Immersed Interface Methods for Incompressible Flow with Moving Interfaces

Long Lee

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2002

Program Authorized to Offer Degree: Department of Applied Mathematics

University of Washington

Abstract

Immersed Interface Methods for Incompressible Flow with Moving Interfaces

by Long Lee

Chair of Supervisory Committee:

Professor Randall J. LeVeque
Department of Applied Mathematics

We introduce a class of immersed interface methods for incompressible Navier-Stokes equations with singular forces. The algorithms are developed using a new hybrid finite volume/finite difference projection method. This projection method uses the high-resolution finite-volume method from *CLAWPACK* to advect the cell-centered velocity using the divergence-free edge velocity. Then a finite difference method is used for the diffusion and the projection step. This projection method is second-order accurate and is suitable for flow at high Reynolds numbers or problems containing discontinuous conserved quantities.

The immersed interface method developed in this thesis is motivated by Peskin's immersed boundary (IB) method, and allows one to model the motion of flexible membranes or other structures immersed in viscous incompressible fluid using a fluid solver on a fixed Cartesian grid. The IB method uses a set of discrete delta functions to spread the entire singular force exerted by the immersed boundary to the nearby fluid grid points. Our method instead incorporates part of this force into jump conditions for the pressure, avoiding discrete dipole terms that adversely affect the accuracy near the immersed boundary. This has been implemented by imposing the correct pressure jump in the course of solving the Poisson problem in the projection step. This gives sharp resolution of the pressure across the interface and better volume conservation than the traditional IB method. A fully implicit method that allows us to model immersed elastic boundaries with mass is also presented.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am ever so grateful to my mentor and thesis advisor, Randall J. LeVeque. I cannot thank him enough for his endless patience, frequent encouragement, and generous support. Randy's invaluable insight into many of the numerical and physical problems was the key reason that I was able to start and finish this research project. I have greatly benefited from countless discussions with Randy. His capability of explaining complicated concepts through simple examples made the process of this research project so much easier.

I would like to thank Charles Peskin of the Courant Institute for useful discussion about adding mass to the membrane and Jamal Mohd-Yusof of Los Alamos for pointing out an inconsistency in an earlier version of this algorithm. For the rest of my committee members, Loyce Adams, Peter Schmid, and Ka-Kit Tung, I would like to thank them for their comments and suggestions. I also would like to thank Harland M. Glaz in the Department of Mathematics at the University of Maryland. It was Harland who first led me into the field of computational fluid dynamics and encouraged me to continue my research with Randy. My friend and officemate James A. Rossmanith gave me many suggestions and assistance through these years. I would like to thank him for his help.

During my Ph.D study, my family have been very supportive, especially my parents. Their unconditional support has given me the strength to finish my study. I am very grateful to them. Finally I would like to thank Yi-Ling for being my friend and soul mate for more than a decade. She is my inspiration for being a better person in society.

This work was supported in part by NSF grant DMS-9803442 and DOE grants DE-FG03-00ER2592 and DE-FC02-01ER25474.

Chapter 1

INTRODUCTION

There are two themes in this thesis. First we develop a new class of projection methods for incompressible Navier-Stokes equations. This projection method is suitable for flow at high Reynolds numbers, $Re = 10^4$ for example, and for problems containing discontinuous conserved quantities. The main theme of this thesis is introducing an immersed interface method for incompressible flow with elastic boundaries or surface tension using this new projection method. For incompressible flow with interfaces, the solution typically has a jump in pressure across the interface. This new immersed interface method incorporates the jump conditions for the pressure into the projection method and gives sharp resolution of the solution across the interface.

1.1 Types of interface problems

Moving interface problems have intrigued the scientific community for decades. For many applications, the immersed interface is a material interface, such as an elastic membrane, and the fluid has uniform density and viscosity across the interface. There are also other interesting fluid problems involving fluid interfaces with no separating material, only surface tension. Furthermore for many of those problems, the densities and viscosities are discontinuous across the interfaces. An example is the free surface that separates two fluids. For such two-fluid problems the surface tension between the fluids acts like the elastic force in the elastic membrane, but purely in the normal direction. Often in order to make a fluid model with interfaces close to reality we need to model fluids with different densities and viscosities across the interface.

In general, fluid interface problems can be classified into the following categories.

- Material interface problems: In this type of problem the interface is a surface consisting of real materials instead of a contact discontinuity between fluids. A very thin flexible membrane that separates the fluid and moves along with the fluid is an example of a material interface. Such fluid-flexible bodies systems have been intensively studied in the literature. The interaction between the material interface and the fluid creates rich dynamics that is an interesting subject to many scientists and mathematicians. An example of a moving fluid-material interface problem is Peskin's heart valves model. In this model, the heart valve and heart wall are modeled as elastic membranes that move and interact with the blood flow. In this thesis, we mainly consider the material interface problem.

- Free surface problems: Examples for free surface problems are motion of vortex sheets with surface tension and multicomponent fluids with surface tension, including Kelvin-Helmholtz and Rayleigh-Taylor instability, the generation of capillary waves on the free surface, bubbly flow, and Hele-Show flows involving pattern formation through the Saffman-Taylor instability. In addition, there is a class of free surface problems involving motion by mean curvature flow, such as the evolution of crystal growth and the formation of thin films. Some references for free surface problems can be found in [26, 52].

- Fluid-structure interaction and multiphase material: These interface problems concern the interaction between fluid and solid structure, or the interaction between solid structures. Examples are offshore structures, acoustical media, liquid or gas storage tanks, pipeline systems, and nuclear reactors.

1.2 *Problem formulation*

A moving interface in incompressible flow is a surface that has codimension one, for example a free interface between two fluids or a membrane immersed in a fluid. For this thesis, we are interested in problems in which this interface exerts a singular force on the fluid, e.g. surface tension or an elastic force. Consider the two-dimensional incompressible Navier-Stokes equations with a singular force as follows

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p &= \mu\nabla^2\boldsymbol{u} + \boldsymbol{F} \\ \nabla\cdot\boldsymbol{u} &= 0 \\ \boldsymbol{u} &= \boldsymbol{b} \quad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

where the singular force has the form

$$\boldsymbol{F}(x,t) = \int_{s_0}^{s_1} \boldsymbol{f}(s,t)\,\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,ds, \tag{1.2}$$

where $\boldsymbol{X}(s,t)$ is the position of the interface parameterized by s, which we assume is the arclength for now. Detailed discussion about the parameter s is in Chapter 5. The force $\boldsymbol{F}$ is singular, since we integrate a two-dimensional delta function along a one dimensional surface. We assume that the density is constant and equal to one and the viscosity μ is uniform across the interface, although it is possible to relax these conditions in the future. The pressure p is typically discontinuous across the interface. The velocity $\boldsymbol{u}$ is continuous, but its normal derivative typically has jumps.

Since the fluid is viscous and the velocity is continuous across the interface, the interface is moving with the fluid. We can write down the relation between the velocity of the interface and its position as follows

$$\frac{\partial \boldsymbol{X}(s,t)}{\partial t} = \boldsymbol{u}\big(\boldsymbol{X}(s,t),t\big) = \int_{x_l}^{x_u}\int_{y_l}^{y_u} \boldsymbol{u}(\boldsymbol{x},t)\,\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,dx\,dy. \tag{1.3}$$

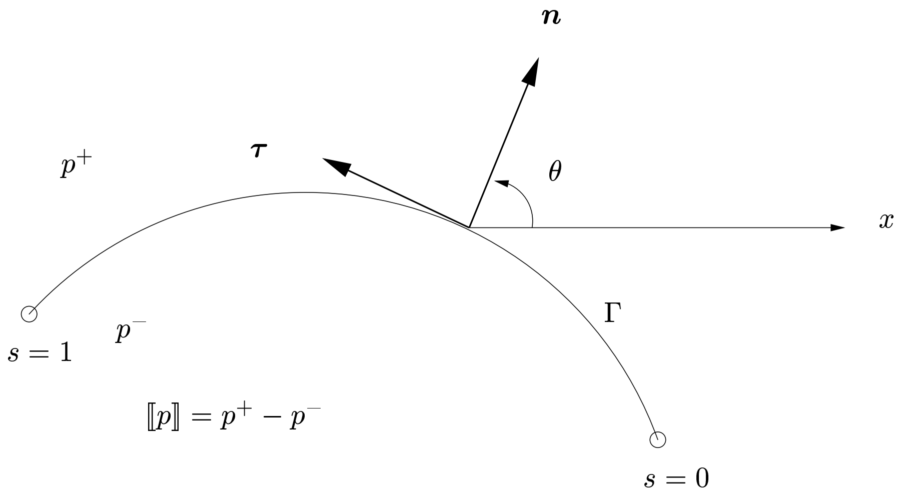


Figure 1.1: The normal and tangent vector of interface Γ. $[\![p]\!]$ is the jump in pressure across Γ.

The next step is to establish the relation between the position of the elastic interface and the singular force exerted on the fluid by this interface. We consider an elastic membrane separating fluids in two-dimensional space. The unit tangent on the surface is

$$\boldsymbol{\tau}(s,t) = \frac{\partial \boldsymbol{X}}{\partial s} \Big/ \left| \frac{\partial \boldsymbol{X}}{\partial s} \right| \tag{1.4}$$

and the tension of the membrane is defined by Hooke's Law

$$T(s,t) = T_0 \left(\left| \frac{\partial \boldsymbol{X}(s,t)}{\partial s} \right| - 1 \right), \tag{1.5}$$

where T_0 is the stiffness constant and s measures arclength of the unstrained membrane with $0 \leq s \leq L_0$ where L_0 is the total length of the unstrained membrane.

In two-dimensional space if we denote $\dfrac{\partial \boldsymbol{X}}{\partial s} = \left(\dfrac{\partial X}{\partial s}, \dfrac{\partial Y}{\partial s} \right)$, and let θ be the angle between the x axis and the outward normal $\boldsymbol{n}$ as shown in Figure 1.1, then the tangent vector and the outward normal are defined as

$$\boldsymbol{\tau} = (-\sin\theta, \cos\theta), \qquad \boldsymbol{n} = (\cos\theta, \sin\theta), \tag{1.6}$$

where

$$\cos\theta = \frac{\partial Y}{\partial s} \Big/ \left| \frac{\partial \boldsymbol{X}}{\partial s} \right|, \qquad \sin\theta = -\frac{\partial X}{\partial s} \Big/ \left| \frac{\partial \boldsymbol{X}}{\partial s} \right|. \tag{1.7}$$

Figure 1.1 shows the normal and tangent vector defined in this thesis. The normal is pointing in the outward direction of the interface Γ and the jump in pressure is defined by $[\![p]\!] = p^+ - p^-$.

The membrane is assumed to have the mass density $m(s)$ with respect to the measure ds. At each point of the interface, Newton's Second Law states that mass times acceleration is a sum of all forces, which in our case are the tension force and the force exerted from the fluid to the membrane

$$m(s)\frac{\partial^2 \boldsymbol{X}}{\partial t^2} = \boldsymbol{F}_T(s,t) + \boldsymbol{F}_R(s,t), \tag{1.8}$$

where $\boldsymbol{F}_T$ is the tension force and can be represented by

$$\boldsymbol{F}_T = \frac{\partial}{\partial s}(T\boldsymbol{\tau}). \tag{1.9}$$

The force $\boldsymbol{F}_R$ is measured in terms of the jumps in pressure and in the derivative of the velocity in the fluid as

$$\boldsymbol{F}_R = \left[\!\left[-p(s,t)\boldsymbol{n} + \mu\frac{\partial \boldsymbol{u}(s,t)}{\partial \boldsymbol{n}} \right]\!\right]. \tag{1.10}$$

The quantity $-p(s,t)\boldsymbol{n} + \mu\dfrac{\partial \boldsymbol{u}(s,t)}{\partial \boldsymbol{n}}$ is known as the normal stresses of the fluid element, in terms of the fluid stress tensor. The jump of equation (1.10) is what we need to know in our algorithm. However, it is not possible to compute this jump directly from the fluid equations. Instead we obtain the jump by computing the tension force and the inertial term on the left hand side of equation (1.8). To see how this works, we denote the force exerted on the fluid from the membrane $\tilde{\boldsymbol{f}}$ and

$$-\tilde{\boldsymbol{f}}(s,t) = \left[\!\left[-p(s,t)\boldsymbol{n} + \mu\frac{\partial \boldsymbol{u}(s,t)}{\partial \boldsymbol{n}} \right]\!\right]. \tag{1.11}$$

From equation (1.8), we have

$$\tilde{\boldsymbol{f}}(s,t) = \frac{\partial}{\partial s}(T\boldsymbol{\tau}) - m(s)\frac{\partial^2 \boldsymbol{X}}{\partial t^2}. \tag{1.12}$$

An important special case is $m(s) = 0$, a massless membrane, as is often used in practice. In this case, the force density (1.12) reduces to

$$\boldsymbol{f}(s,t) = \frac{\partial}{\partial s}(T\boldsymbol{\tau}). \tag{1.13}$$

More discussion on the singular force density is in Section 4.5 and derivations of the jump conditions is in Chapter 5.

1.3 Immersed boundary and immersed interface methods

One popular numerical method for the above problem is the immersed boundary method developed by Peskin and McQueen [77, 78, 80]. The immersed boundary method replaces (1.2) by a sum as follows

$$\boldsymbol{F}_{i,j} = \sum_k \boldsymbol{f}_k\, D_{i,j}(\boldsymbol{X}_k)\Delta s, \tag{1.14}$$

where $D_{i,j}(\boldsymbol{X})$ is a two-dimensional discrete delta function and $\boldsymbol{f}_k$ is a discretization of (1.12) or (1.13). This method has profound influence on many numerical algorithms and has been implemented for many applications in fluid. The immersed boundary method and methods related to it are reviewed in Chapter 2.

A major contribution of this thesis is introducing an immersed interface method that improves the immersed boundary method in terms of resolution, order of accuracy, and volume conservation. The immersed interface method developed in this thesis is for incompressible Navier-Stokes equations with singular forces (1.1). The immersed interface method for Stokes flow with singular forces was introduced by LeVeque and Li [64]. They write the two-dimensional unsteady Stokes equations with a singular force as the form

$$\nabla p(\boldsymbol{x},t) = \mu\nabla^2\boldsymbol{u}(\boldsymbol{x},t) + \boldsymbol{F}(\boldsymbol{x},t) \tag{1.15}$$

$$\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t) = 0. \tag{1.16}$$

The time evolution of the solution is implicitly determined by the external singular force $\boldsymbol{F}(\boldsymbol{x},t)$.

By (1.11)−(1.13) and see that a nonzero singular force corresponds to jumps across the interface and so solutions of the systems (1.1) and the Stokes equations, (1.15) and (1.16), have the property

- p is discontinuous across the interface and the jump in pressure is proportional to the normal force strength.

- $\boldsymbol{u}$ is continuous but not smooth and the jump of the normal derivative of the velocity is proportional to the tangential force strength.

Detailed derivations of the jump conditions for different fluids are given in Chapter 5.

Assuming that the viscosity is uniform across the interface and applying the divergence operator to (1.15) gives

$$\nabla^2 p = \nabla\cdot\boldsymbol{F}, \tag{1.17}$$

where ∇^2 is the Laplacian. Since $\boldsymbol{F}$ is a singular force, the divergence of a singular force is a dipole source. In [64], equation (1.17) is solved by

$$\nabla^2 p = C, \tag{1.18}$$

where C is a correction term determined by the technique developed in [63]. This technique incorporates jump conditions in the pressure into Taylor series expansions to derive the difference equations with the correction C on the right hand side for equation (1.17), so that this equation can be solved by a standard fast Poisson solver. We briefly review this technique and its history in Chapter 2. After the pressure is obtained by (1.18), the velocity in (1.15) is solved by a difference formula with a correction that is derived by jump conditions in the derivative of the velocity and jump conditions in the pressure.

The immersed interface method in this thesis applies to the full Navier-Stokes equations (1.1), and is developed by means of a projection method. The numerical algorithm for this method consists of two steps, a transport step and a projection step. Denoting the singular force $\boldsymbol{F}$ by $\boldsymbol{F} = \boldsymbol{F}_1 + \boldsymbol{F}_2$, we describe the algorithm as follows:

Transport step. Advance $\boldsymbol{U}^n$ to $\boldsymbol{U}^*$ by solving the advection-diffusion equation.

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \mu \nabla^2 \boldsymbol{u} + \boldsymbol{F}_1 \tag{1.19}$$

over time Δt. The force $\boldsymbol{F}_1$ is spread to the fluid by the approximation (1.14) using a two dimensional discrete delta function.

