbf{B}^u)^T \mathbf{Q}^1\right)_i = 0$ for all i positive x node equations. The argument is simular for negative nodes and for the y equations. We reiterate that our choice of discretization for the Lagrangian multiplier $\mathbf{q}$, which is associated with the jump conditions for u and v, is crucial for the incorporation of the constant pressure mode into the kernel of the matrix $\mathbf{K}$. Generally speaking, the discrete $\mathbf{q}$ must be piecewise constant exactly where the discrete interface normals are piecewise constant. Other choices of discretization may prevent this mode from being captured exactly. An example of a discretization which fails to capture this mode is presented in Section 4.3.4. It is shown there that the simulation times are significantly larger when compared to our choice of discretization for $\mathbf{q}$.

Since the matrix $\mathbf{K}$ is symmetric, for any kernel mode $\mathbf{z}$ we have that $\mathbf{z}^T\mathbf{K} = (\mathbf{K}^T\mathbf{z})^T = \mathbf{0}$ because $\mathbf{K} = \mathbf{K}^T$. Hence, from Equation (3.20), we see that the right hand side force terms must be orthogonal to the kernel of $\mathbf{K}$. The constant u, constant v and constant p modes described above provide an orthogonal basis of the kernel of $\mathbf{K}$ and hence we must always

project out components on the right hand side forces which are parallel to the space spanned by these three vectors. Notice that projecting out the constant u and v vectors from the right hand side implies that the total force acting on the fluid in both the horizontal and vertical directions must have zero sum. The right hand side forces on Equation (3.20) are naturally orthogonal to the constant pressure mode.

3.1.4 Interface advection

The interface is advected is performed in the same way as for the continuous viscosity case. Details are therefore omitted here; see Section 2.2.5 for a detailed decription of this step.

CHAPTER 4

Numerical results

We will now discuss a number of numerical tests we used to demonstrate that our method is capable of achieving second order accuracy in L^∞ for u, v and, in the continuous viscosity case, also for p (results for p are first order accurate if the viscosity is discontinuous across the interface). First we will define the precise notion of order of convergence that we use and discuss how we compute it for u, v and p. In the context of continuous fluid viscosity, we present our discretization and convergence results for interfacial elastic and surface tension forces as well as problems in irregular domains. In the context of discontinuous fluid viscosity, we present our discretization and convergence results also for interfacial elastic and surface tension forces and for problems in which the fluid densities are different (with gravitational forces present). In what follows, for irregular domain problems and for problems in which the fluid viscosities are the same, the methods of Chapter 2 were used, while for problems in which the fluid viscosities are different the methods of Chapter 3 were used instead. For a better exposition, all figures mentioned in this chapter are placed at its end.

4.1 Convergence measure

We use N to denote the number of MAC grid cells per direction. We define our discrete approximation to a field g at grid point (x_i, y_j) as $g^N_{i,j}$. Let $E^N_{i,j} = g^N_{i,j} - g(x_i, y_j)$. We examine convergence in the point-wise infinity norm which we define as:

$$e^N = \left\| E^N \right\|_\infty = \max_{i,j} \left| E^N_{i,j} \right| \tag{4.1}$$

We say that the method is r-th order accurate if:

$$e^N =\leq Ch^r \tag{4.2}$$

for some constant C with $h \propto 1/N$ being the cell width. Hence:

$$\begin{aligned}
\left\|g^{2N} - g^N\right\|_\infty &\leq \left\|g^{2N} - g\right\|_\infty + \left\|g^N - g\right\|_\infty \\
&\leq C(h_{2N}^r + h_N^r) \\
&= C'\left(\frac{1}{(2N)^r} + \frac{1}{N^r}\right) \\
&= C'\frac{1}{N^r}\left(\frac{1}{2^r} + 1\right) \\
&= C''\frac{1}{N^r}
\end{aligned} \tag{4.3}$$

where all constants were incorporated into C''. Taking the $\log_{10}$ on both sides of the equation above and defining $a = \log_{10} C''$, we get:

$$\log_{10}\left\|g^{2N} - g^N\right\|_\infty \leq a - r\log_{10} N. \tag{4.4}$$

In other words, the negative of the slope of the plot of $\log_{10}\left\|g^{2N} - g^N\right\|_\infty$ versus $\log_{10} N$ is the order of convergence of the method in the L^∞ norm. We use this procedure to compute the order of convergence in the results that follow. Note that in order to define the quantity $g^{2N} - g^N$, they must be defined at common grid points (i, j). We do this by taking the difference at all grid points (i, j) in the grid associated with g^N as g^{2N} will also be defined at those points.

The discontinuities in the velocities and the pressure at the interface $\hat{\Gamma}_h$ require special treatment in our grid refinement studies. The comparison of g^{2N} and g^N must only be done at points on the same side of the interface. That is, a given grid point (i, j) may be on the interior of the interface for g^{2N} and on the exterior of the interface for g^N (or vice versa). This will happen because the geometry of the interface will change with N since the level set is also defined on a grid with resolution set by N. It would artificially degrade our error convergence estimates if we compare points on opposite sides of the interface. Therefore, we define the infinity norm of the difference between g^{2N} and g^N to only consider points

that are on the same side of the interface on both grids. Note that this also precludes comparing the values at virtual degrees of freedom.

4.2 Examples with continuous fluid viscosity

The following numerical examples show the numerical results we obtained with the VNA for the Stokes problems with continuous fluid viscosities outlined in Chapter 2.

4.2.1 Elastic interface discretization

Our discretization of elastic interfacial forces is very standard, but we briefly cover it here for completeness. Let $\mathbf{x}(\lambda, t)$ be a parametrization of the interface Γ. We assume that the range of λ is $[0, L_0]$, where L_0 is the equilibrium length of the elastic interface. This equilibrium length L_0 is defined such that all elastic forces vanish at all points of Γ if its configuration is a circle of radius $R_0 = 2\pi/L_0$. The elastic force density (per unit length of the parameter λ) at a given point $\mathbf{x}(\lambda, t) \in \Gamma$ is given by the equation:

$$\mathbf{F}(\lambda, t) = \frac{\partial}{\partial \lambda} \left(T(\lambda, t) \boldsymbol{\tau}(\lambda, t) \right) \tag{4.5}$$

where $\boldsymbol{\tau}(\lambda, t)$ is the unit vector tangential to the interface at the point $\mathbf{x}(\lambda, t)$:

$$\boldsymbol{\tau}(\lambda, t) = \frac{\partial \mathbf{x}}{\partial \lambda} \Bigg/ \left\| \frac{\partial \mathbf{x}}{\partial \lambda} \right\| \tag{4.6}$$

and $T(\lambda, t)$ is the tension at that point:

$$T(\lambda, t) = \kappa(\lambda) \left(\left\| \frac{\partial \mathbf{x}}{\partial \lambda} \right\| - 1 \right). \tag{4.7}$$

The function $\kappa(\lambda)$ determines the elastic properties at each point of the interface. Here we will take $\kappa(\lambda)$ to be a constant. The interface is discretely represented with M nodes and we compute the force density $\mathbf{F}_i$ on each node $\mathbf{x}_i$ of Γ_h as:

$$\boldsymbol{\tau}_{i+1/2} = \left(\frac{\partial \mathbf{x}}{\partial \lambda} \right)_{i+1/2} \Bigg/ \left\| \frac{\partial \mathbf{x}}{\partial \lambda} \right\|_{i+1/2} = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}, \tag{4.8}$$

$$T_{i+1/2} = \kappa \left(\frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{\Delta \lambda} - 1 \right) \tag{4.9}$$

and

$$\mathbf{F}_i = \frac{T_{i+1/2}\boldsymbol{\tau}_{i+1/2} - T_{i-1/2}\boldsymbol{\tau}_{i-1/2}}{\Delta\lambda} \tag{4.10}$$

with $\Delta\lambda = L_0/M$. Here the subscript $i + 1/2$ refers to the point at the center of the segment which connects the nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. We divide the force density (4.10) by $\left\|\frac{\partial \mathbf{x}}{\partial \lambda}\right\|_i = \frac{\|\mathbf{x}_i - \mathbf{x}_{i-1}\| + \|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{2\Delta\lambda}$ to define the jump conditions:

$$\mathbf{f}_i^i = \frac{2\Delta\lambda}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\| + \|\mathbf{x}_{i+1} - \mathbf{x}_i\|}\mathbf{F}_i. \tag{4.11}$$