Projection step. Advance $\boldsymbol{U}^*$ to $\boldsymbol{U}^{n+1}$ by

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* - \Delta t \nabla p^{n+1} + \Delta t \boldsymbol{F}_2. \tag{1.20}$$

For the standard immersed boundary method, $\boldsymbol{F}_1$ is the entire force $\boldsymbol{F}$ and $\boldsymbol{F}_2$ is zero. The key new feature of the algorithm we are proposing is that we set $\boldsymbol{F}_1$ to be the tangential direction force and $\boldsymbol{F}_2$ be the normal direction force. The force $\boldsymbol{F}_2$ is imposed by enforcing appropriate jump conditions on the source to a Poisson problem. To see this, applying the discrete divergence operator to (1.20) and using the fact that $\boldsymbol{U}^{n+1}$ is discrete divergence free, we have

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{U}^* + \nabla \cdot \boldsymbol{F}_2. \tag{1.21}$$

Because $\boldsymbol{F}_2$ is a singular force in the form of a delta function, the divergence of $\boldsymbol{F}_2$ is a dipole. Therefore equation (1.21) is a Poisson problem with a dipole source along the interface resulting in the solution p^{n+1} being discontinuous. The correct jump in pressure is determined from the force exerted by the membrane and is imposed in the course of solving this Poisson problem. A correction term is derived on the right hand side of the difference equation and the problem can be solved by a standard fast solver. The detailed discussion of the correction term is in Chapter 4.

The advection part of equation (1.19) is implemented based on the high-resolution finite volume method from *CLAWPACK*. A Crank-Nicolson discretization is applied to the diffusion while $\boldsymbol{F}_1$ is present as an external force. We review some existing projection methods in Chapter 2. The new projection method used for the immersed interface method is discussed in Chapter 3.

1.4 Overview of this thesis

This thesis is organized by the following order. In Chapter 2, we review the projection methods for incompressible Navier-Stokes equations with general external forces and some existing numerical algorithms for incompressible flow with singular forces. After understanding the principal mechanism of the projection method, in Chapter 3 we introduce a new projection method using high-resolution algorithms from *CLAWPACK*. Several test problems are presented to show the strength of this method. Chapter 4 contains the detailed derivation and discussion of the new immersed interface methods. Several test problems with immersed elastic membranes are also presented in this chapter. Part of this chapter appear in the paper [58]. Chapter 5 gives a detailed mathematical background of the jump conditions for different fluids. Finally, in Chapter 6 we summarize the current work and give an outlook for future work.

Chapter 2

EXISTING METHODS FOR INCOMPRESSIBLE FLOW AND INTERFACE PROBLEMS

In the beginning of this chapter, we review the projection methods for the incompressible Navier-Stokes equations with general external forces. After reviewing the projection methods, we discuss some existing numerical methods for incompressible flow with moving interfaces that exert singular forces on the fluid. The existing methods include the immersed boundary method, the immersed interface method for Stokes flow, the blob projection method, and the ghost fluid method.

2.1 Projection methods

Chorin, in a series of papers [18, 19, 20] introduced a practical fractional step method for solving viscous incompressible Navier-Stokes equations. Similar ideas were independently introduced by Temam [95]. This fractional step method, or *projection method*, computes an intermediate velocity without regard to the divergence constraint and then *projects* this velocity onto the divergence-free subspace. Many other different forms of projection methods were developed after those of Chorin and Temam. We discuss two common second-order projection methods and a version of generalized projection method, the gauge method, in this chapter.

We consider the unsteady constant-density incompressible Navier-Stokes equations

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p &= \frac{1}{Re}\nabla^2 \boldsymbol{u} + \boldsymbol{F} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \tag{2.1}$$

on a domain Ω, where $\boldsymbol{u}$ represents the velocity field. p is the hydrodynamic pressure, $\boldsymbol{F}$ represents any external forces, and Re is the Reynolds number. Here we assume the density is constant and equal to one. The boundary conditions for $\boldsymbol{u}$ are either Dirichlet or Neumann conditions. There are no boundary conditions for p. For simplicity, we consider the Dirichlet boundary conditions

$$\boldsymbol{u} = \boldsymbol{b} \qquad \text{on} \qquad \partial\Omega$$

for our analysis.

2.1.1 Projection method 1

The first projection method discussed in this section is a finite difference method introduced by Kim and Moin [55]. This method is similar to the first-order scheme of Chorin. It first computes the intermediate velocity without the gradient pressure term. It then imposes the incompressibility by solving a Poisson equation. The method can be written in a semi-discrete form:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^{n+1/2} \cdot \nabla)\boldsymbol{u}^{n+1/2} = \frac{1}{2Re}(\nabla^2 \boldsymbol{u}^* + \nabla^2 \boldsymbol{u}^n) + \boldsymbol{F} \tag{2.2}$$

$$\boldsymbol{u}^* = \boldsymbol{b} + \triangle t \nabla \phi^n \qquad \text{on} \qquad \partial\Omega \tag{2.3}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\triangle t} = -\nabla \phi^{n+1} \tag{2.4}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0 \tag{2.5}$$

$$\boldsymbol{n} \cdot \boldsymbol{u}^{n+1} = \boldsymbol{n} \cdot \boldsymbol{b} \qquad \text{on} \qquad \partial\Omega, \tag{2.6}$$

where $\boldsymbol{n}$ is the unit normal. The nonlinear convection term $(\boldsymbol{u}^{n+1/2} \cdot \nabla)\boldsymbol{u}^{n+1/2}$ can be approximated, for example, by an explicit Adams-Bashforth formula $\frac{3}{2}(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \frac{1}{2}(\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}$.

The novel idea of this scheme is the treatment for the boundary condition for the intermediate velocity $\boldsymbol{u}^*$. Instead of $\boldsymbol{u}^* = \boldsymbol{b}$, Kim and Moin, following the paper of LeVeque and Oliger [66], suggest (2.3) to be the boundary condition for the intermediate velocity field $\boldsymbol{u}^*$. Their computational results show the evidence of second-order accuracy for the scheme with such modification. In fact, what one really wants is $\boldsymbol{u}^* = \boldsymbol{b} + \triangle t \nabla \phi^{n+1}$ on $\partial\Omega$, but one cannot have ϕ^{n+1} until (2.4) is computed. As for the pressure p, substituting (2.4) into (2.2), eliminating $\boldsymbol{u}^*$ and comparing with (2.1), shows that p^{n+1} can be obtained from ϕ^{n+1} through the relationship

$$p^{n+1} = \phi^{n+1} - \frac{\triangle t}{2Re} \nabla^2 \phi^{n+1}. \tag{2.7}$$

To compute ϕ^{n+1}, we apply the divergence operator $\nabla \cdot$ to (2.4) and use (2.5). We then have the Poisson equation

$$\nabla^2 \phi^{n+1} = \frac{\nabla \cdot \boldsymbol{u}^*}{\triangle t}. \tag{2.8}$$

If we take the normal component of both sides of (2.4) and apply (2.3), (2.5), and (2.6), we have the numerical boundary condition for ϕ^{n+1}:

$$\frac{\partial \phi^{n+1}}{\partial \boldsymbol{n}} = \frac{\partial \phi^n}{\partial \boldsymbol{n}} \qquad \text{on} \qquad \partial\Omega. \tag{2.9}$$

If $\dfrac{\partial \phi^0}{\partial \boldsymbol{n}}$ is the zero vector, we have the homogeneous Neumann condition for (2.8)

$$\frac{\partial \phi^{n+1}}{\partial \boldsymbol{n}} = 0 \qquad \text{on} \qquad \partial\Omega. \tag{2.10}$$

Brown, Cortez, and Minion [8] suggest that the boundary condition for $\boldsymbol{u}^*$ satisfies (2.3) only in the tangential direction. A convenient choice of the boundary condition in the normal direction is

$$\boldsymbol{n} \cdot \boldsymbol{u}^* = \boldsymbol{n} \cdot \boldsymbol{b}, \tag{2.11}$$

which also results in the homogeneous Neumann boundary condition (2.10).

In general, if we take the normal component of both sides of (2.4), we have the boundary condition

$$\frac{\partial \phi^{n+1}}{\partial \boldsymbol{n}} = -\frac{1}{\triangle t}(\boldsymbol{b} - \boldsymbol{u}^*) \cdot \boldsymbol{n} \qquad \text{on} \qquad \partial\Omega. \tag{2.12}$$

A compatibility condition for the Neumann problem (2.8), (2.12) is

$$\frac{1}{\triangle t}\int_{\Omega} \nabla \cdot \boldsymbol{u}^* \, ds = -\frac{1}{\triangle t}\int_{\partial\Omega} (\boldsymbol{b} - \boldsymbol{u}^*) \cdot \boldsymbol{n} \, ds. \tag{2.13}$$

From the divergence theorem, we have

$$\int_{\Omega} \nabla \cdot \boldsymbol{u}^* \, ds = \int_{\partial\Omega} \boldsymbol{u}^* \cdot \boldsymbol{n} \, ds. \tag{2.14}$$

This implies that the compatibility condition for the Neumann problem to have a solution is

$$\int_{\partial\Omega} \boldsymbol{b} \cdot \boldsymbol{n} \, ds = 0. \tag{2.15}$$

Some detailed analysis for this projection method can be seen in [8, 31, 32, 93].

2.1.2 Projection method 2

The second method is the pressure-correction scheme which is first introduced by Van Kan [102]. Later, Bell, Colella, and Glaz (BCG) used Godunov's methodology to compute the advection term [3]. They extrapolate $\boldsymbol{u}$ at the cell edge from the adjacent cell-center values using Taylor series expansion in order to find the upwind state. Similar or adaptive algorithms for this type of methods can be found in [1, 2, 4, 44, 5, 72, 87]. We outline this projection method in the semi-discrete form as follows:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^{n+1/2} \cdot \nabla)\boldsymbol{u}^{n+1/2} + \nabla p^{n-1/2} = \frac{1}{2Re}(\nabla^2 \boldsymbol{u}^* + \nabla^2 \boldsymbol{u}^n) + \boldsymbol{F} \tag{2.16}$$

$$\boldsymbol{u}^* = \boldsymbol{b} \qquad \text{on} \qquad \partial\Omega \tag{2.17}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\triangle t} = -\frac{\nabla p^{n+1/2} - \nabla p^{n-1/2}}{\gamma} = -\nabla \phi^{n+1} \tag{2.18}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0 \tag{2.19}$$

$$\boldsymbol{n} \cdot \boldsymbol{u}^{n+1} = \boldsymbol{n} \cdot \boldsymbol{b} \qquad \text{on} \qquad \partial\Omega. \tag{2.20}$$

From the first equality of (2.18), we have

$$\boldsymbol{u}^{n+1} \approx \boldsymbol{u}^* - \frac{(\triangle t^2)}{\gamma}\frac{\partial \nabla p}{\partial t}. \tag{2.21}$$

This implies $\boldsymbol{u}^* = \boldsymbol{b} + O(\triangle t^2)$ on $\partial\Omega$. Hence it is reasonable to assume (2.17). In (2.21), a larger γ gives a better approximation, but Shen [89] uses functional space methods to show that the choice of γ should not affect the precision of the scheme theoretically. Gresho and Chan [45], however, indicate that $0 \leqslant \gamma \leqslant 2$ is required for numerical stability reasons.

In [8], the authors argue that the Poisson problem (2.8) is solved with the boundary condition (2.10). This implies

$$\boldsymbol{n} \cdot \nabla p^{n+1/2} = \boldsymbol{n} \cdot \nabla p^{n-1/2} \qquad \text{on} \qquad \partial\Omega \tag{2.22}$$

for all n. This means that the pressure updated by (2.18) cannot be correct on the boundary in general. They also point out that this "boundary layer" type of error is well known and has been analyzed in [32, 90, 96]. A treatment suggested in [8] to improve the order of accuracy for pressure is to use the pressure-updating formula

$$\frac{\nabla p^{n+1/2} - \nabla p^{n-1/2}}{\gamma} = \nabla \phi^{n+1} - \frac{\triangle t}{2Re} \nabla \nabla^2 \phi^{n+1} \tag{2.23}$$

instead.

2.2 Gauge methods

Oseledets [75] considers a Hamiltonian formulation of the incompressible Navier-Stokes equations based on a new variable $\boldsymbol{M}$:

$$\begin{aligned} &\boldsymbol{M}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{M} + \boldsymbol{M} \cdot (\nabla \boldsymbol{u})^T = \frac{1}{Re} \nabla^2 \boldsymbol{M} + \boldsymbol{F}, \\ &\boldsymbol{u} = \mathbb{P}\boldsymbol{M}, \end{aligned} \tag{2.24}$$

where $\boldsymbol{u}$ is the velocity and $\mathbb{P}$ is the projection operator which projects any vector to its divergence-free subspace. $\boldsymbol{F}$ is an external force. For a physical interpretation of (2.24), one can see [11, 16, 29]. It is worth pointing out the relation between (2.24) and standard incompressible Navier-Stokes equations (2.1). If $\boldsymbol{M}$ satisfies (2.24) with $\boldsymbol{M} = \boldsymbol{u} + \nabla \phi$ and we set

$$p = \phi_t + (\boldsymbol{u} \cdot \nabla)\phi - \frac{1}{Re} \nabla^2 \phi + \frac{1}{2} |\boldsymbol{u}|^2, \tag{2.25}$$

then $(\boldsymbol{u}, p)$ satisfies (2.1). Conversely, if $(\boldsymbol{u}, p)$ satisfies (2.1) and ϕ satisfies (2.25), then $(\boldsymbol{M}, \phi)$ is a solution to (2.24) with $\boldsymbol{M} = \boldsymbol{u} + \nabla \phi$.

The numerical method based on the formulation (2.24) has been developed by [11, 23, 29, 30] and we will refer to it as the gauge method. By introducing a new variable $\boldsymbol{m} = \boldsymbol{u} + \nabla \phi$, the gauge method can be written as follows:

$$\boldsymbol{m}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \frac{1}{Re} \nabla^2 \boldsymbol{m} + \boldsymbol{F} \tag{2.26}$$

$$\boldsymbol{u} = \boldsymbol{b} \qquad \text{on} \qquad \partial\Omega \tag{2.27}$$

$$\boldsymbol{u} = \boldsymbol{m} - \nabla \phi. \tag{2.28}$$

Although the pressure is not needed to advance the velocity, we can enforce the equivalence of (2.1) and (2.26) to recover the pressure

$$p = \phi_t - \frac{1}{Re} \nabla^2 \phi. \tag{2.29}$$

The semi-discrete form of the gauge method is:

$$\frac{\boldsymbol{m}^{n+1}-\boldsymbol{m}^n}{\triangle t}+(\boldsymbol{u}^{n+1/2}\cdot\nabla)\boldsymbol{u}^{n+1/2}=\frac{1}{2Re}(\nabla^2\boldsymbol{m}^{n+1}+\nabla^2\boldsymbol{m}^n)+\boldsymbol{F} \tag{2.30}$$

$$\boldsymbol{u}^{n+1}=\boldsymbol{b} \qquad \text{on} \qquad \partial\Omega \tag{2.31}$$

$$\boldsymbol{u}^{n+1}=\boldsymbol{m}^{n+1}-\nabla\phi^{n+1}, \tag{2.32}$$

with the pressure solved by

$$p^{n+1}=\frac{\phi^{n+1}-\phi^n}{\triangle t}-\frac{1}{2Re}\nabla^2(\phi^{n+1}+\phi^n). \tag{2.33}$$

We note that solving (2.30) requires not only (2.31) but also the boundary condition for $\boldsymbol{m}^{n+1}$. There is a "gauge" freedom in choosing this boundary condition. In [30], the boundary conditions for $\boldsymbol{m}^{n+1}$ are chosen to be

$$\begin{aligned} \boldsymbol{n}\cdot\boldsymbol{m}&=\boldsymbol{n}\cdot\boldsymbol{b} && \text{on} \quad \partial\Omega \\ \boldsymbol{\tau}\cdot\boldsymbol{m}^{n+1}&=\boldsymbol{\tau}\cdot\boldsymbol{b}+\boldsymbol{\tau}\cdot\nabla(2\phi^n-\phi^{n-1}) && \text{on} \quad \partial\Omega, \end{aligned} \tag{2.34}$$

which are consistent with the compatibility condition

$$\boldsymbol{m}^{n+1}-\nabla\phi^{n+1}=\boldsymbol{b} \qquad \text{on} \qquad \partial\Omega. \tag{2.35}$$

The velocity at the end of each time step is updated by (2.32), where ϕ^{n+1} is the solution to the Neumann problem

$$\begin{aligned} \nabla^2\phi^{n+1}&=\nabla\cdot\boldsymbol{m}^{n+1} \\ \frac{\partial\phi^{n+1}}{\partial\boldsymbol{n}}&=0 \qquad \text{on} \qquad \partial\Omega. \end{aligned} \tag{2.36}$$

The new projection method introduced in the next chapter can also be implemented using the formulation of the gauge method, although we do not pursue this implementation in this thesis.

For detailed analysis of gauge methods one can see, for instance, [8].

2.3 Existing numerical methods for interface problems

Numerical methods such as finite element methods, volume of fluid methods, and boundary integral methods are heavily used for studying free surface or moving surface problems and the fluid-structure interaction [50, 51, 52, 73, 87, 103]. Recently, overset grid methods are introduced to study the interface dynamics of free surface flows [35, 37], We begin our discussion with the immersed boundary method which is the motivation for the development of the immersed interface method for incompressible Navier-Stokes equations in Chapter 5. We also briefly introduce the immersed interface method for the Stokes equation with a singular force or surface tension. Finally we show two numerical methods that are closely related to the immersed boundary method or the immersed interface method.