For all elastic interface tests we use a rectangular periodic domain $[-1, 1] \times [-1, 1]$ with an initial elliptical interface of semi-major radius $a = 0.7$ and semi-minor radius $b = 0.4$ centered at the domain origin at $t = 0$. The ellipse is given uniform elasticity constant $\kappa = 10$. The fluid viscosity is set to $\mu = 1$ (uniformly). For a MAC grid with N cells per direction, the interface is represented with $M = N/2$ segments. The nodes of the interface are positioned initially according to the expression:

$$\mathbf{x}_i = (a\cos\theta_i, b\sin\theta_i), \tag{4.12}$$

where $\theta_i = 2\pi i/N_b = 4\pi i/N$ for $i = 1, 2, \ldots, N/2$. This choice of the parametrization of the interface ensures each of its segments has length between $2.5h$ and $4.5h$ at all times of the simulation. The linear systems for the pressure and for the velocities are solved using multigrid. The time step was $\Delta t = 5h^2 = 20/N^2$. We verified the order of convergence for the pressure, for the velocities and for the interface Γ_h according to the L^∞ norm at the times $t_1 = 0.1$, $t_2 = 0.2$, $t_3 = 0.3$ and $t_4 = 0.4$. Also, we checked how the total interface volume change $\Delta V_h =$ (interface volume at time t) $-$ (interface volume at time$t = 0$) converged to zero as the grid was refined and the order of convergence of the maximum and minimum distances between the interface Γ_h and the origin of the rectangular domain (these distances are denoted below as $R_{\max}$ and $R_{\min}$ respectively). The numbers of grid cells per direction used on the convergence test were $N = 32, 64, 128, 256$. As can be seen from Table 4.1, all the results obtained are second order accurate for all these quantities. Figure 4.1 shows the time evolution of the interface Γ_h.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.44	2.30	2.22	2.24
v_h	2.31	2.20	2.26	2.15
p_h	2.74	2.01	2.23	2.23
Γ_h	2.63	2.24	2.11	2.10
$R_{\max}$	2.10	2.05	2.02	2.02
$R_{\min}$	1.76	1.95	1.98	1.99
ΔV_h	2.05	2.01	2.01	2.01

Table 4.1: Orders of convergence for Example 4.2.1.

4.2.2 Surface tension

Let $\mathbf{x}(s,t)$ be the parameterization of the interface with respect to arclength. The surface tension force density (per unit interface length) at a given point $\mathbf{x}(s,t) \in \Gamma$ is:

$$\mathbf{f}^i(s,t) = 2\sigma \frac{\partial \boldsymbol{\tau}(s,t)}{\partial s} \tag{4.13}$$

where again $\boldsymbol{\tau}(s,t)$ is the unit vector tangential to the interface at the point $\mathbf{x}(s,t)$ and σ is the surface tension constant of the interface Γ. Since we are using the interface arclength as our paramertization, we have:

$$\left\| \frac{\partial \mathbf{x}}{\partial s} \right\| = 1. \tag{4.14}$$

The surface tension force density is then in this case:

$$\mathbf{f}_i^i = 2\sigma \frac{\tau_{i+1/2} - \tau_{i-1/2}}{\overline{\Delta s}} \quad , \quad \overline{\Delta s} = \frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\| + \|\mathbf{x}_i - \mathbf{x}_{i-1}\|}{2}. \tag{4.15}$$

Here, $\tau_{i+1/2}$ is the same as in Equation (4.8). Note that the surface tension force density (4.13) is always normal to the interface.

For all surface tension tests we use a rectangular periodic domain $[-1,1] \times [-1,1]$ with an elliptical interface of semi-major radius $a = 0.7$ and semi-minor radius $b = 0.4$ centered at the domain origin at $t = 0$. The fluid viscosity is again set to μ, the surface tension constant is $\sigma = 10$. We use the same number of interface nodes as in the elastic interface

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.25	1.90	1.99	2.05
v_h	2.05	1.84	2.06	2.12
p_h	1.75	1.79	1.92	2.05
Γ_h	2.06	1.95	2.37	2.10
$R_{\max}$	2.24	1.95	1.93	1.95
$R_{\min}$	1.96	1.96	1.97	1.97
ΔV_h	1.83	1.90	1.91	1.92

Table 4.2: Orders of convergence for Example 4.2.2.

examples: $M = N/2$ segments. The nodes of the interface are positioned initially as in the elastic interface case and the linear systems for the pressure and for the velocities are solved using multigrid. Also the time step is again $\Delta t = 5h^2 = 20/N^2$.

We verified the order of convergence for the pressure, for the velocities and for the interface Γ_h according to the L^∞ norm at the times $t_1 = 0.1$, $t_2 = 0.2$, $t_3 = 0.3$ and $t_4 = 0.4$. Also, we again checked the the order of convergence of the total interface volume change ΔV_h towards zero as the grid was refined and the order of convergences of $R_{\max}$ and $R_{\min}$. The numbers of MAC grid cells per direction used on the convergence tests were $N = 32, 64, 128, 256$. As can be seen from Table 4.2 all the results obtained are second order accurate for all these quantities. Figure 4.2 shows the time evolution of the interface Γ_h.

4.2.3 Irregular domain

We also demonstrate second order convergence with the following irregular domain problem. Consider two concentric circles with radii $R_1 = 0.3$ and $R_2 = 0.8$ respectively with a Stokesian fluid in between having viscosity $\mu = 1$. The inner cylinder rotates counterclockwise with angular velocity $\omega_1 = 2$ and the outer circle rotates counterclockwise with angular velocity $\omega_2 = 1$. We used the values $N = 40, 80, 160, 320, 640$ for the MAC grid resolutions.

Quantity	$t = 0.0$
u_h	2.16
v_h	2.11

Table 4.3: Orders of convergence for Example 4.2.3.

The linear systems are solved using conjugate gradients with Jacobi preconditioning. In order to visualize the fluid motion, we divided the fluid domain in two halves: $x > 0$ and $x < 0$. Marker particles were placed at these two domains and allowed to be advected with the fluid using forward Euler. As can be seen from Table 4.3, the results obtained are second order accurate for the velocities. Figure 4.3 shows position of the marker particles at different time steps.

4.3 Examples with discontinuous fluid viscosity

The following numerical examples show the numerical results we obtained with the VNA for the Stokes problems with discontinuous fluid viscosities outlined in Chapter 3.

4.3.1 Jacobi preconditioned MINRES

Our discretization requires the solution of a symmetric linear system for the velocities, the pressure and the Lagrange multipliers. In all examples described below, we use MINRES to solve the linear systems involved. As noted in [1], the cut cells can lead to rows in the matrix with very small magnitude. This can significantly degrade the condition number of the system. We alleviate this with a simple Jacobi preconditioning strategy. This is slightly different that in [1] where the matrices were all symmetric positive definite. Our system is symmetric but with a zero lower right block. We therefore use a block Jacobi approach to alleviate rows with small magnitude.

We first construct a Jacobi preconditioner for the $\mathbf{A}$ matrix portion of $\mathbf{K}$. We define

this block of the preconditioner $\mathbf{J}^{(1)}$ as:

$$J_{ii}^{(1)} = \begin{cases} \dfrac{1}{\sqrt{K_{ii}}} & \text{if } i < N_u + N_v \\ 1 & \text{if } i \geq N_u + N_v \end{cases} \tag{4.16}$$

Let $\mathbf{K}^{(1)} = \mathbf{J}^{(1)}\mathbf{K}\mathbf{J}^{(1)}$. The diagonal entries in the first $N_u + N_v$ rows of $\mathbf{K}^{(1)}$ are 1 and its off-diagonal entries are smaller than 1. In the second step, we construct a preconditioning matrix that normalizes each row $i > N_u+N_v$ of $\mathbf{K}^{(1)}$, leaving its leading $(N_u+N_v)\times(N_u+N_v)$ block unchanged. Letting $\mathbf{k}_i$ be the ith row of $\mathbf{K}^{(1)}$, we construct the second preconditioning matrix $\mathbf{J}^{(2)}$ as:

$$J_{ii}^{(2)} = \begin{cases} 1 & \text{if } i < N_u + N_v \\ \dfrac{1}{\|\mathbf{k}_i\|_2} & \text{if } i \geq N_u + N_v \end{cases}. \tag{4.17}$$

Letting $\mathbf{J} = \mathbf{J}^{(2)}\mathbf{J}^{(1)}$ and writing the solution vector in Equation (3.19) as $\mathbf{x}$ and the right hand side as $\mathbf{b}$, we then solve the following linear system:

$$\mathbf{JKJJ}^{-1}\mathbf{x} = (\mathbf{JKJ})\mathbf{y} = \mathbf{Jb}, \tag{4.18}$$

where $\mathbf{y} = \mathbf{J}^{-1}\mathbf{x}$.

4.3.2 Elastic interface discretization

Consider the elastic interface problem of Section 4.2.1. We simulated the evolution of an elastic interface as in that section but with fluid viscosities inside and outside the interface set to $\mu^- = 3$ and $\mu^+ = 1$ respectively. All other parameters of the simulation were kept the same. As can be seen from Table 4.4, all the results obtained are second order accurate for all these quantities except for the pressure, which is first order accurate. Figure 4.4 shows the time evolution of the interface Γ_h.

4.3.3 Surface tension

Consider the surface tension problem of Section 4.2.2. We simulated the evolution of the contact interface as in that section but with fluid viscosities inside and outside the interface

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	1.76	1.80	2.40	1.85
v_h	1.60	1.85	2.03	2.03
p_h	0.63	0.85	0.82	0.69
Γ_h	2.55	2.47	2.48	2.47
$R_{\max}$	2.03	2.05	2.09	2.10
$R_{\min}$	2.02	2.08	2.08	2.10
ΔV_h	2.23	2.16	2.21	2.24

Table 4.4: Orders of convergence for Example 4.3.2.

Quantity	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$
u_h	2.05	1.80	2.25	1.88
v_h	1.89	2.16	1.85	1.87
p_h	0.84	0.97	1.06	0.86
Γ_h	1.82	2.01	2.21	2.34
$R_{\max}$	2.22	2.28	2.28	2.26
$R_{\min}$	1.95	1.96	1.98	1.98
ΔV_h	2.12	2.15	2.16	2.19

Table 4.5: Orders of convergence for Example 4.3.3.

set to $\mu^- = 3$ and $\mu^+ = 1$ respectively. All other parameters of the simulation were kept the same. As can be seen from Table 4.5, all the results obtained are second order accurate for all these quantities except for the pressure, which is first order accurate. Figure 4.5 shows the time evolution of the interface Γ_h.

4.3.4 Pressure null mode test

We provide evidence of the importance of capturing the constant pressure null mode. Our initial investigations used a doubly-fine level set to form the approximated Lagrangian

N	Our method	Doubly-fine level set
32	978	1897
64	2500	5402
128	6635	16095

Table 4.6: Average number of MINRES iterations per time step for Example 4.3.4

N	Our method	Doubly-fine level set
32	1.40	1.91
64	4.63	8.80
128	46.3	86.0

Table 4.7: Average run time per time step (in seconds) for Example 4.3.4

interface $\hat{\Gamma}_h$ following the approach in [83]. That is, we first used a level set defined over a grid with twice as many cells per dimension as the pressure grid. The nodes of the pressure grid are also nodes of this doubly-fine grid. Although this procedure is still adequate for consistently computing the integrals which define the system matrix $\mathbf{K}$ and the right hand side components, the resulting matrix $\mathbf{K}$ does not have the constant pressure mode in its kernel. This mode is then only captured approximately as $N \to \infty$. The effective conditioning of the system was significantly degraded by this. To illustrate, we ran the tests in the elasticity example section for $N = 32, 64, 128$ up to $t = 0.1$ and computed the average number of MINRES iterations for convergence and also the average amount of execution time per time step. Our residual tolerance was $r_{TOL} = 10^{-7}$. As can be seen from Tables 4.6 and 4.7, performance was significantly improved by capturing these modes.

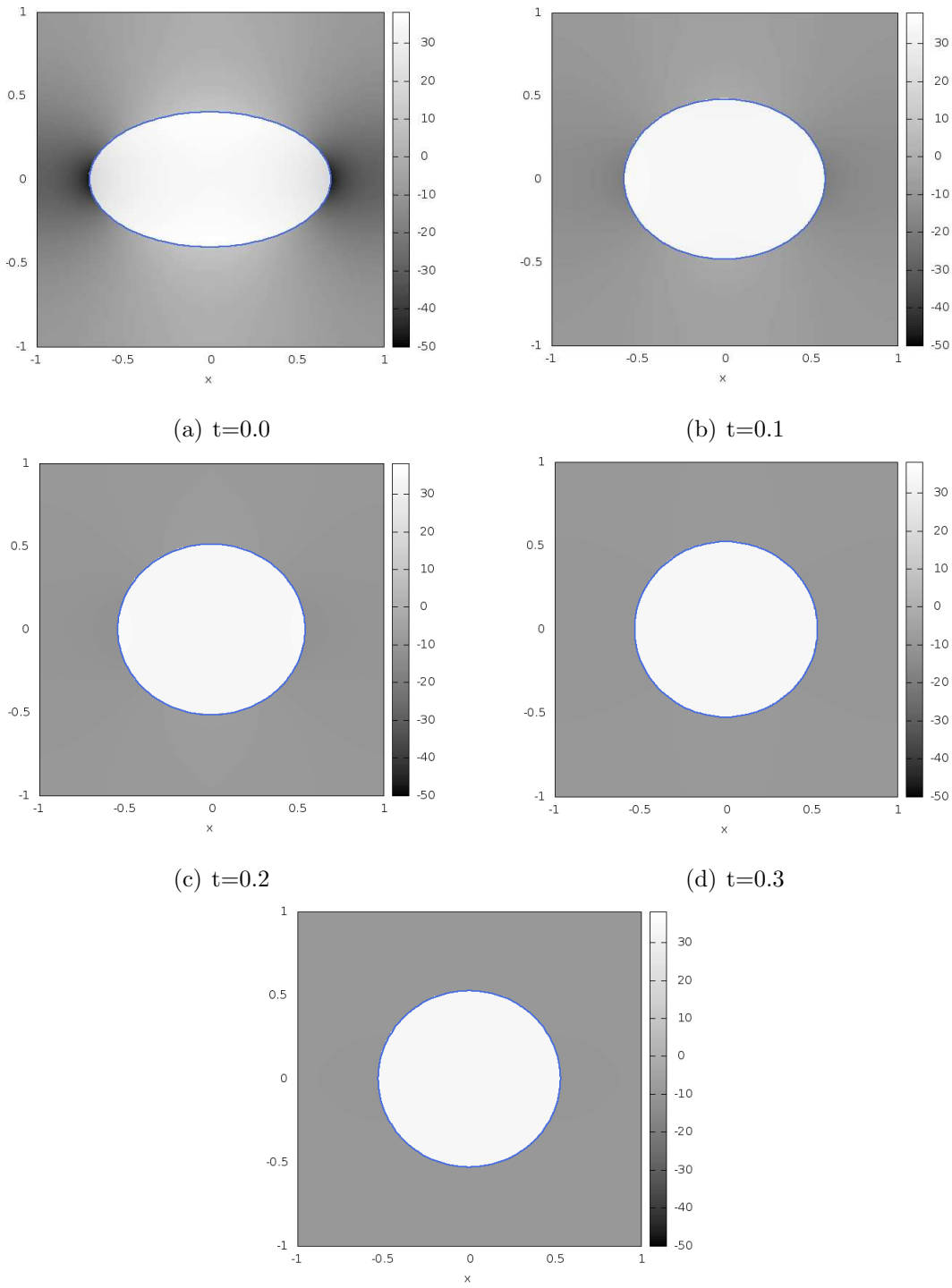
4.3.5 Rising drop

We also demonstrate the effect of discontinuous material properties by simulating fluids with different densities and viscosities under with a gravitational body force. We use a uniform gravity $g = 10$, with fluid densities $\rho^- = 1$ inside the interface and the $\rho^+ = 2$

outside the interface. The viscosities are set to either $\mu^+ = 1$ and $\mu^- = 3$ or to $\mu^+ = 1$ and $\mu^- = 3$. We use a rectangular periodic domain $[-1, 1] \times [-1, 1]$ with an initial elliptical interface of semi-major radius $a = 0.5$ and semi-minor radius $b = 0.2$ centered at the the position $(0, -0.7)$. We simulate interfaces with no surface tension and with surface tension constant $\sigma = 1$ for both these viscosity combinations. The interface is represented with $M = N$ segments for the cases in which the surface tension constant is set to zero and with $M = N/2$ segments otherwise. The nodes of the interface are positioned initially according to the expression:

$$\mathbf{x}_i = (a \cos \theta_i, b \sin \theta_i - 0.7), \tag{4.19}$$

where $\theta_i = 2\pi i/N_b = 4\pi i/N$ for $i = 1, 2, \ldots, N/2$. The time step is $\Delta t = 50h^2 = 200/N^2$. The results are shown on Figure 4.6.



(a) t=0.0

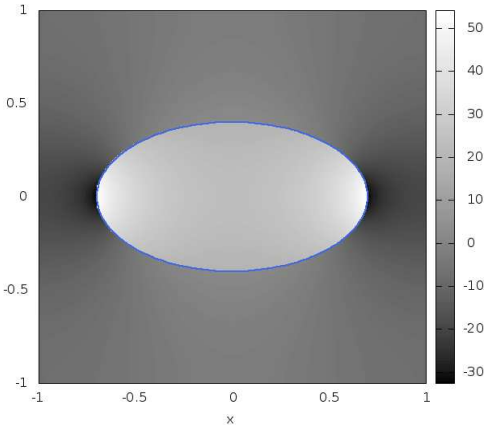
(b) t=0.1

(c) t=0.2

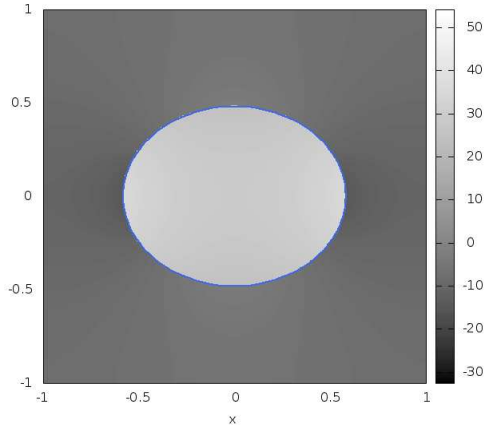
(d) t=0.3

(e) t=0.4

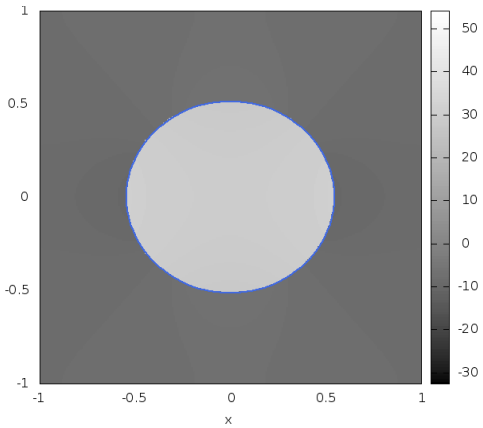
Figure 4.1: Configuration of Γ_h at different times for the elastic interface problem of Section 4.2.1. Pressure is shown in gray. Dark regions have lower pressure; brighter regions have higher pressure.