2.3.1 The immersed boundary method

One of the most widespread methods to study incompressible fluid with moving interfaces is the immersed boundary method. In a sequence of papers [77, 78, 80], Peskin and McQueen introduced the immersed boundary method for solving the viscous incompressible Navier-Stokes equations with moving elastic boundaries. This method is developed further in [81, 82, 83, 100]. While originally developed to model blood flow in the heart and through heart valves, it has since been used for many applications, particularly in biofluid dynamics where interesting fluid flow problems with immersed elastic membranes or structures abound. Examples include modeling the swimming of organisms [38], platelet aggregation in blood clotting [40], cochlea dynamics [7], biofilm processes [27] and wood pulp fiber dynamics [91]. A recent paper of Peskin has a thorough review for the immersed boundary method [79].

The key idea in the immersed boundary method is to replace the material interface with discrete control points $\boldsymbol{X}_k^n$ that exert singular forces to the fluid. The singular force exerted of each control point $\boldsymbol{F}_k^n$ is spread to the nearby Cartesian grid points by a discrete delta function so that it yields nonzero force term $\boldsymbol{F}_{i,j}$ at grid points near the interface. After the spreading process, the internal boundaries are eliminated and the fluid equation can be solved by a simple finite difference method. Then the resulting velocity field is interpolated back to the control points to obtain $\boldsymbol{U}_k^{n+1}$. Finally we move the control points by solving

$$\boldsymbol{X}_k^{n+1} = \boldsymbol{X}_k^n + \Delta t \boldsymbol{U}_k^{n+1}. \tag{2.37}$$

A typical example of a one-dimension discrete delta function is defined as follow:

$$\delta_h(x) = \begin{cases} \frac{1}{4h}\left(1 + \cos\frac{\pi x}{2h}\right) & |x| \leq 2h, \\ 0 & |x| \geq 2h, \end{cases} \tag{2.38}$$

where h is the grid size. The numerical delta function is plotted in Figure 2.1.

Using (2.38), we define a two-dimensional discrete delta function as

$$D_{i,j}(\boldsymbol{x}) = \delta_h(x - ih)\delta_h(y - jh), \tag{2.39}$$

where (ih, jh) are the coordinates of the mesh points.

Figure 2.2 is an irregular interface embedded in a rectangle with a uniform Cartesian grid. Applying the two-dimensional discrete delta function (2.39) to the interface, we spread the singular force at the control point, "•", to the nearby cell-centered grid points, "**x**". The nonzero force $\boldsymbol{F}_{i,j}$ at the $(i,j)th$ grid point is a sum of the spread forces from all control points (see (1.14)). The function (2.39) also serves the purpose of interpolating the fluid velocity to the boundary points.

The advantages of the immersed boundary method using the smooth discrete delta function are:

- It is simple and easy to implement.
- It is flexible for complex geometries.

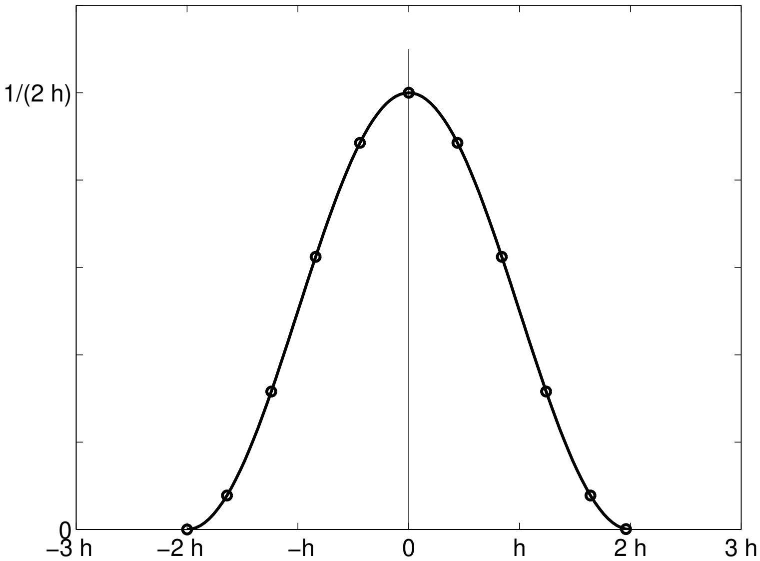


Figure 2.1: One-dimension discrete delta function.

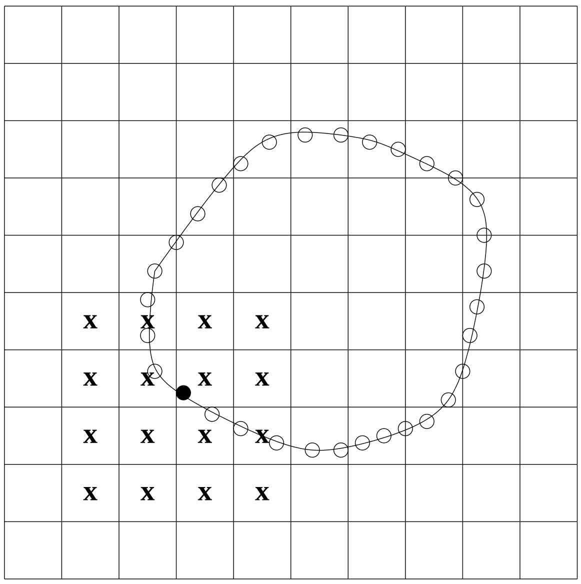


Figure 2.2: The singular force of a control point is spread to nearby cell-centered grid points.

However, the trade-off for the simplicity and flexibility is

- The method is at most first-order accurate, because the smooth delta function smears out all the kinks or discontinuities in the solution. Most interface problems have either kinks or discontinuities in their solutions. For example, the fluid-material interface in incompressible flow has a discontinuous pressure across the interface. The velocity is continuous but its derivative is not.

- The method has poor volume conservation, if the interpolated velocity, in which the control points move is not continuously divergence free [79, 83]. This is a serious problem, if one needs the volume conservation to depict the accurate dynamics. Peskin and Printz [83] have shown that this problem can be remedied by introducing a modified projection operator on a doubly periodic domain, but this modified projection operator is rather complicated.

2.3.2 The immersed interface method

The idea of the immersed interface method for difference schemes first appeared in the paper of Tikhonov and Samarskii [97]. The authors used jump conditions to derive a second order accurate difference method on uniform grids for one-dimensional two-point boundary value problems with discontinuous coefficients.

In two dimensions, Mayo [70] used the same idea to derive a second-order-accurate interpolation for the Laplacian in the context of solving Poisson or biharmonic equations on irregular regions by an integral equation. The region is embedded in a uniform Cartesian grid where a fast solver can be used provided that the right hand side of the resulting linear system is appropriately modified near the original boundary.

LeVeque and Li [63] considered an interface embedded in a uniform Cartesian grid and derived a finite difference method for solving elliptic equations with discontinuous coefficients and singular sources. The key idea of this method is to incorporate jump conditions into Taylor series expansions to derive difference equations for the Laplacian near the interface that achieve pointwise second-order accuracy. Jump conditions in the solution and/or its derivatives are used to find the correction terms and the corresponding coefficients for stencils near the interface (see figure 2.3). If the elliptic equation has discontinuous coefficients, then near the interface a six-point stencil is needed to discretize the Laplacian. This immersed interface method was motivated by the immersed boundary method and also successfully implemented for solving Stokes flow with moving elastic boundaries or surface tension [64].

There are other immersed interface methods for solving partial differential equations [49, 67, 104, 105, 106, 107]. Cartesian grid finite volume methods for the advection-diffusion equation and for stream function vorticity formulation of incompressible flow in irregular geometries can be found in [12, 13, 14].

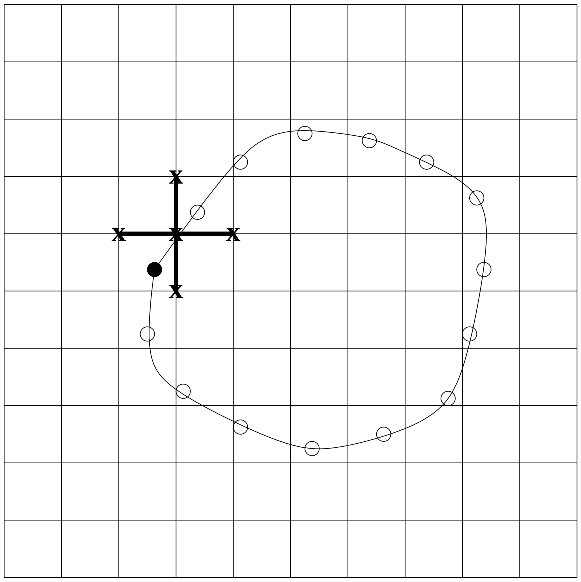


Figure 2.3: A stencil for the interpolation formula in [63].

2.3.3 The blob projection method

In [24] the authors propose to replace the discrete delta function with compact support (2.38) by a cutoff function for solving the incompressible Navier-Stokes equations with singular forces. An example of the cutoff is $\phi_\delta(x) = \delta^{-2}\phi(|x|/\delta)$ where ϕ is a radially symmetric function satisfying appropriate moment conditions, for instance a Gaussian distribution. The cutoff radius δ is a numerical parameter that is not necessarily a multiple of the grid size h, although it is usually chosen to be so. It can be chosen independently. The singular force is then regularized by the cutoff the same way as by the discrete delta function. The nice property of the cutoff is that the projection of the regularized force field evaluated at an arbitrary point x can be found explicitly. We summarize the blob projection method as follows:

1. compute the new force f_k at the membrane points shown in Figure 2.2;

2. compute the projection of the advection-diffusion equation to get u_1;

3. evaluate the projection of the regularized force field to get u_2;

4. update the new grid velocity with $u^{n+1} = u^n + \Delta t(u_1 + u_2)$;

5. update the velocity at the membrane points and advance the membrane by (2.37).

The method is reported to have high-order accuracy and a good volume conservation. However, theoretically the blob projection method suffers the similar difficulty that immersed boundary methods have. We address this issue in Section 4.8.

2.3.4 The ghost fluid method

The ghost fluid method was originally introduced to solve compressible fluid problems with contact discontinuities [39]. The idea of the ghost fluid method is very simple. Consider the case of two different fluids, water and air, in one dimension. The contact discontinuity of the fluids is represented by the zero contour of a level set function. Assume that the pressure and velocity are continuous across the interface. In order to solve for the air, the method replaces the water with "*ghost air*" that acts like the water (in velocity and pressure) but appears to be air (in entropy). In other words, the method sets the pressure and the velocity of the ghost air identically equal to the pressure and the velocity of water at each point, but sets the extrapolated entropy of air. Vice versa the method solves for the water. After solving for each fluid, based on the level set function, the method keeps the values of the real fluids and tosses the ghost values. As in the immersed interface method, the ghost fluid method uses the jump conditions to define the "*ghost cells*" values and extends the method to elliptic equations with discontinuous coefficients and multiphase incompressible flow [53, 69]. The ghost fluid type methods are only first-order accurate.

2.4 Methods for the surface representation

The surface representation is an important issue for numerical methods that solve interface problems. A method used to represent the interface should not only capture the evolution of the interface accurately, but also must be efficient to implement. Above all, the method cannot overlook any external physics associated with the evolution of the interface. One popular method for tracking the evolution of the interface is the front tracking method. This method is frequently used for incompressible flow with interfaces [48, 101]. Both the immersed boundary method and blob projection method explicitly track a set of control points for the evolution of the interface, and thus are one form of front tracking methods. In fact, for elastic membranes the elastic force depends on the manner in which the surface is stretched and not just its location, and hence it may be necessary to track the control points explicitly. In this thesis, we follow the ideas in [64, 84] to represent the membrane by cubic splines through a smaller set of control points than those used in the immersed boundary method or the blob projection method. The advantages of using the spline function are

1. The force density, and hence the jump conditions, at any position of the membrane can be accurately evaluated by the spline function.

2. A smaller Hessian matrix is formed for the quasi-Newton's method in our implicit immersed interface method, which reduces the computational cost.

When the singular force depends only on the curvature of the interface, the level set method could be used to track the interface. In this method, the interface is represented by the zero contour of the level set function, $\phi(\boldsymbol{x}, t) = 0$. The curvature of the interface can be determined directly from ϕ. A recent review of the progress of level set methods is in [76].

Hybrid front-tracking/level-set methods have been developed recently in [33, 98].

Chapter 3

A PROJECTION METHOD FOR INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

We introduce a second-order Godunov-type projection method for incompressible flows in this chapter. This new version of projection method uses high-resolution finite volume methods from CLAWPACK [59] for the advective terms and finite difference methods for the diffusion and the Poisson problem. The high-resolution algorithm advects the cell-centered velocities using the divergence-free cell edge velocities. It is a robust method for flows at high Reynolds numbers. With a simple change in the projection operator, this second-order method can be applied to variable density flows. It is suitable for problems containing discontinuous conserved quantities, such as Rayleigh-Taylor instability and the Boussinesq equations. Numerical studies of the convergence and order of accuracy and simulations for several test problems are presented.

3.1 *Conservative algorithms for advection*

The projection method proposed in the next section is a hybrid of finite-volume and finite-difference methods. This method is motivated by the high-resolution wave propagation algorithms developed by LeVeque [59, 60, 61]. In this section we discuss the wave propagation algorithm for advection in incompressible flows.

We consider the scalar advection equation in a specified velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ in one, two, or three space dimensions

$$q_t + \nabla \cdot (\boldsymbol{u}q) = 0 \tag{3.1}$$

where $q(\boldsymbol{x}, t)$ is a conservative quantity which can be the scalar concentration or the density function. If the flow is incompressible, $\nabla \cdot \boldsymbol{u} = 0$, the conservative form (3.1) can be written in a advective form

$$q_t + \boldsymbol{u} \cdot \nabla q = 0. \tag{3.2}$$

Although (3.1) and (3.2) are mathematically equivalent for incompressibe flow, numerical algorithms based on the two may behave differently. In two dimensions, a staggered grid as shown in Figure 3.1 is used in [61] so that the numerical algorithms modeled either from the conservative form or from the advective form are the same. The notations in Figure 3.1 are as follows. Let C_{ij} be the (i, j) grid cell $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$. The "edge velocities" $u_{i\pm1/2,j}$ and $v_{i,j\pm1/2}$ are the velocities at the midpoints of the interfaces $(x_{i\pm1/2}, y_j)$ and $(x_i, y_{j\pm1/2})$, giving rise to the waves being propagated. They should satisfy the discrete divergence-free relationship

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\triangle x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\triangle y} = 0. \tag{3.3}$$

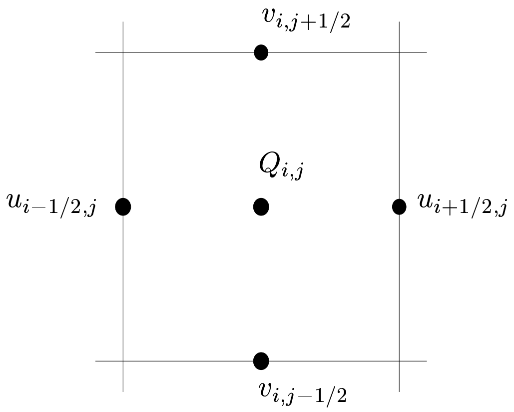


Figure 3.1: Cell-variable locations for the conservative algorithm.

$Q_{i,j}$ represents an approximation to the cell average,

$$Q_{i,j} \approx \frac{1}{\triangle x \triangle y} \int_{C_{i,j}} q(x, y, t)\, dx\, dy. \tag{3.4}$$

Using the relationship (3.3) and the upwind wave-propagation method, LeVeque [61] developed the high-resolution conservative algorithms for (3.2) in incompressible flow. In these algorithms, a flux-limiter is applied to the second-order Lax-Wendroff method to deal with steep gradients or even discontinuities in $q(x, y, t)$ without introducing spurious numerical oscillations or excessive numerical diffusion. This high-resolution property gives our fractional step method the edge in handling flows where the conserved quantities have a sharp discontinuity between fluids, such as the density in Rayleigh-Taylor instability . In two dimensional space, the high-resolution algorithms can use either dimensional split or unsplit methods. The unsplit method improves the first-order corner transport upwind (CTU) [21] by adding correction waves and transverse propagation of the correction waves.