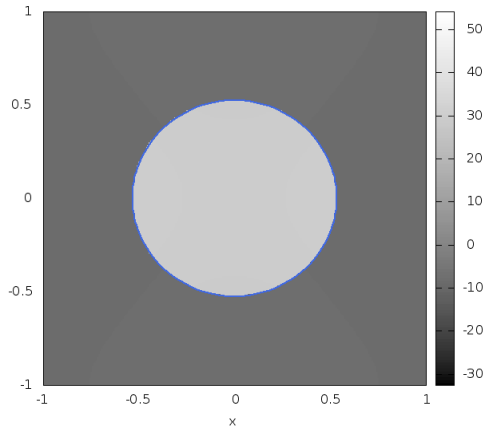
(a) t=0.0



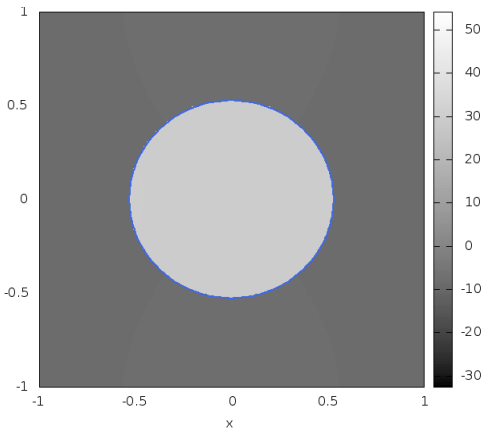
(b) t=0.1



(c) t=0.2

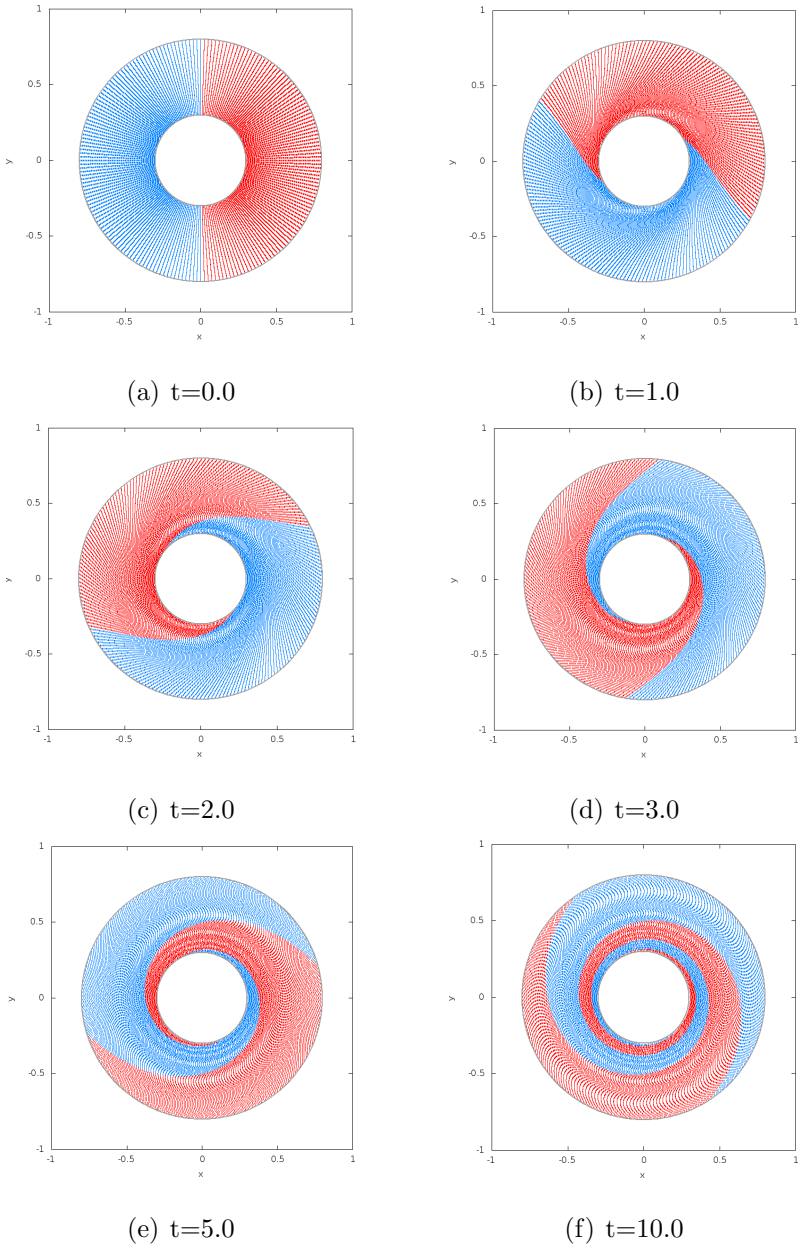


(d) t=0.3



(e) t=0.4

Figure 4.2: Configuration Γ_h at different times for the surface tension problem of Section 4.2.2. Pressure values are shown in gray. Dark regions have lower pressure; brighter regions have higher pressure. Compare these results with the ones of Figure 4.1.



(a) t=0.0

(b) t=1.0

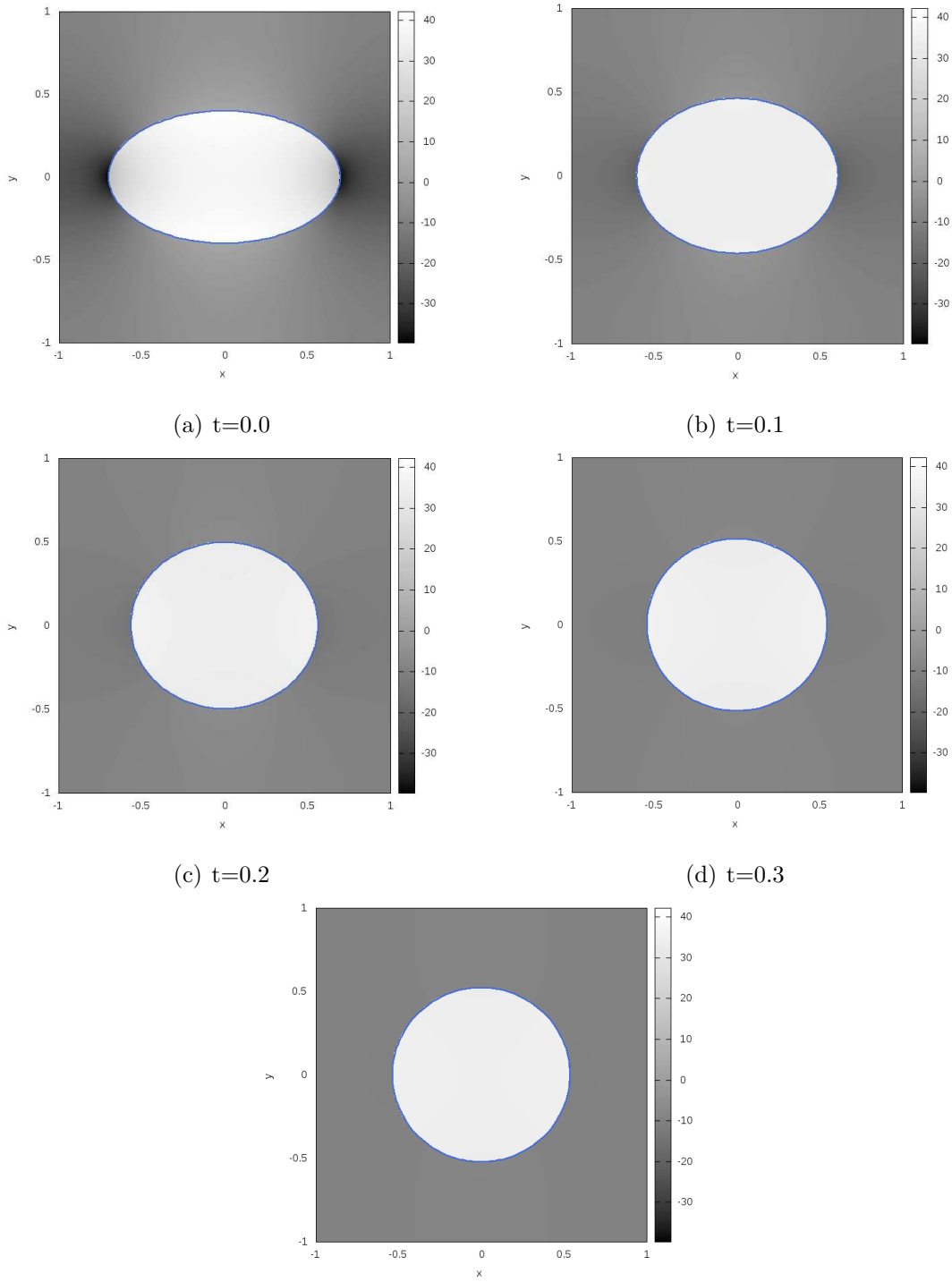
(c) t=2.0

(d) t=3.0

(e) t=5.0

(f) t=10.0

Figure 4.3: Position of the marker particles at different times for the problem of Section 4.2.3. The fluid pressure is spatially uniform in this example.