3.1.1 The advection-diffusion equation

If we add a diffusive term to (3.2), we have the advection-diffusion equation. The equations (2.2) and (2.16) are the semi-discrete forms of the advection-diffusion equations. For the high-resolution algorithms, one way to tackle the advecton-diffusion equation is to use fractional step methods [14]. We consider the semi-discrete advection-diffusion equation

$$\frac{Q^{n+1} - Q^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla) Q^{n+1} = \frac{\mu}{2}(\nabla^2 Q^{n+1} + \nabla^2 Q^n) \tag{3.5}$$

where the spatial discretization of $\boldsymbol{u}^n$ satisfies (3.3) and μ is the viscosity. The first-order fractional step method, sometimes called the *Godunov splitting*, can be written as follows

$$\frac{Q^* - Q^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla) Q^* = 0 \tag{3.6}$$

$$\frac{Q^{n+1}-Q^*}{\triangle t} = \frac{\mu}{2}(\nabla^2 Q^{n+1} + \nabla^2 Q^*). \tag{3.7}$$

We first compute Q^* in (3.6) using the high-resolution method described in the previous section. Equation (3.7) is a Crank-Nicolson discretization for the diffusion. It results in a Helmholtz type equation. This splitting method is only first-order accurate. Using the same notation, we can write down a formally second-order method, *the Strang splitting*, as follows

$$\frac{Q^*-Q^n}{\triangle t/2} = \frac{\mu}{2}(\nabla^2 Q^* + \nabla^2 Q^n). \tag{3.8}$$

$$\frac{Q^{**}-Q^*}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla)Q^{**} = 0 \tag{3.9}$$

$$\frac{Q^{n+1}-Q^{**}}{\triangle t/2} = \frac{\mu}{2}(\nabla^2 Q^{n+1} + \nabla^2 Q^{**}). \tag{3.10}$$

In practice, (3.8) from one step and (3.10) from the following step can be combined into a single step of length $\triangle t$ to reduce numerical diffusion and computational cost. By doing so, Strang splitting uses $\triangle t/2$ only in the first and last time step. In between, Strang splitting is identical to Godunov splitting. In practice (see [14]) one can obtain better than first-order accuracy using the Godunov splitting.

3.2 A projection method: conservative algorithm approach

We begin this section with the definitions of our discrete divergence and discrete gradient operators. In order to use the high-resolution algorithms from Section 3.1, the discrete values of the cell interface velocities have to satisfy(3.3). Let

$$\boldsymbol{u}_{i\pm1/2,j\pm1/2} = (u_{i\pm1/2,j}, v_{i,j\pm1/2})$$

denote the horizontal and vertical components of the discrete velocity field at the edges of the (i,j) cell $C_{i,j}$ and let

$$\boldsymbol{U}_{i,j} = (U_{i,j}, V_{i,j})$$

denote the approximation to the cell average

$$\boldsymbol{U}_{i,j} \approx \frac{1}{\triangle x \triangle y}\int_{C_{i,j}} \boldsymbol{U}(x,y,t)\,dx\,dy.$$

Assuming that ϕ is a scalar function, we consider a staggered grid as shown in Figure 3.2 and define the discrete gradient and divergence operators as follows

$$(\mathcal{D}\boldsymbol{u})_{i,j} = \frac{u_{i+1/2,j}-u_{i-1/2,j}}{\triangle x} + \frac{v_{i,j+1/2}-v_{i,j-1/2}}{\triangle y} \tag{3.11}$$

$$(\mathcal{G}\phi)_{i,j} = (\frac{\phi_{i,j}-\phi_{i-1,j}}{\triangle x}, \frac{\phi_{i,j}-\phi_{i,j-1}}{\triangle y}) \tag{3.12}$$

$$(\mathcal{D}^\circ \boldsymbol{U})_{i,j} = \frac{U_{i+1,j}-U_{i-1,j}}{2\triangle x} + \frac{V_{i,j+1}-V_{i,j-1}}{2\triangle y} \tag{3.13}$$

$$(\mathcal{G}^\circ\phi)_{i,j} = (\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2\triangle x}, \frac{\phi_{i,j+1}-\phi_{i,j-1}}{2\triangle y}). \tag{3.14}$$

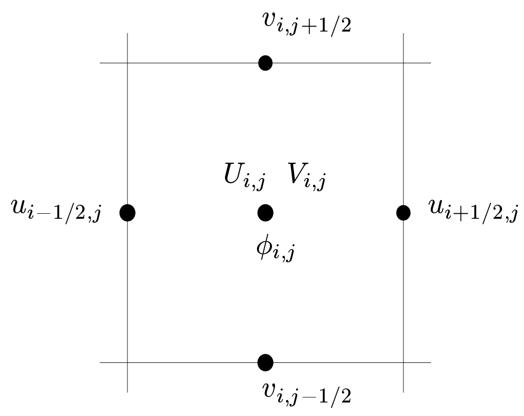


Figure 3.2: Cell variables of the projection method .

Using these operators, we introduce the algorithm of the new projection method in two-dimensional space. Let $\boldsymbol{U}$ be the divergence-free velocity field with respect to the continuous divergence operator. $\boldsymbol{U}_{i,j}$ is the discrete value of $\boldsymbol{U}$ at the cell center of $C_{i,j}$. We find the discrete edge velocity by taking the average the cell-centered values. For example, the left edge value of cell $C_{i,j}$ is

$$u_{i-1/2,j} = \frac{1}{2}(U_{i-1,j} + U_{i,j}) \tag{3.15}$$

and the bottom edge value is

$$v_{i,j-1/2} = \frac{1}{2}(V_{,j-1} + V_{i,j}). \tag{3.16}$$

If the edge values (3.15) and (3.16) satisfy (3.3) initially, the edge velocities are discrete divergence free with respect to the operator (3.11). Otherwise, we need to apply one projection step so that (3.3) is satisfied. We will explain the projection step in Step P2.

The algorithm then takes the following form.

Step P1. Using the edge values $u^n_{i-1/2,j}$ and $v^n_{i,j-1/2}$, we advect and diffuse the cell-centered velocities over a time step $\triangle t$ using the algorithm in Section 3.1, with the conservative quantity $Q_{i,j}$ to be the centered value $U^n_{i,j}$ or $V^n_{i,j}$. This results in the intermediate velocity field $\boldsymbol{U}^*_{i,j} = (U^*_{i,j}, V^*_{ij})$.

Step P2. We average the discrete centered values to get the edge values

$$u^*_{i-1/2,j} = \frac{1}{2}(U^*_{i-1,j} + U^*_{i,j}) \tag{3.17}$$

$$v^*_{i,j-1/2} = \frac{1}{2}(V^*_{i,j-1} + V^*_{i,j}). \tag{3.18}$$

To compute divergence-free edge values that satisfy (3.3) from $u^*_{i-1/2,j}$ and $v^*_{i,j-1/2}$, we need to find the scalar field ϕ by solving

$$\mathcal{DG}(\triangle t\, \phi)^{n+1}_{i,j} = \mathcal{D}\boldsymbol{u}^*_{i,j} \tag{3.19}$$

where the $\mathcal{D}$, $\mathcal{G}$ operators and $\mathcal{D}\boldsymbol{u}^*_{i,j}$ are defined as (3.11) and (3.12). The $\mathcal{DG}$ operator is a compact five-point Laplacian operator and (3.19) is the discrete version of the Poisson equation (2.8).

Step P3. Let

$$\mathcal{G}(\triangle t\,\phi)^{n+1}_{i,j} = (\mathcal{G}_{1,(i,j)},\,\mathcal{G}_{2,(i,j)}). \tag{3.20}$$

We update the interface values by

$$u^{n+1}_{i-1/2,j} = u^*_{i-1/2,j} - \mathcal{G}_{1,(i,j)} \tag{3.21}$$

$$v^{n+1}_{i,j-1/2} = v^*_{i,j-1/2} - \mathcal{G}_{2,(i,j)} \tag{3.22}$$

so that $\mathcal{D}\boldsymbol{u}^{n+1}_{i,j} = 0$ and (3.3) is satisfied.

Step P4. Let

$$\mathcal{G}^0(\triangle t\,\phi)^{n+1}_{i,j} = (\mathcal{G}^0_{1,(i,j)},\,\mathcal{G}^0_{2,(i,j)}). \tag{3.23}$$

We update the cell-centered values by

$$U^{n+1}_{i,j} = U^*_{i,j} - \mathcal{G}^0_{1,(i,j)} \tag{3.24}$$

$$V^{n+1}_{i,j} = V^*_{i,j} - \mathcal{G}^0_{2,(i,j)}. \tag{3.25}$$

The exactly divergence-free edge values are obtained by subtracting $\mathcal{G}_{1,(i,j)}$ and $\mathcal{G}_{2,(i,j)}$ from the intermediate edge velocities. It is easy to show

$$\mathcal{G}^0_{1,(i,j)} = \frac{1}{2}(\mathcal{G}_{1,(i,j)} + \mathcal{G}_{1,(i+1,j)}) \tag{3.26}$$

$$\mathcal{G}^0_{2,(i,j)} = \frac{1}{2}(\mathcal{G}_{2,(i,j)} + \mathcal{G}_{2,(i,j+1)}) \tag{3.27}$$

by the definition of equations (3.12) and (3.14). So it is natural to update cell-centered values $\boldsymbol{U}^{n+1}$ using (3.24) and (3.25). One can expect $\boldsymbol{U}^{n+1}$ to be divergence free up to second order, since the edge values are exactly divergence free. This completes one time step. Go to Step P1 for the next time step.

3.2.1 Remarks

1. From (3.11), (3.13), (3.17), and (3.18), it is easy to show

$$\mathcal{D}\boldsymbol{u}^*_{i,j} = \mathcal{D}^0\boldsymbol{U}^*_{i,j}. \tag{3.28}$$

Combining (3.19) and (3.28) gives

$$\mathcal{DG}(\triangle t\,\phi)^{n+1}_{i,j} = \mathcal{D}^0\boldsymbol{U}^*_{i,j}. \tag{3.29}$$

If we apply the operator $\mathcal{D}^0$ on (3.24) and (3.25) and compare with (3.29), we see that the cell-centered $\boldsymbol{U}^{n+1}$ is not discrete divergence free with respect to the discrete divergence operator $\mathcal{D}^0$, but is, however, divergence free up to second order. Minion [72] used the similar approach in his BCG version of adaptive projection methods to avoid the decoupled stencil arising from the projection step.

2. Although in Step P1 we do not need the pressure p to advance the velocity, we can obtain the pressure p^{n+1} from (2.7) within each time step. The projection method stated in the previous section is parallel to the finite difference scheme described in Section 2.1.1.

3. If we update the pressure gradient $\nabla p^{n+1/2}$ using (2.18) and add $\nabla p^{n-1/2}$ in the advection-diffusion equation in Step P1, we modify our projection method to a method which is parallel to the pressure-correction scheme in Section 2.1.2. The term $\nabla p^{n-1/2}$ in the advection-diffusion equation is treated as an external force and is solved together with the diffusive term using the fractional step method in Section 3.1.1.

4. Since the high-resolution method used in Step P1 to advect the intermediate velocity field is explicit, a CFL condition must be imposed to ensure stability. The method requires
$$\max_{i,j}\left(\frac{\triangle t}{\triangle x}|u_{i-1/2,j}|, \frac{\triangle t}{\triangle x}|v_{i,j-1/2}|\right) \leqslant 1. \tag{3.30}$$

3.3 *The projection method for variable-density flows*

We consider the unsteady incompressible Navier-Stokes equations for flows with finite-amplitude density variation. Conservation of mass is described by an advection equation. We write the system of equations as follows
$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} &= \frac{1}{\rho}(\nabla\cdot\mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) - \nabla p + \boldsymbol{F}) \\ \rho_t + (\boldsymbol{u}\cdot\nabla)\rho &= 0 \\ \nabla\cdot\boldsymbol{u} &= 0 \end{aligned} \tag{3.31}$$
where ρ is the density and μ is the viscosity which is allowed to depend on density. Function $\boldsymbol{F}$ is an external force while $\boldsymbol{u}$ and p are the velocity and hydrodynamic pressure respectively. In two-dimensional space, let $\boldsymbol{u} = (u, v)$ and define $\boldsymbol{q} = (u, v, \rho) = (\boldsymbol{u}, \rho)$. For simplicity, we assume μ is a constant. Again for simplicity, we consider the Dirichlet boundary conditions. The projection method based on the high-resolution conservative algorithm for the above system can be stated as follows.

Step V1. Use the centered values $\boldsymbol{Q}^n = (U^n, V^n, \rho^n)$ and edge values $\boldsymbol{u}^n = (u^n, v^n)$ to solve the advection equation
$$\frac{\boldsymbol{Q} - \boldsymbol{Q}^n}{\triangle t} + \boldsymbol{u}^n\cdot\nabla\boldsymbol{Q} = 0. \tag{3.32}$$
This is solved on a finite-volume grid using the explicit second-order high-resolution algorithm from *CLAWPACK*. The resulting solution is $\boldsymbol{Q}^\dagger = (\boldsymbol{U}^\dagger, \rho^\dagger) = (U^\dagger, V^\dagger, \rho^\dagger)$.

Step V2. Use a Crank-Nicolson discretization for the diffusion
$$\frac{\boldsymbol{U}^* - \boldsymbol{U}^\dagger}{\triangle t} = \frac{\mu}{2\rho^\dagger}(\nabla^2\boldsymbol{U}^* + \nabla^2\boldsymbol{U}^\dagger + \boldsymbol{F}). \tag{3.33}$$
This gives the intermediate velocity $\boldsymbol{U}^*$ at cell center.

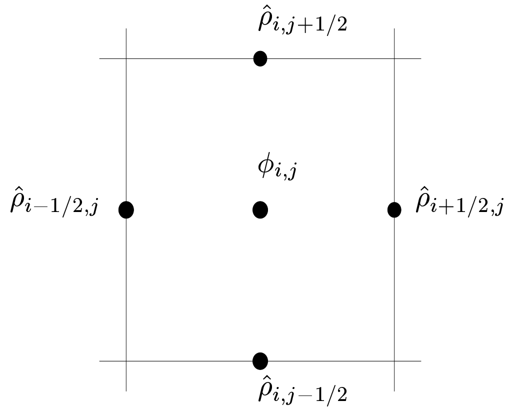


Figure 3.3: Staggered grid for the Poisson problem with variant coefficients

Step V3. Obtain the edge velocity $\boldsymbol{u}^{n+1}$ by

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{1}{\hat\rho}\mathcal{G}(\triangle t\phi^{n+1}), \tag{3.34}$$

where $\boldsymbol{u}^*$ is the average of the adjacent $\boldsymbol{U}^*$ and $\hat\rho$ is the average of the adjacent $\rho^\dagger$.

The update (3.34) requires the value ϕ^{n+1}, which can be obtained by solving a discrete Poisson problem:

Taking the discrete divergence of (3.34) and using the fact that we want $\mathcal{D}u^{n+1} = 0$, we obtain the variable-coefficient Poisson problem

$$\mathcal{D}\left(\frac{1}{\hat\rho}\mathcal{G}\phi^{n+1}\right) = \frac{1}{\triangle t}\mathcal{D}u^* \tag{3.35}$$

Consider the staggered grid shown in Figure 3.3. If we denote

$$\beta_{i-1/2,j} = \frac{1}{\hat\rho_{i-1/2,j}}, \tag{3.36}$$

then a finite difference discretization for (3.35) is

$$\begin{aligned}&\frac{\phi_{i+1,j}\,\beta_{i+1/2,j} - \phi_{i,j}(\beta_{i+1/2,j} + \beta_{i-1/2,j}) + \phi_{i-1,j}\,\beta_{i-1/2,j}}{(\triangle x)^2} + \\ &\frac{\phi_{i,j+1}\,\beta_{i,j+1/2} - \phi_{i,j}(\beta_{i,j+1/2} + \beta_{i,j-1/2}) + \phi_{i,j-1}\,\beta_{i,j-1/2}}{(\triangle y)^2} = \\ &\frac{1}{\triangle t}\left(\frac{u^*_{i+1,j} - 2u^*_{i,j} + u^*_{i-1,j}}{(\triangle x)^2} + \frac{v^*_{i,j+1} - 2v^*_{i,j} + v^*_{i,j-1}}{(\triangle y)^2}\right).\end{aligned} \tag{3.37}$$

The resulting linear system from (3.37) is

$$\boldsymbol{A\phi} = \boldsymbol{f}, \tag{3.38}$$

where the coefficient matrix $\boldsymbol{A}$ is symmetric and positive definite [43].

Step V4. In the final step, we update the cell-centered values by

$$U^{n+1} = U^* - \frac{\triangle t}{2}\left(\frac{1}{\hat{\rho}_{i-1/2,j}}(\phi_{i,j} - \phi_{i-1,j}) + \frac{1}{\hat{\rho}_{i+1/2,j}}(\phi_{i+1,j} - \phi_{i,j})\right) \tag{3.39}$$

$$V^{n+1} = V^* - \frac{\triangle t}{2}\left(\frac{1}{\hat{\rho}_{i,j-1/2}}(\phi_{i,j} - \phi_{i,j-1}) + \frac{1}{\hat{\rho}_{i,j-1/2}}(\phi_{i,j+1} - \phi_{i,j})\right), \tag{3.40}$$

and set $\rho^{n+1} = \rho^{\dagger}$. This completes one time step. Go to Step V1 for next time step.

We notice that if the density $\rho \equiv 1$ is constant, then the algorithm is identical to the projection method in Section 3.2.

3.4 The projection method for the Boussinesq flow

We consider the convection of a Boussinesq fluid in a two-dimension rectangular cavity on $x - z$ plane. The flow is governed by the dimensionless Boussinesq equations [42].

$$\begin{aligned}
Pr^{-1}\,[\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla\boldsymbol{u})\boldsymbol{u}\,] &= -\nabla p + \nabla^2\boldsymbol{u} + Ra\,T\boldsymbol{e}_z \\
T_t + (\boldsymbol{u}\cdot\nabla)T &= \nabla^2 T \\
\nabla\cdot\boldsymbol{u} &= 0 \\
\boldsymbol{u} = 0, \qquad \text{on} \quad z = Z_a &\quad \text{and} \quad z = Z_b \\
T = T_a \quad \text{on} \quad z = Z_a, \qquad T = T_b &\quad \text{on} \quad z = Z_b,
\end{aligned} \tag{3.41}$$

where T is the temperature, Pr the Prantal number, and Ra the Rayleigh number. Here $\boldsymbol{e}_z$ is the unit vector in the vertical direction. The dimensionless parameter Ra is given by

$$Ra = \frac{\alpha\, g\,(T_b - T_a)\,d^3}{\nu\,\kappa}, \tag{3.42}$$

where α is the thermal expansion coefficient, T_a and T_b the temperatures of the top and bottom plates, $d = Z_a - Z_b$ the height of the cavity, g the gravity acceleration, ν the kinematic viscosity, and κ the thermal diffusion. The other parameter Pr is given by

$$Pr = \frac{\nu}{\chi}, \tag{3.43}$$

where χ is the thermal conductivity.

The Boussinesq equation is a good approximation for studying Rayleigh-Bénard convection, in which a viscous fluid in a cavity is heated from the bottom plate and the top plate is maintained at a lower temperature. When the temperature between the top and bottom plates is a linear function of the height of the cavity and the initial velocity is zero everywhere, the linear stability theorem shows that a static solution exists to the problem (3.41) [15]. Increasing the temperature of the hot plate until the Rayleigh number is above a critical value, Ra_c, the static solution becomes unstable to any small disturbance. The system then turns from *conduction* to *convection*. In [10] some properties of non-linear thermal convection are analyzed and compared with experimental observations.

Similar to section 3.3, in two-dimensional space, let $\boldsymbol{u} = (u, v)$ and define $\boldsymbol{q} = (u, v, T) = (\boldsymbol{u}, T)$. The projection method for the Boussinesq approximation can be described in the following:

Step B1. Use the centered values $\boldsymbol{Q}^n = (U^n, V^n, T^n)$ and edge values $\boldsymbol{u}^n = (u^n, v^n)$ to solve the advection equation

$$\frac{\boldsymbol{Q} - \boldsymbol{Q}^n}{\triangle t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{Q} = 0. \tag{3.44}$$

Similar to Step V1, this is solved by the high-resolution algorithm from *CLAWPACK*, and the resulting solution is $\boldsymbol{Q}^\dagger = (\boldsymbol{U}^\dagger, T^\dagger) = (U^\dagger, V^\dagger, T^\dagger)$.

Step B2. Use a Crank-Nicolson discretization for the thermal diffusion and the diffusive term.

$$\frac{\boldsymbol{T}^{n+1} - \boldsymbol{T}^\dagger}{\triangle t} = \frac{1}{2}(\nabla^2 \boldsymbol{T}^{n+1} + \nabla^2 \boldsymbol{T}^\dagger), \tag{3.45}$$

and

$$\frac{\boldsymbol{U}^* - \boldsymbol{U}^\dagger}{\triangle t} = \frac{Pr}{2}(\nabla^2 \boldsymbol{U}^* + \nabla^2 \boldsymbol{U}^\dagger) + Pr\, Ra\, T^{n+1}\, e_z. \tag{3.46}$$

Both **Step B1** and **Step B2** are subject to boundary conditions accordingly.

Step B3. Obtain the edge velocity $\boldsymbol{u}^{n+1}$ by

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - Pr\, \mathcal{G}(\triangle t \phi^{n+1}), \tag{3.47}$$

and update the centered velocity

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* - Pr\, \mathcal{G}^0(\triangle t \phi^{n+1}). \tag{3.48}$$

Again $\boldsymbol{u}^*$ is the average of the adjacent $\boldsymbol{U}^*$, and ϕ^{n+1} is obtained by solving (3.19) with Neumann boundary condition (2.8). $\mathcal{G}$ and $\mathcal{G}^0$ are defined as before.

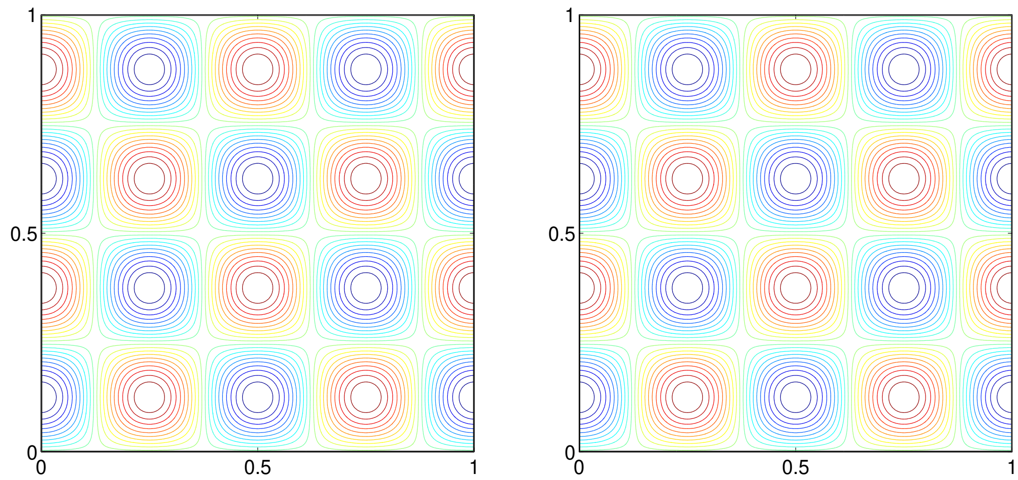
3.5 Numerical results

In this section, several examples are presented that validate the convergence properties of the methods developed in this chapter. We demonstrate the performance of the methods via numerical results and show their potential for solving more realistic problems.

Example 3.1 The first example is the stationary Taylor's vortices [94]. We use this example to demonstrate the rate of convergence of our methods. A stationary inviscid flow is considered to show that our method is suitable for fluids at high Reynolds numbers.

In the absence of viscosity, an exact steady solution of the incompressible Navier-Stokes equations in a periodic unit square is given by

$$\begin{aligned} u(x, y) &= -\cos(2m\pi x)\sin(2m\pi y) && (3.49)\\ v(x, y) &= \sin(2m\pi x)\cos(2m\pi y), && (3.50) \end{aligned}$$



(a) t=0 (b) t=1

Figure 3.4: Velocity contour for the Taylor's vortex

where m is some integer. We consider the case $m = 2$. Using (3.49) and (3.50) as initial data, we show the results for time $t = 1$. Ideally the solution would be unchanged. Table 3.1 shows the l_2 and infinity norms for the error of the u-component velocity. The l_2 norm is defined by

$$\parallel u \parallel_2 = \sqrt{\frac{\sum |u|^2}{m_x m_y}}, \tag{3.51}$$

where m_x and m_y are the numbers of grid cells in x and y directions respectively.

The CFL number is fixed at 0.9. It appears that the method is formally second-order accurate. The method used to compute the results we show here has the same formulation as the *projection method 2* in Section 2.1.2. It is stable for $0 < \gamma \leq 2/3$. We choose $\gamma = 2/3$ and a MC limiter is used to eliminate the numerical oscillation. The projection method uses the formulation resembling the *projection method 1* has the similar result.

Table 3.1: Convergence rate for the projection method based on high-resolution algorithms

	32×32	rate	64×64	rate	128×128	rate	256×256	rate	512×512
$\parallel u - u_{exact} \parallel_2$	5.64E-2	1.91	1.50E-2	3.20	1.63E-3	2.54	2.81E-4	2.35	5.50E-5
$\parallel u - u_{exact} \parallel_\infty$	1.14E-1	1.72	3.47E-2	3.13	3.96E-3	2.60	6.46E-4	2.47	1.17E-4

Figure 3.4 shows the $u-$component velocity contours. The computation is done using a 256×256 grid. Because of the doubly periodic boundaries, a FFT method is used to solve both the diffusion and the Poisson problem. The interval between plotted contour lines is 0.1. There are 20 contour lines with equally spaced values between $\pm$ 1.

Example 3.2. The second example is the bench-mark lid-driven cavity flow, which has been intensively studied. It is a hard test problem, particularly at high Reynolds numbers, due to the two singular points at the upper corners. In [45], several finite difference schemes based on projection methods were tested and compared with [17]. They all delivered weaker flows than desired. Comparing with [45], our methods gives a stronger primary vortex. For Fig 3.5, we use a uniform 256×256 grid and the Reynolds number is Re=5000. The CFL number is 0.9. Figure 3.5 is almost identical to Fig 3 in [17], except the top left vortex. The primary vortex is slightly weaker than the one in [17], but the difference is in a satisfactory range. We use the same set of the contour values as in [17] for comparison. Stronger primary vortices than Figure 3.5 can be achieved by decreasing the CFL number. No limiter is used for this problem. The Poisson problem is solved by a Fast Poisson Solver.

Table 3.2: Values for stream function contour for Fig 3.5

Contour letter	values	Contour number	values
d	-1.0E-4	5	1.0E-4
e	-1.0E-2	6	2.5E-4
f	-3.0E-2	7	5.0E-4
g	-5.0E-2	8	1.0E-3
h	-7.0E-2	9	1.5E-3
i	-9.0E-2	10	2.7E-3
j	-1.0E-1		
k	-1.1E-1		

Example 3.3. The example shown here is the Rayleigh-Taylor problem first documented by Tryggvason [99] for inviscid fluids using boundary integral methods. Later Quartapelle etc. [41, 46] repeat the same calculation for viscous fluids by means of finite elements and finite volumes. The problem consists of two layers of fluid initially at rest in a gravity field. A heavy fluid is put on top of the light fluid. The heavy fluid accelerates into the light fluid under the action of gravity. The domain for the fluid is $(-d/2, d/2) \times (2d, 2d)$ and the initial position of the interface of the layers is given by

$$\eta(x) = -0.1\, d\, \cos\left(\frac{2\pi x}{d}\right).$$

The density ratio of heavy fluid to light fluid is 3 which makes the Atwood number to be 0.5. The Atwood number At is defined as $At = (\rho_{max} - \rho_{min})/(\rho_{max} + \rho_{min})$. In order

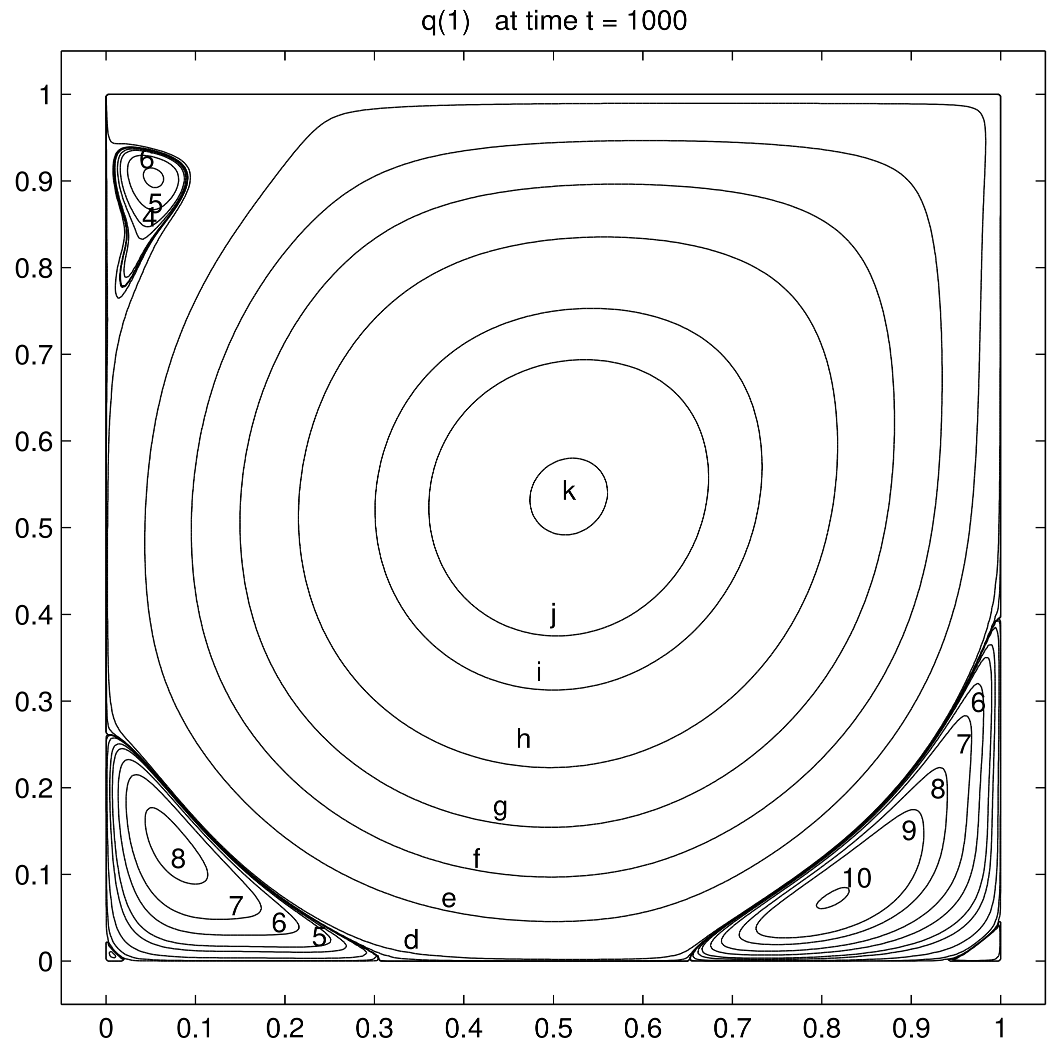


Figure 3.5: Stream function contour.

to compare with the results in [41, 46], we regularize the transition between the two fluids by a tanh profile:

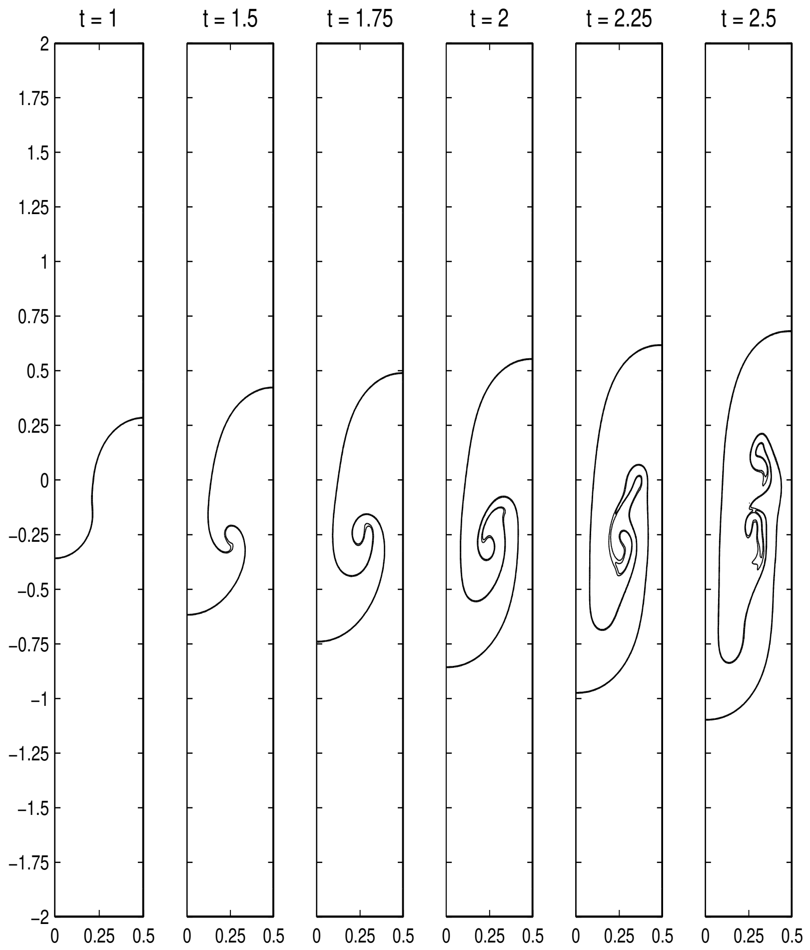
$$\frac{\rho(x,y,t=0)}{\rho_{min}} = 2 + \tanh\left(\frac{y-\eta(x)}{0.01d}\right).$$

The governing equations are made dimensionless by using the following references: ρ_{min} for density, d for length, $d^{1/2}g^{-1/2}$ for time, where g is the magnitude of gravity field, and g for gravity. The reference velocity is $d^{1/2}g^{1/2}$ and the Reynolds number is defined by $Re = \rho_{min}d^{2/3}g^{1/2}/\mu$. No-slip boundary conditions are applied to the top and bottom walls while periodic boundary conditions are imposed on the two vertical sides. The output time $\hat{t}$ is in the scale of Tryggvason. $\hat{t}$ is related to our dimensionless time scale t by $\hat{t} = t\sqrt{At}$.