(a) t=0.0

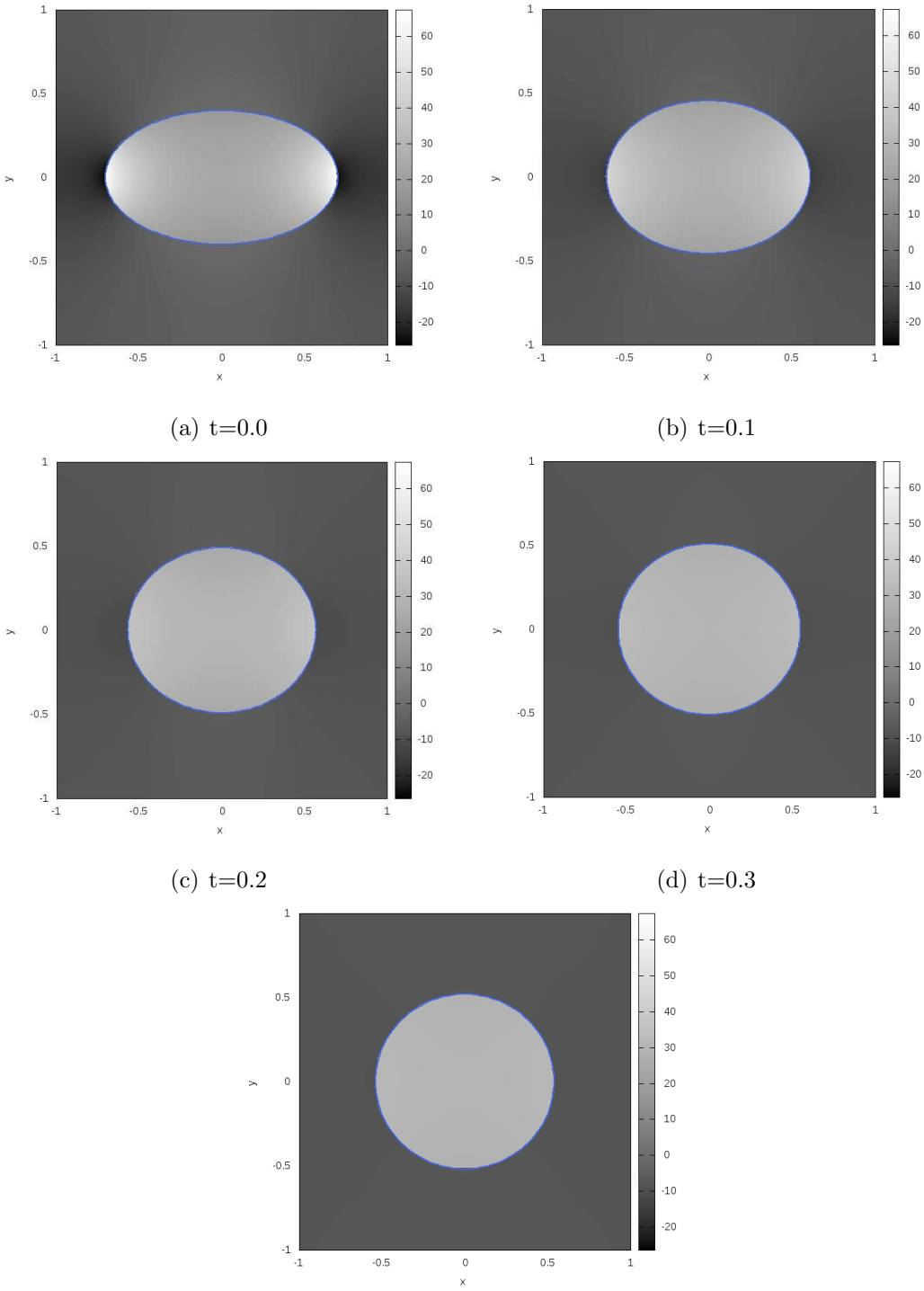
(b) t=0.1

(c) t=0.2

(d) t=0.3

(e) t=0.4

Figure 4.4: Configuration of Γ_h at different times for the elastic interface problem of Section 4.3.2. Pressure is shown in gray. Dark regions have lower pressure; brighter regions have higher pressure.



(a) t=0.0

(b) t=0.1

(c) t=0.2

(d) t=0.3

(e) t=0.4

Figure 4.5: Configuration Γ_h at different times for the surface tension problem of Section 4.3.3. Pressure values are shown in gray. Dark regions have lower pressure; brighter regions have higher pressure. Compare these results with the ones of Figure 4.4.

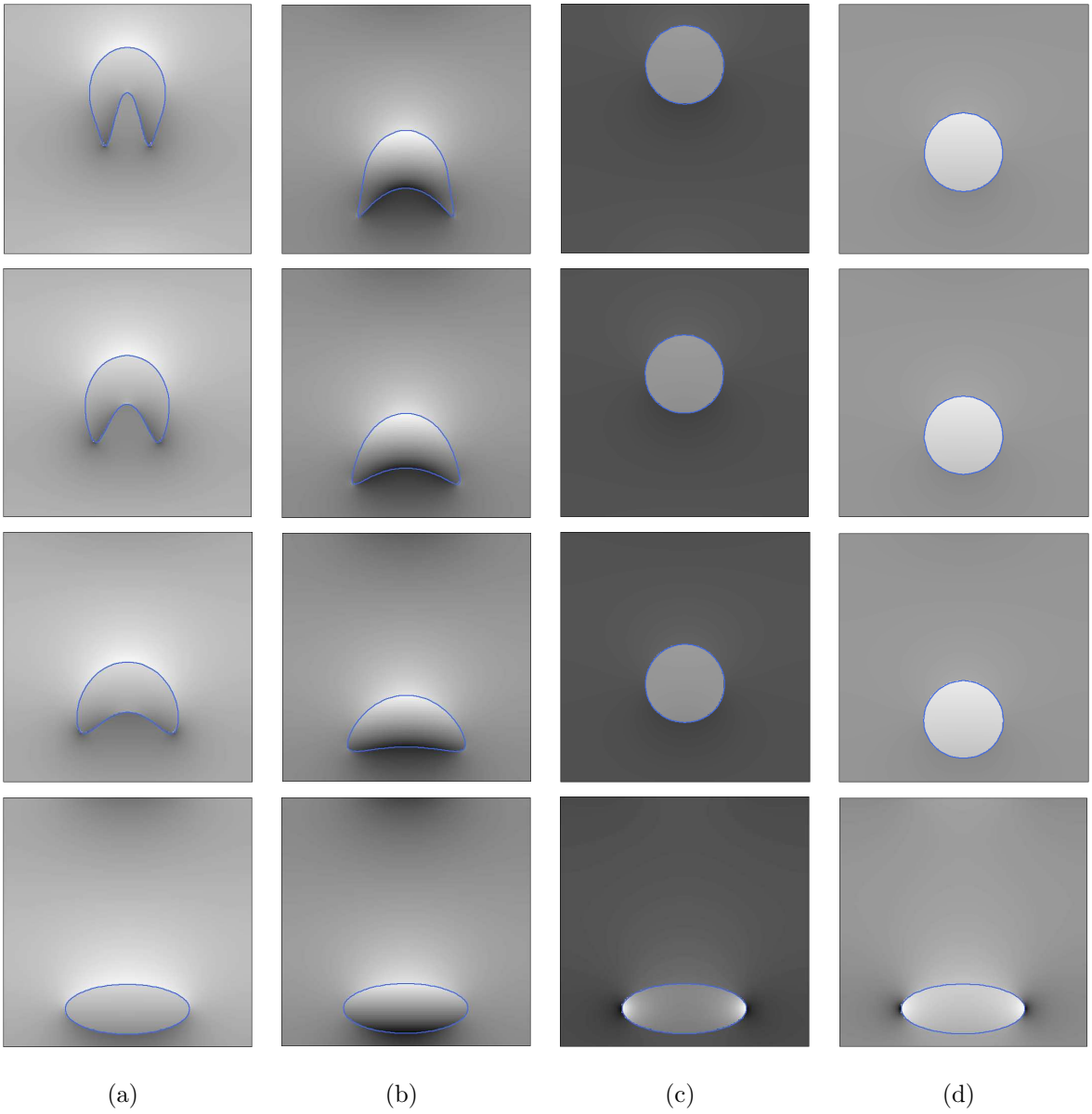


Figure 4.6: Rising drop. Configurations of the interface Γ_h are shown at different times and for different combinations of σ, μ^+ and μ^-. Pressure is shown in gray. Dark regions have lower pressure; brighter regions have higher pressure. Each column represents a different test case; from bottom to top, the figures show the interface configurations at $t = 0.0$, $t = 3.0$, $t = 5.0$ and $t = 7.0$. From left to right, each column represents a test with each of these parameter combinations: $\mu^- = 3, \mu^+ = 1, \sigma = 0$ at column (a), $\mu^- = 1, \mu^+ = 3, \sigma = 0$ at column (b), $\mu^- = 3, \mu^+ = 1, \sigma = 1$ at column (c) and $\mu^- = 1, \mu^+ = 3, \sigma = 1$ at column (d).