Figure 3.6 are the results for $Re = 1000$ which use 128×512 cells. In order to compare the positions of the falling jet and the uprising bubble of our results with [41, 46], we only plot a half of the domain, which is composed of 32768 cells. Both the positions and the development of the roll-up structure are comparable to the results in [41, 46], for which 30189 $\mathbb{P}_2$ nodes are used. There are no noticeable differences between the two sets of results, except our results reveal more detailed roll-up structure. One thing is worth pointing out. In [41, 46], a rather small time step $\triangle\hat{t} = 5 \times 10^{-4}$ is chosen for the spatial discretization. Our CFL number is fixed at 0.9 and time step can be chosen by $\triangle\hat{t} \sim 6 \times 10^{-3}$. The MC limiter is used for both velocity and density field. The linear system (3.38) is solved by an incomplete Cholesky conjugate-gradient method.

Based on their results, [41, 46] indicate that an accurate and detailed prediction of the large-time phenomena for the viscous Rayleigh-Taylor instability at high Reynolds numbers is very difficult. They report that for $\hat{t} \geq 1.5$ and $Re \geq 5000$, the numerical results of this problem are not only sensitive to the grid resolutions but also to the numerical methods used to compute those results. The solutions computed by either the finite element method or the finite volume method for Reynolds number $Re = 5000$ are very different from the inviscid one reported in [99] at or beyond $\hat{t} = 1.5$. They are even very different from each other. The authors speculate that this problem has no smooth inviscid solution for large time, based on the Birkhoff's conjecture. Birkhoff's conjecture says that the initial-value problem for inviscid stratified flows might be ill-posed, as a consequence of the fact that the growth rate of an infinitely small unstable wave is proportional to the square root of its wave number. Figure 3.7 shows the solutions computed by our algorithm for $Re = 5000$. (a) uses the MC limiter and (b) uses the superbee limiter to eliminate the numerical oscillation due to the density discontinuity. In contrast to the one reported in [41, 46], we find our results are compatible with [99] before $\hat{t} = 2$, especially the one using the MC limiter. However, after $\hat{t} = 2$ we did observe that the results are sensitive to the limiters used for the computation. We conclude that it is hard to give an accurate and detailed prediction of the large-time phenomena for this problem at high Reynolds numbers, as [41, 46] already indicated.

Because this example does not consider the surface tension that is a curvature-dependent effect, no interface tracking scheme is required. When the surface tension is taken into account, different approaches such as the volume of fluid method [87], a front-tracking algorithm [85], or the immersed interface method developed in the next chapters will be needed to do this work.



(a) (b) (c) (d) (e) (f)

Figure 3.6: Density profile of the Rayleigh-Taylor instability. $Re = 1000$; Atwood number is 0.5 (density ration=3). Only a half of the computational domain is shown above. The initial amplitude is 10% of the wavelength. Density contours shown are $1.4 \leq \rho \leq 1.6$.

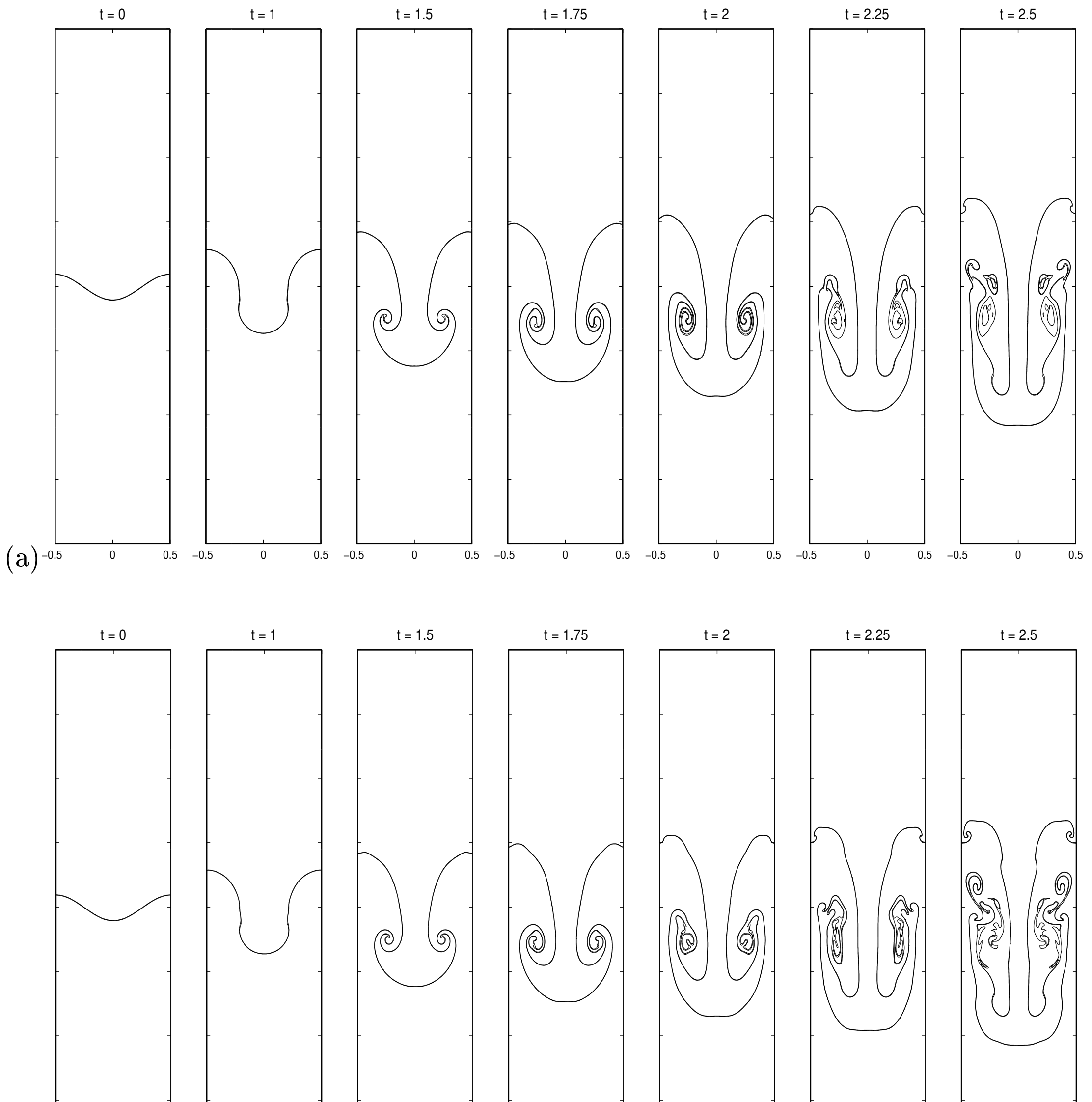


Figure 3.7: Re=5000; density ratio 3. The grid is 128×512. The interface is shown at times 0, 1, 1.5, 1.75, 2, 2.25, and 2.5 using the density contours $1.4 \leq \rho \leq 1.6$. The initial amplitude is 10% of the wavelength. (a) is computed with the MC limiter; (b) is computed with the superbee limiter.

Example 3.4. The last example of this chapter is two-dimensional Rayleigh-Bénard convection. Linear stability theory has predicted that for both rigid top and bottom while the horizontal is infinite, at the onset of instability the critical Rayleigh number is $Ra = 1707.762$ and the corresponding dimensionless critical wave number is $a = 3.117$. This implies that the convection roll develops most readily in cells with an aspect ratio of $2\pi/a = 2.016$ (see [15] page 43). In order to compare with the linear stability theory, we set periodic boundary conditions in the horizontal direction for both hydrodynamic and thermal equations. No-slip boundary conditions are imposed on the top and bottom for the velocity while the temperature for the top is -0.5 and 0.5 for the bottom. The channel is chosen with an aspect ratio 2:1.

We compute three different cases with Rayleigh numbers, $Ra = 5,000$, 10,000, and 50,000 respectively. They are all above the predicted critical Rayleigh number. We expect Rayleigh-Bénard convection to occur. Figure 3.8 shows the temperature contours in Rayleigh-Bénard convection for the three cases. While comparing with the results in [47], we see that for $Ra = 5,000$ and 10,000 our results are identical to those in [47]. However, for $Ra = 50,000$, our result is more reasonable than theirs. Their result shows the loss of symmetry of the front between hot and cold fluids. For comparison, our computational grid is 80×40 and the Prantal number is $Pr = 0.71$. Both are the same as [47].

3.6 Conclusions

We presented a class of projection methods based on the high-resolution algorithms. In the following chapters, the projection method is used as the underlying fluid solver for our numerical algorithms involving immersed interfaces. In addition, in order to study a class of problems that require capturing some fine local structures, it will be one of our great interests to develop an adaptive-mesh-refinement (AMR) projection method. Because AMR versions of the wave-propagation algorithms *AMRCLAW* and *BEARCLAW* [6, 65] are available, to develop an AMR projection method based on our algorithms will be feasible.

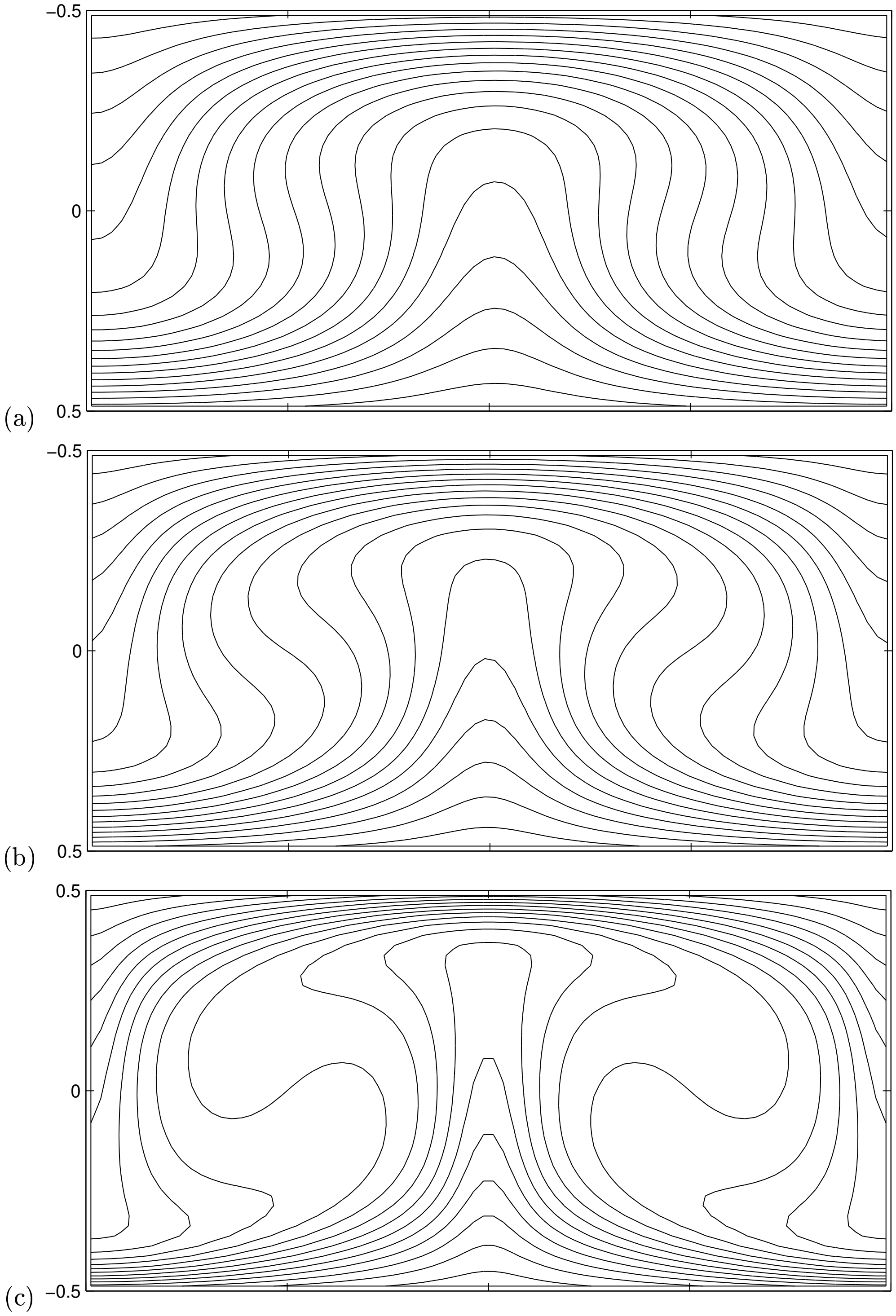


Figure 3.8: The temperature contour of the Rayleigh-Bénard convection. (a) Ra=5,000, (b) Ra=10,000, and (c) Ra=50,000. A total of 21 equally spaced contour lines that are of the values between -0.5 and 0.5 (including -0.5 and 0.5) for each plot.

Chapter 4

IMMERSED INTERFACE METHODS FOR INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

The method developed in this chapter is motivated by Peskin's immersed boundary (IB) method, and allows one to model the motion of flexible membranes or other structures immersed in viscous incompressible fluid using a fluid solver on a fixed Cartesian grid. The IB method uses a set of discrete delta functions to spread the entire singular force exerted by the immersed boundary to the nearby fluid grid points. Our method instead incorporates part of this force into jump conditions for the pressure, avoiding discrete dipole terms that adversely affect the accuracy near the immersed boundary. This has been implemented for the two-dimensional incompressible Navier-Stokes equations using a high-resolution finite-volume method for the advective terms and a projection method to enforce incompressibility. In the projection step, the correct jump in pressure is imposed in the course of solving the Poisson problem. This gives sharp resolution of the pressure across the interface and also gives better volume conservation than the traditional IB method. Comparisons between this method and the IB method are presented for several test problems. Numerical studies of the convergence and order of accuracy are included.

4.1 Introduction

The IB method represents the flexible membrane using Lagrangian markers at which force-densities are computed, based on the position of the membrane, and spread to a Cartesian fluid grid by a set of discrete delta functions. The support of the delta function is comparable to the mesh width. The Navier-Stokes equations with this forcing are then advanced one time step on the Cartesian grid using a projection method. The velocities are first updated ignoring the pressure gradient term in the Navier-Stokes equations in a *transport step*, and then a *projection step* is applied to compute the pressure and impose the incompressibility constraint on the velocity field. The resulting velocities are interpolated back to the Lagrangian markers using the same set of discrete delta functions. The markers are moved based on these velocities and the process is repeated in the next time step. This algorithm is written out in more detail in Section 4.3.

Spreading the singular boundary force to the fluid grid results in large forces at the fluid grid points closest to the boundary. Using this force field to update velocities leads to large unphysical velocities at these points. The discrete divergence of this velocity field has a discrete dipole behavior near the boundary. This is used as the right hand side of a Poisson problem in the projection step of the algorithm, which projects to the space of divergence-free velocity fields and computes the pressure in the process. In general the pressure should be discontinuous across the interface and the computed pressure has approximately this behavior due to the discrete dipole nature of the right hand side. While relatively simple to implement, this smearing of the force gives a method that is generally only first-order accurate.

We have found that the accuracy can be greatly improved by handling part of the singular force in a different manner, imposing the proper jump condition on the pressure directly in the course of solving the Poisson problem and eliminating the discrete dipole nature of the forcing term. The jump in pressure is determined by the normal component of the force exerted by the interface. We still use discrete delta functions to spread the tangential component of the force. To impose the jump in pressure while solving the Poisson problem, we use an approach based on the *immersed interface method* (IIM) of LeVeque and Li [63], which was applied to Stokes flow problems in [64]. For the application here, to a constant-coefficient Poisson problem, the complicated 6-point stencil of [63] is not needed and the approach is closely related to Mayo's method for Poisson problems on irregular regions [70].

We refer to our method below as the IIM, though there are other ways in which immersed interface methods could be used in the context of incompressible Navier-Stokes equations. In particular, Li and Lai [68] have recently developed a method that dispenses with discrete delta functions entirely. Their approach is contrasted with ours in Section 4.3. Our method also avoids discrete delta functions entirely in the case where the force is purely in the normal direction, as discussed at the end of Section 4.2. This is a special case, but one that often arises in practice, e.g., when the force results from surface tension at a free boundary between different fluids.

In Section 4.2 we review the nature of the solution near an interface where singular forces are applied, and present the jump condition needed in our algorithm. The basic explicit form of the algorithm is presented in Section 4.3. The correction terms required to implement the jump conditions are discussed further in Section 4.4. The case when the membrane has mass is presented in Section 4.5, and an implicit version of the algorithm is developed in Section 4.6. In Section 4.7 we show that our approach maintains certain steady states exactly and does not suffer from the leaking across a pressurized membrane that can be a problem with the IB method. The relation between our approach and the blob projection method of Cortez and Minion [24] is briefly explored in Section 4.8. Finally, some numerical results are presented in Section 4.9.

4.2 Jump conditions across the interface

The two-dimensional incompressible Navier-Stokes equations take the form

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p &= \mu\nabla^2\boldsymbol{u} + \boldsymbol{F} \\ \nabla\cdot\boldsymbol{u} &= 0 \\ \boldsymbol{u} &= \boldsymbol{b} \quad \text{on } \partial\Omega. \end{aligned} \tag{4.1}$$

We assume that the density $\rho \equiv 1$ is constant throughout the fluid, as is the viscosity μ. We impose Dirichlet boundary conditions on the velocity, e.g., $u = 0$ on no-slip boundaries, though periodic or Neumann conditions could also be used. We consider these equations in two space dimensions and use bold symbols to denote vectors, with the convention $\boldsymbol{u} = (u, v)$ for the velocity field and $\boldsymbol{x} = (x, y)$ for spatial position.

The vector $\boldsymbol{F}$ represents an applied body force. In this paper we consider problems where this is a singular distribution applied along some interface $\Gamma(t)$ immersed in the fluid. This has the form

$$\boldsymbol{F}(x,t) = \int_{l_0}^{l_1} \hat{\boldsymbol{f}}(l,t)\, \delta(\boldsymbol{x} - \hat{\boldsymbol{X}}(l,t))\, dl, \tag{4.2}$$

where $\hat{\boldsymbol{X}}(l,t)$ is the parameterization of $\Gamma(t)$ in terms of arclength l, and $\hat{\boldsymbol{f}}(l,t)$ is the force per unit length along Γ exerted by the interface at the point $\hat{\boldsymbol{X}}(l,t)$. The strength depends on the nature of the interface. Later in this section we present the formulas for an elastic membrane.

The singular force along Γ results in the solution to the equations being nonsmooth across the interface. Since we assume that the interface Γ moves with the fluid and there is no fluid slip at the interface, the velocity is continuous and $[\![\boldsymbol{u}]\!] = 0$ across the interface, where $[\![\cdot]\!]$ denotes the jump in a quantity across the interface. There may, however, be jumps in the pressure and in the normal derivatives of both the pressure and velocity at the interface. Let $\boldsymbol{n}(l,t)$ be the unit normal vector to Γ and $\boldsymbol{\tau}(l,t)$ the tangent vector, so that the normal and tangential components of the force density are

$$\begin{aligned} \hat{\boldsymbol{f}}_n(l,t) &= \hat{\boldsymbol{f}}(l,t) \cdot \boldsymbol{n}(l,t) \\ \hat{\boldsymbol{f}}_\tau(l,t) &= \hat{\boldsymbol{f}}(l,t) \cdot \boldsymbol{\tau}(l,t). \end{aligned} \tag{4.3}$$

Corresponding forces $\boldsymbol{F}_n(x,t)$ and $\boldsymbol{F}_\tau(x,t)$ can be obtained by integrating these as in (4.2). Then at each point on Γ the solution to the incompressible Navier-Stokes equations satisfies the jump conditions

$$\begin{aligned} [\![p]\!](l,t) &= \hat{\boldsymbol{f}}_n(l,t) \\ \left[\!\!\left[\frac{\partial p}{\partial n}\right]\!\!\right](l,t) &= \frac{\partial}{\partial l}\hat{\boldsymbol{f}}_{\boldsymbol{\tau}}(l,t) \\ [\![\boldsymbol{u}]\!](l,t) &= 0 \\ \mu\left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial n}\right]\!\!\right](l,t) &= -\hat{\boldsymbol{f}}_{\boldsymbol{\tau}}(l,t)\boldsymbol{\tau}(l,t). \end{aligned} \tag{4.4}$$

These jump conditions are derived in Chapter 5.

It is often more convenient to use a parameterization s of the interface that flows with the fluid, so that the interface location $\boldsymbol{X}(s,t)$ satisfies

$$\frac{\partial}{\partial t}\boldsymbol{X}(s,t) = \boldsymbol{u}(\boldsymbol{X}(s,t),t), \tag{4.5}$$

where $\boldsymbol{u}(x,t)$ is the fluid velocity. We assume that there is no flow of fluid through the interface, and also that there is no slip along the interface so that the velocity $\boldsymbol{u}(\boldsymbol{X}(s,t),t)$ is uniquely defined. In practice we track the interface via a set of marker points $\boldsymbol{X}_k$ that are associated with fixed values s_k and that move with the fluid. In this case s is typically not arclength. Then the force formula (4.2) is replaced by

$$\boldsymbol{F}(x,t) = \int_{s_0}^{s_1} \boldsymbol{f}(s,t)\, \delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\, ds, \tag{4.6}$$

where $\boldsymbol{f}(s,t)$ is now the force density with respect to the measure ds. The force per unit length is given by

$$\frac{\boldsymbol{f}(s,t)}{|\partial \boldsymbol{X}(s,t)/\partial s|}. \tag{4.7}$$

Three of the jump conditions from (4.4) transform easily to expressions in terms of the normal and tangential components of $\boldsymbol{f}(s,t)$,

$$\begin{aligned} [\![p]\!](s,t) &= \boldsymbol{f}_n(s,t)/|\partial \boldsymbol{X}(s,t)/\partial s| \\ [\![\boldsymbol{u}]\!](s,t) &= 0 \\ \mu \left[\!\!\left[\frac{\partial u}{\partial n}\right]\!\!\right](s,t) &= -\boldsymbol{f}_{\boldsymbol{\tau}}(s,t)\boldsymbol{\tau}(s,t)/|\partial \boldsymbol{X}(s,t)/\partial s|. \end{aligned} \tag{4.8}$$

The transformed version of $[\![\partial p/\partial n]\!]$ is more complicated and is not derived in this thesis. In our formulation, however, we do not explicitly use this jump condition.

As a particular form of interface problem, we consider the case of an elastic membrane under tension, with s representing arclength along the unstretched membrane and the force density

$$\boldsymbol{f}(s,t) = \frac{\partial}{\partial s}(T(s,t)\boldsymbol{\tau}(s,t)), \tag{4.9}$$

where $\boldsymbol{\tau}(s,t)$ is the unit tangent vector and the tension is given by

$$T(s,t) = T_0\left(\left|\frac{\partial \boldsymbol{X}(s,t)}{\partial s}\right| - 1\right), \tag{4.10}$$

with T_0 constant. See Peskin [78] for a derivation.

Other types of interfaces give rise to forces specified in different ways. In particular, if the interface is simply a free surface separating two different fluids, then the force will be given by surface tension and typically has the form

$$\boldsymbol{f}(s,t) = \gamma \frac{\partial^2}{\partial l^2}\boldsymbol{X}(l,t), \tag{4.11}$$

where γ is some constant. This force points purely in the normal direction. Note that in this case we can dispense with discrete delta functions entirely. Of course in such problems the two fluids typically have different densities and viscosities, which introduces other potential difficulties near the interface. This can be handled with an extension of our algorithm that is currently being developed.

4.3 The numerical algorithm

Consider the incompressible Navier-Stokes equations with a singular force $\boldsymbol{F}$. We propose a fractional step method with the following basic form.

Transport step. Advance $\boldsymbol{U}^n$ to $\boldsymbol{U}^*$ by solving the advection-diffusion equation

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \mu\nabla^2\boldsymbol{u} + \boldsymbol{F}_1 \tag{4.12}$$

over time Δt. This equation contains part or all of the force $\boldsymbol{F}$ as discussed below.

Projection step. Advance $\boldsymbol{U}^*$ to $\boldsymbol{U}^{n+1}$ by

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* - \Delta t \nabla p^{n+1} + \Delta t \boldsymbol{F}_2, \tag{4.13}$$

where $\boldsymbol{F}_2 = \boldsymbol{F} - \boldsymbol{F}_1$ is the remainder of the force. This is a discretization of

$$\boldsymbol{u}_t + \nabla p = \boldsymbol{F}_2. \tag{4.14}$$

The pressure p^{n+1} is determined by solving the Poisson problem that arises from applying the discrete divergence operator to (4.13) and using the fact that we want $\nabla \cdot \boldsymbol{U}^{n+1} = 0$. This results in

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{U}^* + \nabla \cdot \boldsymbol{F}_2. \tag{4.15}$$

The original immersed boundary method of Peskin is of this form, with $\boldsymbol{F}_1 = \boldsymbol{F}$ and $\boldsymbol{F}_2 = 0$. It is implemented by spreading the singular force $\boldsymbol{F}$ to the uniform grid points by means of a discrete delta function. The intermediate velocity $\boldsymbol{U}^*$ is typically not divergence free since both the nonlinear term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ and the force $\boldsymbol{F}$ may fail to be. The projection step, with $\boldsymbol{F}_2 = 0$, then corresponds to a projection of $\boldsymbol{U}^*$ to $\boldsymbol{U}^{n+1}$, the divergence-free portion.

The main new feature of the algorithm we are proposing is that we split the singular force $\boldsymbol{F}$ into components tangential and normal to the interface via (4.3), and use these components for $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ respectively. The tangential component $\boldsymbol{F}_1 = \boldsymbol{F}_\tau$ is spread to the grid using discrete delta functions as in the immersed boundary method. The normal component $\boldsymbol{F}_2 = \boldsymbol{F}_n$ is not spread to the grid. Instead, we use the fact that the jump in pressure across the interface should be equal to the normal component of the force. This jump condition is built into the Poisson solver in Step 2 using the immersed interface approach of LeVeque and Li [63, 64]. We solve the equation

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{U}^* \tag{4.16}$$

together with the proper jump condition in p, so that accurate pointwise values of the pressure can be computed even near the interface.

Note that since $\boldsymbol{F}_2$ has the form of a delta function, $\nabla \cdot \boldsymbol{F}_2$ is a dipole along the interface, which does indeed lead to a jump in p in the solution. In the original immersed boundary method, this jump arises in the following manner: The entire force is spread to the grid, giving discrete delta function behavior in the intermediate velocity $\boldsymbol{U}^*$. Applying the discrete divergence to this leads to discrete dipole behavior in $\nabla \cdot \boldsymbol{U}^*$, and using this as the right hand side in the Poisson problem leads to a smeared jump in the numerical approximation to the pressure. Updating $\boldsymbol{U}^*$ by the discrete version of ∇p then should remove the nonphysical dipole behavior from the velocity. We believe that this discrete dipole in $\boldsymbol{U}^*$ and its use in the Poisson problem are the principle causes of inaccuracy in the immersed boundary method and that much better accuracy can be achieved by avoiding these discrete dipoles, instead imposing the correct jump conditions directly on p.

A more extreme version of this is proposed by Li and Lai [68], who impose all the jump conditions from (4.4) directly on the velocity and pressure and avoid any use of discrete delta functions. Their approach may give somewhat better accuracy than ours, but is more complicated to implement in general (only certain special cases are considered in [68]). We have found that simply handling the pressure jump properly gives a substantial improvement over the original immersed boundary method. This is a relatively simple change that could be applied to many existing immersed boundary codes.

Our implementation of the IIM uses the projection method developed in Chapter 3. This new projection method is based on a combination of high-resolution finite volume methods for the advection portion of the equations and finite difference methods for the immersed interface solution of the Poisson problem. It is a "pressure-free projection method" discussed in Section 2.1.1.

Recall that in this projection method we maintain two different velocity fields, cell-centered values $\boldsymbol{U}^n = (U^n_{ij}, V^n_{ij})$ that represent cell averages over the (i,j) grid cell, and edge values $u^n_{i-1/2,j}$ and $v^n_{i,j-1/2}$ that represent approximations to the normal velocity at the left and bottom edge of the (i,j) grid cell, respectively. At the beginning of each time step, we assume that the edge velocities $\boldsymbol{u}^n$ are discrete divergence free, $\nabla_h \cdot \boldsymbol{u}^n = 0$, where

$$(\nabla_h \cdot \boldsymbol{u}^n)_{ij} = \frac{u^n_{i+1/2,j} - u^n_{i-1/2,j}}{\Delta x} + \frac{v^n_{i,j+1/2} - v^n_{i,j-1/2}}{\Delta y}. \tag{4.17}$$

At the beginning of the time step we also have a set of points $\boldsymbol{X}^n_k$ for $k = 1,\ 2,\ ,\ \ldots,\ N$ that mark the location of the interface. This vector of points is denoted $\boldsymbol{X}^n$.

Assuming the viscosity is uniform across the interface and the density is $\rho \equiv 1$, the algorithm then takes the following form.

Step E1. Solve the advection equation

$$\boldsymbol{U}_t + (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{U} = 0 \tag{4.18}$$

over time Δt to update the cell averages $\boldsymbol{U}^n$ to values $\boldsymbol{U}^\dagger$. We use the high-resolution finite volume method implemented in CLAWPACK [59, 61, 62]. The edge velocities $\boldsymbol{u}^n$ are used in solving Riemann problems at the cell edges. Limiters are then applied to avoid unphysical oscillations near steep gradients.

Step E2. Use $\boldsymbol{U}^\dagger$ as data to solve the diffusion equation along with the tangential portion of the force,

$$\boldsymbol{U}_t = \mu \nabla^2 \boldsymbol{U} + \boldsymbol{F}_\tau. \tag{4.19}$$

The force is transferred from the interface to nearby cell centers using the discrete delta function

$$d_h(\boldsymbol{x}) = \delta_h(x)\delta_h(y), \qquad \text{with } \delta_h(r) = \begin{cases} \dfrac{1}{4h}\left(1 + \cos\dfrac{\pi r}{2h}\right) & |r| \le 2h, \\ 0 & |r| \ge 2h. \end{cases} \tag{4.20}$$

The equation (4.19) is then discretized using a Crank-Nicolson method over time Δt to yield new cell-centered values $\boldsymbol{U}^*$.

Step E3. Average $\boldsymbol{U}^*$ from adjacent grid cells to obtain edge values $\boldsymbol{u}^*$,

$$\begin{aligned} u^*_{i-1/2,j} &= \frac{1}{2}(U^*_{i-1,j} + U^*_{ij}), \\ v^*_{i,j-1/2} &= \frac{1}{2}(V^*_{i,j-1} + V^*_{ij}). \end{aligned} \tag{4.21}$$

In general $\nabla_h \cdot \boldsymbol{u}^* \neq 0$.
Step E4. Compute p^{n+1} by solving a discrete version of

$$\Delta t\, \nabla^2 p^{n+1} = \nabla \cdot \boldsymbol{u}^* \tag{4.22}$$

with the jump condition $[\![p^{n+1}]\!] = f_n(s,t)/|\partial \boldsymbol{X}(s,t)/\partial s|$. This is accomplished by solving the discrete system

$$\nabla_h^2 p_{ij}^{n+1} = \frac{1}{\Delta t}(\nabla_h \cdot \boldsymbol{u}^*)_{ij} + C_{ij} \tag{4.23}$$

where ∇_h^2 is the discrete 5-point Laplacian and C_{ij} are correction terms needed to accurately impose the jump conditions on p. These values will only be nonzero in cells near the interface and are specified below.
Step E5. Update the edge velocities by

$$\begin{aligned} u_{i-1/2,j}^{n+1} &= u_{i-1/2,j}^* - \Delta t[(p_{ij} - p_{i-1,j})/\Delta x - B_{i-1/2,j}^1] \\ v_{i,j-1/2}^{n+1} &= v_{i,j-1/2}^* - \Delta t[(p_{ij} - p_{i,j-1})/\Delta y - B_{i,j-1/2}^2]. \end{aligned} \tag{4.24}$$

The correction terms $B_{i-1/2,j}^1$ and $B_{i,j-1/2}^2$ are needed to take into account the jump conditions in p so that accurate pointwise values of the derivative are captured,

$$\begin{aligned} (p_{ij} - p_{i-1,j})/\Delta x - B_{i-1/2,j}^1 &\approx p_x(x_{i-1/2}, y_j) \\ (p_{ij} - p_{i,j-1})/\Delta y - B_{i-1/2,j}^2 &\approx p_y(x_i, y_{j-1/2}). \end{aligned} \tag{4.25}$$

These terms are derived below.
Step E6. Update the cell-centered velocities by

$$\begin{aligned} U_{ij}^{n+1} &= U_{ij}^* - \Delta t[(p_{i+1,j} - p_{i-1,j})/(2\Delta x) - B_{ij}^1] \\ V_{ij}^{n+1} &= V_{ij}^* - \Delta t[(p_{i,j+1} - p_{i,j-1})/(2\Delta y) - B_{ij}^2] \end{aligned} \tag{4.26}$$

where

$$\begin{aligned} B_{ij}^1 &= \frac{1}{2}(B_{i-1/2,j}^1 + B_{i+1/2,j}^1), \\ B_{ij}^2 &= \frac{1}{2}(B_{i,j-1/2}^2 + B_{i,j+1/2}^2). \end{aligned} \tag{4.27}$$

This is consistent with the manner in which the edge velocities are updated and gives the pressure corrections for the cell-centered velocities.
Step E7. Interpolate the velocities to the Lagrangian points $\boldsymbol{X}_k^n$ that mark the interface location. Bilinear interpolation from cell-centered velocities $\boldsymbol{U}_{ij}^{n+1}$ is used to evaluate velocities at the points $\boldsymbol{X}^n$. The resulting vector of velocities is denoted by $\mathcal{U}^{n+1}$. The marker points are then moved with the fluid,

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \Delta t\, \mathcal{U}^{n+1}(\boldsymbol{X}^n). \tag{4.28}$$

A more stable implicit method is described in Section 4.6 below, in which this is replaced by

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \frac{1}{2}\Delta t\, (\mathcal{U}^n(\boldsymbol{X}^n) + \mathcal{U}^{n+1}(\boldsymbol{X}^{n+1})). \tag{4.29}$$

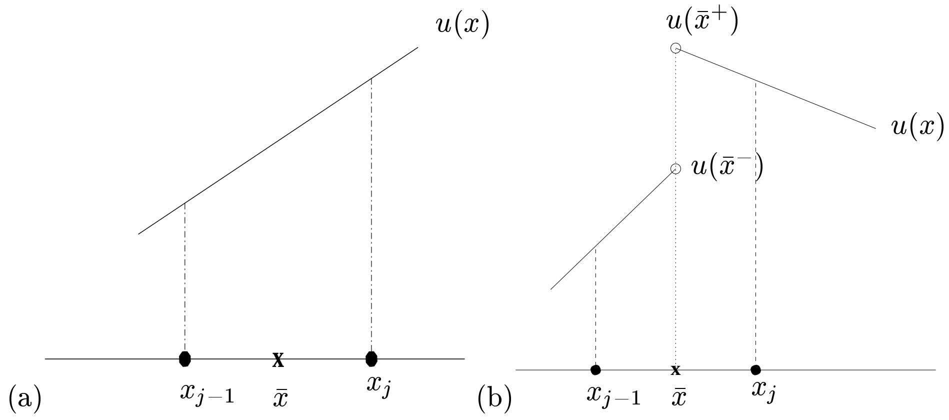


Figure 4.1: (a) A 1-D smooth solution. (b) Jumps in a 1-D solution and its derivative.