CHAPTER 5

Summary and discussion

In this dissertation, we have presented two Virtual Node Algorithms (VNAs) designed to accurately solve the incompressible Stokes equations with geometrically irregular domains and interfaces. Both algorithms handle singular forces such as surface tension or elastic forces supported along the interfaces.

Our first algorithm is an efficient and easy to implement second order accurate solver for the Stokes equations with immersed interfaces and geometrically irregular domains. The assumed spatial uniformity of the fluid viscosity permits a reduction of the problem to solving three independent Poisson equations for the velocities and the pressure. This task can be optimally performed using fast Poisson solvers such as multigrid. An additional Poisson equation can then be solved to enforce a discrete divergence-free condition for the velocities without sacrificing second order accuracy. The linear system associated with each of these Poisson equations is symmetric positive definite. Examples with elastic interfaces and surface tension were presented as well as flow of a single fluid in the region separating two rotating concentric cylinders. All results obtained have shown second order accuracy in L^∞ for the velocities, the pressure and all other quantities measured.

Our second algorithm is a second order accurate solver for the Stokes equations with immersed interfaces and discontinuous fluid properties. We considered examples in which both the fluid viscosities and densities are discontinuous across the interface. Our method is capable of resolving discrete counterparts of the continuous null modes for these interface problems and we showed that this is necessary for efficient performance. Also, the method is easy to implement and yields a symmetric linear system of equations. Examples with elastic interfaces and surface tension were presented, as well as examples with drops which

rise due to the presence of gravitational forces coupled with discontinuous fluid densities. All results obtained have shown second order accuracy in L^∞ for the velocities and all other quantities measured except for the pressure, which is first order accurate.

It is the intention of the authors to extend the current methods to deal with the full incompressible Navier-Stokes equations in the context of interfacial forces and discontinuous fluid properties.

APPENDIX A

Derivation of pressure boundary conditions

Let Ω be an irregular domain whose surface is a smooth curve $\partial\Omega$. We will assume here that Dirichlet boundary conditions are know for the velocities: $u(\mathbf{x}) = U(\mathbf{x})$ and $v(\mathbf{x}) = V(\mathbf{x})$ for all $\mathbf{x} \in \partial\Omega$, with $U(\mathbf{x})$ and $V(\mathbf{x})$ being functions defined only along $\partial\Omega$.

Let $\mathbf{x_0} = (x_0, y_0)$ be a point in $\partial\Omega$. Let (ξ, η) be an orthogonal set of coordinates with its origin at $\mathbf{x_0}$ constructed as in figure A.1. The ξ axis defines a line which is tangential to $\partial\Omega$ at the point $\mathbf{x_0}$, while η defines a line which is parallel to the outward normal of $\partial\Omega$ at that point.

Denoting V^ξ and V^η the components of the velocity on the (ξ, η) coordinates, we have:

$$\frac{\partial V^\xi}{\partial \xi} + \frac{\partial V^\eta}{\partial \eta} = 0 \tag{A.1}$$

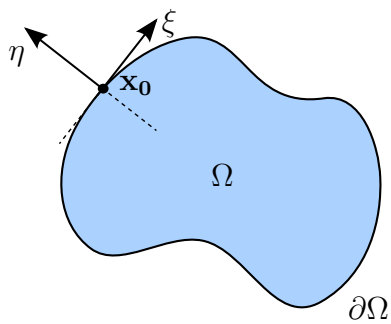


Figure A.1: Local coordinates centered at the point $\mathbf{x_0}$ of the domain boundary $\partial\Omega$. The ξ axis is tangential to $\partial\Omega$ at $\mathbf{x_0}$. The η axis is normal to $\partial\Omega$ at $\mathbf{x_0}$.

and

$$\left(\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}\right)V^\eta-\frac{\partial p}{\partial\eta}+f^\eta=0 \tag{A.2}$$

$$\left(\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}\right)V^\xi-\frac{\partial p}{\partial\xi}+f^\xi=0. \tag{A.3}$$

Equation (A.1) is merely the divergence-free condition expressed in the set of orthogonal coordinates defined above. Similarly, the pair of equations (A.2) and (A.3) are the Stokes equations also written in the same set of coordinates, with f^ξ and f^η being the components of $\mathbf{f}$ along the ξ axis and the η axis respectively.

Notice that, from equation (A.1), we have:

$$\begin{aligned}\left(\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}\right)V^\eta &= \frac{\partial^2 V^\eta}{\partial\xi^2}+\frac{\partial}{\partial\eta}\left(-\frac{\partial V^\xi}{\partial\xi}\right)\\ &= \frac{\partial}{\partial\xi}\left(\frac{\partial V^\eta}{\partial\xi}-\frac{\partial V^\xi}{\partial\eta}\right)\\ &= \frac{\partial\omega}{\partial\xi}\end{aligned} \tag{A.4}$$

where ω is the fluid vorticity. Hence, as $\mathbf{x}\longrightarrow\mathbf{x_0}$, we have from equation (A.2) that:

$$\frac{\partial p}{\partial\eta}\longrightarrow f^\eta+\frac{\partial\omega}{\partial\xi} \tag{A.5}$$

where the right hand side is computed at $\mathbf{x_0}$. In other words, the Neumann boundary conditions for the fluid pressure at a given point of the domain surface are determined by the normal component of the body force and by the tangential derivative of the fluid vorticity at that point.

It is also important to keep in mind that the incompressibility condition $\nabla\cdot\mathbf{u}=0$ imposes a restriction of the functions $U(\mathbf{x})$ and $V(\mathbf{x})$ since, by the divergence theorem:

$$\int_\Omega \nabla\cdot\mathbf{u}\,dA=0\Longrightarrow\int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}\,dl=\int_{\partial\Omega}(U(\mathbf{x}),V(\mathbf{x}))\cdot\mathbf{n}(\mathbf{x})\,dl=0 \tag{A.6}$$

where $\mathbf{n}=\mathbf{n}(\mathbf{x})$ is the unit outward normal to $\partial\Omega$ at the point $\mathbf{x}\in\partial\Omega$.

APPENDIX B

Equivalence between original and modified Stokes problems

As described in Section 3.1.3.1, whenever the viscosity of a Stokesian fluid is spatially uniform, we can derive Poisson equations for each of the fluid variables:

$$\Delta p = \nabla \cdot \mathbf{f}, \tag{B.1}$$

$$\mu \Delta u = p_x - f_1, \tag{B.2}$$

$$\mu \Delta v = p_y - f_2, \tag{B.3}$$

valid for all $\mathbf{x} \in \Omega \setminus \Gamma$. In this appendix, we show that the velocities and pressure which are solutions of equations (B.1-B.3) solve in fact the incompressible Stokes equations:

$$\nabla \cdot \sigma = \mu \Delta \mathbf{u} - \nabla p = -\mathbf{f}, \tag{B.4}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{B.5}$$

for all $\mathbf{x} \in \Omega \setminus \Gamma$. In both set of equations, we assume Dirichlet boundary conditions for the velocities and the Neumann boundary conditions for the pressure given in equation (A.5):

$$u(\mathbf{x}) = U(\mathbf{x}), \tag{B.6}$$

$$v(\mathbf{x}) = V(\mathbf{x}), \tag{B.7}$$

$$\frac{\partial p}{\partial \eta}(\mathbf{x}) = f^{\eta}(\mathbf{x}) + \frac{\partial \omega}{\partial \xi}(\mathbf{x}), \tag{B.8}$$

for all $\mathbf{x} \in \partial\Omega$. We assume also that the Dirichlet boundary conditions for the velocities preserve the total fluid volume inside the domain Ω:

$$\int_{\partial\Omega} (U, V) \cdot \mathbf{n} = 0. \tag{B.9}$$

Since equations (B.2) and (B.3) are merely equation (B.4) written in component form, all we need to do is prove that the solution of equations (B.1-B.3) is such that the velocity field (u, v) obtained satisfies the incompressibility condition $\nabla \cdot (u, v) = 0$. To start, notice that from equations (B.2) and (B.3) we have:

$$\nabla \cdot (\mu \Delta \mathbf{u} - \nabla p + \mathbf{f}) = \mu \Delta (\nabla \cdot \mathbf{u}) - \Delta p + \nabla \cdot \mathbf{f} = 0, \tag{B.10}$$

but from the equation (B.1) we must then have $\mu \Delta (\nabla \cdot \mathbf{u}) = 0 \implies \Delta (\nabla \cdot \mathbf{u}) = 0$. Let $g(\mathbf{x}) = (\nabla \cdot \mathbf{u})(\mathbf{x})$ for all $\mathbf{x} \in \Omega$ and let $\mathbf{x}_0 \in \partial\Omega$ as in Appendix A. Taking the limit $\mathbf{x} \longrightarrow \mathbf{x_0}$, we can follow similar steps as we did with when we derived equation (A.4) to obtain:

$$\begin{aligned} \left(\frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2} \right) V^\eta &= \frac{\partial^2 V^\eta}{\partial \xi^2} + \frac{\partial}{\partial \eta} \left(-\frac{\partial V^\xi}{\partial \xi} + g(\mathbf{x}) \right) \\ &= \frac{\partial}{\partial \xi} \left(\frac{\partial V^\eta}{\partial \xi} - \frac{\partial V^\xi}{\partial \eta} \right) + \frac{\partial g(\mathbf{x})}{\partial \eta} \\ &= \frac{\partial \omega}{\partial \xi} + \frac{\partial g(\mathbf{x})}{\partial \eta}. \end{aligned} \tag{B.11}$$

Following again the steps of Appendix A and using equation (B.11), we find that as $\mathbf{x} \longrightarrow \mathbf{x_0}$:

$$\frac{\partial p}{\partial \eta} \longrightarrow f^\eta + \frac{\partial \omega}{\partial \xi} + \frac{\partial g(\mathbf{x})}{\partial \eta} \tag{B.12}$$

and hence, from equation (B.8), we obtain:

$$\frac{\partial g(\mathbf{x})}{\partial \eta} = \frac{\partial (\nabla \cdot \mathbf{u})}{\partial \eta} = 0 \tag{B.13}$$

for all $\mathbf{x} \in \partial\Omega$. In other words, equation (B.13) defines Neumann boundary conditions for the divergence of the velocity field $\mathbf{u}$. Since $\Delta(\nabla \cdot \mathbf{u}) = 0$, then $\nabla \cdot \mathbf{u} = c$ for some constant c for all $\mathbf{x} \in \Omega$. Finally, from (B.9), we get:

$$\int_\Omega \nabla \cdot \mathbf{u} \, dA = \int_{\partial\Omega} (U, V) \cdot \mathbf{n} \, dl = 0 \tag{B.14}$$

implying that $\nabla \cdot \mathbf{u} = 0$ for all $\mathbf{x} \in \Omega$. Therefore, the solution of equations (B.1-B.3) with Dirichlet boundary conditions for the velocities (satisfying equation (B.9)) and Neumann boundary conditions for the pressure given in equation (B.8) also solve equations (B.4-B.5) with the same boundary conditions applied.

APPENDIX C

Integration of polynomials over polygonal domains/curves

The integrals that define the matrices and right hand side elements in the Virtual Node Algorithms presented in this dissertation can be computed analytically.

In fact, let $\mathcal{P}$ be a polygonal domain consisting of d boundary segments s_i. Let (x_i, y_i) and $(x_i + \Delta x_i, y_i + \Delta y_i) = (x_{i+1}, y_{i+1})$ be the coordinates of the end points of s_i. If $p_2(x, y) = a + bx + cy + dx^2 + exy + fy^2$ is a quadratic function of x and y, then

$$\begin{aligned}
\int_{\mathcal{P}} p_2(x,y)\, dA = \sum_{i=0}^{d-1} \Delta y_i \Bigg[& a \left(\frac{\Delta x_i}{2} + x_i \right) \\
& + b \left(\frac{(\Delta x_i)^2}{6} + \frac{(\Delta x_i) x_i}{2} + \frac{x_i^2}{2} \right) \\
& + c \left(\frac{(\Delta x_i)(\Delta y_i)}{3} + \frac{(\Delta x_i) y_i + (\Delta y_i) x_i}{2} + x_i y_i \right) \\
& + d \left(\frac{(\Delta x_i)^3}{12} + \frac{x_i (\Delta x_i)^2}{3} + \frac{x_i^2 (\Delta x_i)}{2} + \frac{x_i^3}{3} \right) \\
& + e \Bigg(\frac{(\Delta x_i)^2 (\Delta y_i)}{8} + \frac{(\Delta x_i)(\Delta y_i) x_i}{3} + \frac{x_i^2 (\Delta y_i)}{4} + \frac{(\Delta x_i)^2 y_i}{6} \\
& \quad + \frac{(\Delta x_i) x_i y_i}{2} + \frac{x_i^2 y_i}{2} \Bigg) \\
& + f \Bigg(\frac{(\Delta x_i)(\Delta y_i)^2}{4} + \frac{2(\Delta x_i)(\Delta y_i) y_i}{3} + \frac{(\Delta x_i) y_i^2}{2} + \frac{x_i (\Delta y_i)^2}{3} \\
& \quad + (\Delta y_i) y_i x_i + x_i y_i^2 \Bigg) \Bigg].
\end{aligned} \tag{C.1}$$

All the functions that need to be integrated when computing the matrices and right hand

side elements of the VNAs are at most quadratic over the relevant polygonal domains. We use the procedure above to compute these integrals. Equation (C.1) is derived as in [1] by applying the divergence theorem:

$$\begin{aligned}
\int_{\mathcal{P}} p_2(x,y)\, dA &= \int_{\mathcal{P}} \left(a + bx + cy + dx^2 + exy + fy^2\right)\, dA \\
&= \int_{\mathcal{P}} \nabla \cdot \left(ax + \frac{bx^2}{2} + cxy + \frac{dx^3}{3} + \frac{ex^2y}{2} + fxy^2, 0\right)\, dA \\
&= \sum_{i=1}^{d} \int_{s_i} \left(ax + \frac{bx^2}{2} + cxy + \frac{dx^3}{3} + \frac{ex^2y}{2} + fxy^2, 0\right) \cdot \tilde{\mathbf{n}}_i dl \\
&= \sum_{i=1}^{d} \int_{s_i} \left(ax + \frac{bx^2}{2} + cxy + \frac{dx^3}{3} + \frac{ex^2y}{2} + fxy^2\right) \tilde{n}_i^1 dl
\end{aligned} \tag{C.2}$$

where $\tilde{n}_i^1$ is the x component of the normal $\tilde{\mathbf{n}}_i$ to the segment s_i. The computation of the line integrals is simplified if we parametrize each segment s_i as $\mathbf{p}(s) = (x_i + s\Delta x_i, y_i + s\Delta y_i)$ for $s \in [0,1]$.

Line integrals of quadratic polynomials $p_2(x,y)$ over the oriented polygonal boundaries are also needed for the right-hand-side terms and for the entries in the jump constraint matrices. These can also be computed analytically. For a polygonal curve $\mathcal{S}$ with d segments, we get:

$$\begin{aligned}
\int_{\mathcal{S}} p_2(x,y) dl = \sum_{i=1}^{d} l_i \Bigg[& a + b\left(x_i + \frac{\Delta x_i}{2}\right) + c\left(y_i + \frac{\Delta y_i}{2}\right) \\
& + d\left(x_i^2 + x_i(\Delta x_i) + \frac{(\Delta x_i)^2}{3}\right) \\
& + e\left(x_i y_i + \frac{x_i(\Delta y_i)}{2} + \frac{y_i(\Delta x_i)}{2} + \frac{(\Delta x_i)(\Delta y_i)}{3}\right) \\
& + f\left(y_i^2 + y_i(\Delta y_i) + \frac{(\Delta y_i)^2}{3}\right) \Bigg],
\end{aligned} \tag{C.3}$$

where $l_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}$ is the length of the ith segment of $\mathcal{S}$.

References

[1] J. Bedrossian, J. H. von Brecht, S. Zhu, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains, J. Comput. Phys. 229 (2010) 6405–6426.

[2] I. M. Cohen, P. K. Kundu, Fluid mechanics, 3rd Edition, Academic Press, 2004.

[3] J. Hellrung, L. Wang, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys. 231 (4) (2012) 2015–2048.

[4] A. Lew, G. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, Internat. J. Numer. Methods Engrg. 76 (4) (2008) 427–454.

[5] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[6] C. S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.

[7] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–+.

[8] D. M. McQueen, C. S. Peskin, A three-dimensional computational method for blood flow in the heart. ii. contractile fibers, J. Comput. Phys. 82 (1989) 289–297.

[9] C. S. Peskin, D. M. McQueen, A three-dimensional computational method for blood flow in the heart i. immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (2) (1989) 372 – 405.

[10] C. S. Peskin, D. M. McQueen, Modeling prosthetic heart valves for numerical analysis of blood flow in the heart, J. Comput. Phys. 37 (1) (1980) 113 – 132.

[11] L. J. Fauci, C. S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.

[12] A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, J. Comput. Phys. 56 (1) (1984) 111 – 134.

[13] A. L. Fogelson, Continuum models of platelet aggregation: formulation and mechanical properties, SIAM J. Appl. Math. 52 (1992) 1089–1110.

[14] J. M. Teran, L. Fauci, M. Shelley, Viscoelastic fluid response can increase the speed and efficiency of a free swimmer, Phys. Rev. Lett. 104 (3) (2010) 038101.

[15] J. Teran, L. Fauci, M. Shelley, Peristaltic pumping and irreversibility of a Stokesian viscoelastic fluid, Phys. Fluids 20 (7) (2008) 073101–+.

[16] L. J. Fauci, Interaction of oscillating filaments: a computational study, J. Comput. Phys. 86 (1990) 294–313.

[17] A. Lima E Silva, A. Silveira-Neto, J. J. R. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, J. Comput. Phys. 189 (2003) 351–370.

[18] J. M. Teran, C. S. Peskin, Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain, SIAM J. Sci. Comput. 31 (2009) 3404–3416.

[19] R. P. Beyer, Jr., A computational model of the cochlea using the immersed boundary method, J. Comput. Phys. 98 (1992) 145–162.

[20] R. Dillon, L. Fauci, A. Fogelson, D. Gaver, III, Modeling biofilm processes using the immersed boundary method, J. Comput. Phys. 129 (1996) 57–73.

[21] J. M. Stockie, S. I. Green, Simulating the motion of flexible pulp fibres using the immersed boundary method, J. Comput. Phys. 147 (1998) 147–165.

[22] C. S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.

[23] Z. Li, M.-C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, J. Comput. Phys. 171 (2) (2001) 822 – 842.

[24] B. E. Griffith, C. S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.

[25] B. E. Griffith, R. D. Hornung, D. M. McQueen, C. S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, J. Comput. Phys. 223 (1) (2007) 10 – 49.

[26] A. M. Roma, C. S. Peskin, M. J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (2) (1999) 509 – 534.

[27] C. S. Peskin, B. F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.

[28] R. Cortez, M. Minion, The blob projection method for immersed boundary problems, J. Comput. Phys. 161 (2000) 428–453.

[29] R. J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.

[30] R. J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.

[31] Z. Tan, D. V. Le, K. M. Lim, B. C. Khoo, An immersed interface method for the incompressible Navier-Stokes equations with discontinuous viscosity across the interface, SIAM J. Sci. Comput. 31 (2009) 1798–1819.

[32] Z. Tan, D. V. Le, Z. Li, K. M. Lim, B. C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane, J. Comput. Phys. 227 (2008) 9955–9983.

[33] S. Xu, Z. J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.

[34] D. V. Le, B. C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109–138.

[35] L. Lee, R. J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.

[36] T. Y. Hou, Z. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, J. Comput. Phys. 134 (1997) 236–252.

[37] Z. Li, K. Ito, M.-C. Lai, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, Comput. & Fluids 36 (3) (2007) 622 – 635.

[38] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 225 (1) (2007) 1066 – 1099.

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

[40] S. Xu, Z. J. Wang, A 3d immersed interface method for fluid-solid interaction, Comput. Methods Appl. Mech. Engrg. 197 (25-28) (2008) 2068 – 2086.

[41] Z. Li, An overview of the immersed interface method and its applications, Taiwanese J. Math. 7 (1) (2003) 1 – 49.

[42] X.-D. Liu, R. P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.

[43] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.

[44] M. Kang, R. P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[45] M. Hyman, Non-iterative numerical solution of boundary-value problems, Appl. Scientific Research, Section B 2 (1) (1952) 325–351.

[46] V. Sau'lev, On solving boundary value problems on high-performance computers by fictitious domain methods, Siberian Mathematical J. 4 (1963) 912–925.

[47] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (5) (1999) 755–794.

[48] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 112 (1-4) (1994) 133–148.

[49] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flows, J. Comput. Phys. 169 (2001) 363–426.

[50] L. Parussini, V. Pediroda, Fictitious domain approach with hp-finite element approximation for incompressible fluid flow, J. Comput. Phys. 228 (2009) 3891–3910.

[51] L. Parussini, Fictitious domain approach via Lagrange multipliers with least squares spectral element method, J. Sci. Comput. 37 (2008) 316–335.

[52] F. Bertrand, P. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, Internat. J. Numer. Methods Fluids 25 (6) (1997) 719–736.

[53] J. M. Teran, C. S. Peskin, Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain, SIAM J. Sci. Comput. 31 (5) (2009) 3404–3416.

[54] G. Biros, L. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (1) (2004) 317–348.

[55] V. Rutka, A staggered grid-based explicit jump immersed interface method for two-dimensional Stokes flows, Internat. J. Numer. Methods Fluids 57 (10) (2008) 1527–1543.

[56] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, Internat. J. Numer. Methods Engrg. 48 (12) (2000) 1741–1760.

[57] N. Sukumar, D. Chopp, N. Moës, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, Comput. Methods Appl. Mech. Engrg. 190 (46-47) (2001) 6183–6200.

[58] A. Almgren, J. Bell, P. Colella, T. Marthaler, A Cartesian grid projection method for the incompressible euler equations in complex geometries, SIAM J. Sci. Comput. 18 (5) (1997) 1289–1309.

[59] N. Moës, E. Béchet, M. Tourbier, Imposing Dirichlet boundary conditions in the extended finite element method, Internat. J. Numer. Methods Engrg. 67 (12) (2006) 1641–1669.

[60] J. Dolbow, A. Devan, Enrichment of enhanced assumed strain approximations for representing strong discontinuities: addressing volumetric incompressibility and the discontinuous patch test, Internat. J. Numer. Methods Engrg. 59 (1) (2004) 47–67.

[61] G. Wagner, N. Moës, W. Liu, T. Belytschko, The extended finite element method for rigid particles in Stokes flow, Internat. J. Numer. Methods Engrg. 51 (3) (2001) 293–313.

[62] R. Becker, E. Burman, P. Hansbo, A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity, Comput. Methods Appl. Mech. Engrg. 198 (41-44) (2009) 3352–3360.

[63] A. Coppola-Owen, R. Codina, Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions, Internat. J. Numer. Methods Fluids 49 (12) (2005) 1287–1304.

[64] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, J. Appl. Math. 70 (1) (2003) 10–17.

[65] A. Gerstenberger, W. Wall, An extended finite element method/Lagrange multiplier based approach for fluid-structure interaction, Comput. Methods Appl. Mech. Engrg. 197 (19-20) (2008) 1699–1714.

[66] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface Cartesian grid method i: an easily implemented technique for 3d moving boundary computations, J. Comput. Phys. 210 (1) (2005) 1–31.

[67] Y. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (2009) 8807–8829.

[68] C. Batty, F. Bertails, R. Bridson, A fast variational framework for accurate solid-fluid coupling, ACM Transactions on Graphics 26.

[69] I. Bijelonja, I. Demirdžić, S. Muzaferija, A finite volume method for incompressible linear elasticity, Comput. Methods Appl. Mech. Engrg. 195 (44-47) (2006) 6378–6390.

[70] L. Beirão da Veiga, V. Gyrya, K. Lipnikov, G. Manzini, Mimetic finite difference method for the Stokes problem on polygonal meshes, J. Comput. Phys. 228 (19) (2009) 7215–7232.

[71] P. Barton, D. Drikakis, An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces, J. Comput. Phys. 229 (15) (2010) 5518–5540.

[72] D. Hill, D. Pullin, M. Ortiz, D. Meiron, An Eulerian hybrid weno centered-difference solver for elastic-plastic solids, J. Comput. Phys. 229 (24) (2010) 9053–9072.

[73] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.

[74] H. S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries: 345, J. Comput. Phys. 174 (2001) 380–.

[75] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.

[76] G. Biros, L. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (2004) 317–348.

[77] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (2) (1984) pp. 285–299.

[78] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, J. Comput. Phys. 176 (2) (2002) 231 – 275.

[79] K. Ito, Z. Li, Interface conditions for Stokes equations with a discontinuous viscosity and surface sources, Appl. Math. Letters 19 (3) (2006) 229 – 234.

[80] S. Xu, Z. J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, SIAM J. Sci. Comput 2006.

[81] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[82] H. Han, X. Wu, A new mixed finite element formulation and the MAC method for the Stokes equations, SIAM J. Numer. Anal. 35 (2) (1998) 560–571.

[83] D. C. Assêncio, J. M. Teran, A second order virtual node algorithm for Stokes flow problems with interfacial forces and irregular domains, To appear.