This completes a time step, and we return to Step 1.

We should note that the quantity p^{n+1} computed by this algorithm is probably at most a first-order accurate approximation to the pressure. This quantity is called ϕ^{n+1} in [9] and they note, following Kim and Moin [55], that for "pressure-free projection methods" a more accurate approximation to the pressure is obtained as

$$P^{n+1/2} = \phi^{n+1} - \frac{\mu \Delta t}{2} \nabla^2 \phi^{n+1}. \tag{4.30}$$

This does not affect the accuracy of the velocities, however, since our p^{n+1} is the proper term to use in the velocity updates. It is only if the pressure itself is needed to high accuracy that the post-processing correction (4.30) may be desirable.

4.4 Correction terms

Before discussing the choice of the correction terms B and C, we use the following one-dimensional example to demonstrate how jump conditions can be built in a difference scheme to obtain a correction term, when the solution or its derivative has jumps across an interface.

Consider that u is a solution to a one-dimensional steady-state differential equation. u is a function of x only. Let x_j, $j = 1, 2, ...$ be the grid points and denote $u(x_j) = u_j$. Figure 4.1 (a) shows that if the solution is smooth and x_{j-1} , x_j are close to each other, then for any $\bar{x} \in [x_{j-1}, x_j]$, we can approximate $\frac{du}{dx}|_{x=\bar{x}}$ by

$$\frac{u_j - u_{j-1}}{x_j - x_{j-1}} \approx u_x(\bar{x}). \tag{4.31}$$

However, Figure 4.1 (b) shows that when the solution or its derivative has jumps at $x = \bar{x}$, both $u(\bar{x})$ and $u_x(\bar{x})$ are not defined at this point and the approximation (4.31) is no longer valid. Suppose that we define $u(\bar{x}^+)$ to be $\lim\limits_{x\to\bar{x}^+} u(x)$ from the right and denote $u(\bar{x}^\pm) = u^\pm$ and $u_x(\bar{x}^\pm) = u_x^\pm$. Assume that the jumps of the solution and its derivative at $\bar{x}$ are given by

$$[\![u]\!] = u^+ - u^- = \alpha, \qquad [\![u_x]\!] = u_x^+ - u_x^- = \beta. \tag{4.32}$$

Taking the Taylor series expansions for $u(x_j)$ and $u(x_{j-1})$ with respect to $\bar{x}$ to the order-two term, we have

$$u(x_j) \approx u^+ + u_x^+ (x_j - \bar{x}) \tag{4.33}$$
$$u(x_{j-1}) \approx u^- + u_x^- (x_{j-1} - \bar{x}). \tag{4.34}$$

Equation (4.33) $-$ equation (4.34) results in

$$\begin{aligned} u(x_j) - u(x_{j-1}) &\approx [\![u]\!] + [\![u_x]\!]\bar{x} + u_x^+ x_j - u_x^- x_{j-1} \\ &\approx [\![u]\!] + [\![u_x]\!] (x_j - \bar{x}) - [\![u_x]\!] x_j + u_x^+ x_j - u_x^- x_{j-1}. \end{aligned} \tag{4.35}$$

Therefore,

$$\begin{aligned} & u(x_j) - u(x_{j-1}) - [\![u]\!] - [\![u_x]\!](x_j - \bar{x}) \\ &= -u_x^+ x_j + u_x^- x_j + u_x^+ x_j - u_x^- x_{j-1} + O(h^2) \\ &= u_x^- (x_j - x_{j-1}) + O(h^2). \end{aligned} \tag{4.36}$$

If we approximate the derivative $u_x(\bar{x}^-)$ by the interpolation

$$\frac{u_j - u_{j-1} - [\![u]\!] - [\![u_x]\!](x_j - \bar{x})}{x_j - x_{j-1}}, \tag{4.37}$$

then the approximation is accurate up to $O(h)$. Similarly, we can find the interpolation for $u_x(\bar{x}^+)$.

This one-dimensional example tells us that if the interface position and the jumps in the solution or its derivative across the interface are given, we can find finite difference approximations to the derivatives at the grid points near the interface and hence the differential equation can be solved accurately by a difference scheme.

We now discuss the choice of the correction terms B and C. First we recall that at the end of the time step we need the edge velocities $\boldsymbol{u}^{n+1}$ to be divergence free in preparation for the next step. Taking the discrete divergence of the expressions (4.24) used in Step 5 yields

$$\nabla_h \cdot \boldsymbol{u}^{n+1} = \nabla_h \cdot \boldsymbol{u}^* - \Delta t\, [\nabla_h^2 p^{n+1} - \nabla_h \cdot \boldsymbol{B}]. \tag{4.38}$$

This will be zero provided that p^{n+1} is defined by solving the discrete Poisson problem (4.23) with a set of correction terms that satisfy

$$\begin{aligned} C_{ij} &= (\nabla_h \cdot \boldsymbol{B})_{ij} \\ &= \frac{B^1_{i+1/2,j} - B^1_{i-1/2,j}}{\Delta x} + \frac{B^2_{i,j+1/2} - B^2_{i,j-1/2}}{\Delta y}. \end{aligned} \tag{4.39}$$

Thus we only need to determine the correction terms B^1 and B^2 and the C terms can be computed from (4.39).

The correction terms B^1 and B^2 are determined by the requirements (4.25), together with the jump conditions on p. If the interface lies between cell centers $(i-1,j)$ and (i,j) then let $[\![p]\!]_{i-1/2,j}$ denote the jump in p at the point where the interface cuts the line segment between these cell centers. We assume we can evaluate the force and hence this jump at any point along the interface. (In practice this is done using a cubic spline interpolant through the forces computed at the marker points.) If the interface does not cut between these cell centers, then we set $[\![p]\!]_{i-1/2,j} = 0$. In either case, a Taylor series expansion then gives

$$p_{ij} = p_{i-1,j} + [\![p]\!]_{i-1/2,j} + \Delta x\, p_x(x_{i-1/2,j}, y_j) + O(\Delta x^2), \tag{4.40}$$

provided we assume that there is no jump in p_x at the interface. We will discuss this assumption in a moment. Ignoring the $O(\Delta x^2)$ term in (4.40) and comparing to (4.25) shows that we should take

$$B^1_{i-1/2,j} = \frac{[\![p]\!]_{i-1/2,j}}{\Delta x}. \tag{4.41}$$

Similarly, we find that

$$B^2_{i,j-1/2} = \frac{[\![p]\!]_{i,j-1/2}}{\Delta y} \tag{4.42}$$

where $[\![p]\!]_{i,j-1/2}$ is defined in an analogous manner, as the jump in p across the portion of the interface cutting between cell centers $(i, j-1)$ and (i,j), if it does so, and zero otherwise.

The Taylor series expansion (4.40) is valid provided that there is no jump in p_x at the interface. In general, however, there will be a jump in the derivatives of p at the interface according to the jump conditions on $[\![\partial p/\partial n]\!]$ in (4.4). By ignoring these jumps in (4.40) it appears that we are invalidating (4.25). However, because the tangential force component $\boldsymbol{f}_\tau$ is imposed on $\boldsymbol{U}^*$ using discrete delta functions, and the divergence of $\boldsymbol{u}^*$ is then used as the right hand side in the Poisson problem for p, the error incurred in (4.25) by ignoring this jump gives a term that is in fact needed to remove this singularity from $\boldsymbol{u}^*$ when computing $\boldsymbol{u}^{n+1}$ in (4.24). This is analogous to what happens in the original IB method, where the full force is imposed using discrete delta functions. In this case $B^1_{i-1/2,j} = B^2_{i,j-1/2} = 0$ is used in (4.24), even though (4.25) is not valid at all pointwise. The resulting error approximately removes the singularity in $\boldsymbol{u}^*$ and is a crucial part of the method. In our IIM, the normal force component, and resulting jumps in p, are handled more accurately, but jumps in the gradient of p are still approximated as in the IB method.

4.5 Immersed interfaces with mass

For many applications it is sufficient to model the immersed boundary as an elastic surface with zero mass. The IIM outlined in the previous sections provides a good tool to simulate the fluid dynamics at the interface for this type of problem. Nevertheless there are some problems that in order to generate desired numerical solutions to the problem, a *nonzero mass density* is preferred to model the interface immersed in the fluid. For example, to simulate the experiment in [108] it is important to assume that the immersed filament has a nonzero mass density. In [108], a filament is pinned in a flowing soap film. One can observe that Von Kármán type vortices are shed from the tip of the free end. Several simulations have been done for this experiment (e.g. Zhu and Peskin [109]; Fast and Henshaw [36]). They report that if a massless filament is modeled, the flapping frequency and amplitude of the filament computed by the simulation are different from those reported in [108].

In this section we illustrate that one can add a nonzero mass density to an immersed interface via the implicit method introduced in the next section.

Recall Section 1.2 and Section 4.2, for massless membranes, the elastic force density is given by

$$\begin{aligned} \boldsymbol{f}(s,t) &= \frac{\partial}{\partial s}\big(T(s,t)\boldsymbol{\tau}(s,t)\big) \\ &= f_n\boldsymbol{n} + f_\tau\boldsymbol{\tau}. \end{aligned} \tag{4.43}$$

Because mass times acceleration is zero, the elastic force exerted to the fluid form the membrane equals to the jump conditions in the pressure and derivative of the velocity. If we decompose the jump of the normal stress into the normal and tangential components, we have

$$\begin{aligned} [\![p]\!] &= f_n \\ \mu\left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right]\!\!\right] &= -f_\tau\boldsymbol{\tau}. \end{aligned} \tag{4.44}$$

Now consider the membrane with mass density $m(s)$. Newton's Second Law says the mass times acceleration equals to the sum of the elastic force from the membrane and the force from the fluid to the membrane, in which the action of the latter is jumps in the pressure and derivative of the velocity

$$\begin{aligned} m(s)\frac{\partial^2 \boldsymbol{X}(s,t)}{\partial t^2} &= \boldsymbol{f}(s,t) - [\![p]\!]\boldsymbol{n} + \mu\left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right]\!\!\right] \\ &= \frac{\partial}{\partial s}\big(T(s,t)\boldsymbol{\tau}(s,t)\big) - [\![p]\!]\boldsymbol{n} + \mu\left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right]\!\!\right]. \end{aligned} \tag{4.45}$$

Therefore the force now used in the fluid equations is

$$\begin{aligned} \tilde{\boldsymbol{f}}(s,t) &= \boldsymbol{f}(s,t) - m(s)\frac{\partial^2 \boldsymbol{X}(s,t)}{\partial t^2} \\ &= \frac{\partial}{\partial s}\big(T(s,t)\boldsymbol{\tau}(s,t)\big) - m(s)\frac{\partial^2 \boldsymbol{X}(s,t)}{\partial t^2}. \end{aligned} \tag{4.46}$$

Split this force into normal and tangential components

$$\tilde{\boldsymbol{f}}(s,t) = \tilde{f}_n\boldsymbol{n} + \tilde{f}_\tau\boldsymbol{\tau} \tag{4.47}$$

and the jumps in the pressure and derivative of the velocity are

$$\begin{aligned} [\![p]\!] &= \tilde{f}_n \\ \mu\left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right]\!\!\right] &= -\tilde{f}_\tau\boldsymbol{\tau}. \end{aligned} \tag{4.48}$$

In the next section we develop an implicit method, in which equations (4.44) and (4.48) are the jumps used for the massless membrane and the membrane with mass respectively.

4.6 An implicit method

In this section we discuss the manner in which the motion of the interface markers $\boldsymbol{X}_k$ can be made implicit in order to improve the stability of the method and allow reasonable time steps to be taken. We would like the time steps to be determined by the CFL condition needed for the advection step. The high-resolution method used for this step only requires that the Courant number be less than 1,

$$\max_{i,j}\left(\frac{\Delta t}{\Delta x}|u_{i-1/2,j}|,\ \frac{\Delta t}{\Delta y}|v_{i,j-1/2}|\right) \leq 1, \tag{4.49}$$

as shown in [61]. To achieve this, the viscous diffusion step must of course be done implicitly, and we use a standard Crank-Nicolson method for this in Step 2. But we must also typically make the interface motion of Step 7 implicit, which is more complicated to implement. The simple explicit approach (4.28) is based on interpolating the updated velocity field $\boldsymbol{U}^{n+1}$ to the old interface locations $\boldsymbol{X}^n$, and moving these points using forward Euler. This can lead to an unstable method when the membrane is stiff unless very small time steps are used, since a small perturbation of the interface can lead to large forces and hence large transient velocities near the interface. With the explicit method, this typically causes the membrane to overshoot its proper location, resulting in even larger forces the next time step and an exponentially growing instability. This stiffness is well known for immersed boundary methods, and various semi-implicit or implicit approaches have been used to overcome it, e.g., [71, 100]. Here we use essentially the same fully-implicit quasi-Newton approach as developed for Stokes flow in [64], although it is more complicated to implement for Navier-Stokes.

We use a trapezoidal method of the form (4.29), where $\mathcal{U}^n(\boldsymbol{X}^n)$ is obtained by interpolating $\boldsymbol{U}^n$ to the old locations $\boldsymbol{X}^n$, and $\mathcal{U}^{n+1}(\boldsymbol{X}^{n+1})$ results from interpolating the new velocity field $\boldsymbol{U}^{n+1}$ to the new locations $\boldsymbol{X}^{n+1}$. But a crucial aspect in making this fully implicit is that the new velocity field $\boldsymbol{U}^{n+1}$ is itself computed by using the new interface locations $\boldsymbol{X}^{n+1}$. In Step 2 of the algorithm, the trapezoidal method is applied to equation (4.19) in such a way that $\boldsymbol{F}_\tau$ is discretized using the average of the forces at t_n based on locations $\boldsymbol{X}^n$ and the forces at t_{n+1} based on $\boldsymbol{X}^{n+1}$. Similarly, the correction terms (4.41) and (4.42) used in the projection step are evaluated by computing the jumps in p at the old interface location and also the jump in p at the new interface location, and then averaging the two. This process is simplified by the fact that the force exerted by the membrane described by marker points $\boldsymbol{X}_k$ depends only on the location of these points, and so for given vectors $\boldsymbol{X}^n$ and $\boldsymbol{X}^{n+1}$ we can compute the forces and jumps needed. The complication is that, although $\boldsymbol{X}^n$ is already known, the correct $\boldsymbol{X}^{n+1}$ is not known when Steps 2 through 6 are being performed. Instead it is necessary to embed these steps in an iterative method of the following form.

Step I1. Apply Step E1, the advection step, which does not require $\boldsymbol{X}^{n+1}$.

Step I2. Make a guess $\boldsymbol{X}^{[0]}$ at $\boldsymbol{X}^{n+1}$ based on the previous motion of the interface, e.g.,

$$\boldsymbol{X}^{[0]} = 2\boldsymbol{X}^n - \boldsymbol{X}^{n-1}. \tag{4.50}$$

Set $I = 0$.

Step I3. Perform Steps 2 through 6 using $\boldsymbol{X}^{[I]}$ as $\boldsymbol{X}^{n+1}$. The result is a provisional velocity field $\boldsymbol{U}^{n+1}$.

Step I4. Evaluate

$$g(\boldsymbol{X}^{[I]}) = \boldsymbol{X}^{[I]} - \boldsymbol{X}^n - \frac{1}{2}\Delta t\left(\mathcal{U}^n(\boldsymbol{X}^n) + \mathcal{U}^{n+1}(\boldsymbol{X}^{[I]})\right), \tag{4.51}$$

where $\mathcal{U}^{n+1}(\boldsymbol{X}^{[I]})$ are the interpolated velocities at $\boldsymbol{X}^{[I]}$ based on the provisional velocity field $\boldsymbol{U}^{n+1}$.

Step I5. If $\|g(\boldsymbol{X}^{[I]})\| \leq \epsilon$ for some tolerance, then set $\boldsymbol{X}^{n+1} = \boldsymbol{X}^{[I]}$ and we are done with this time step. Otherwise, update $\boldsymbol{X}^{[I]}$ to $\boldsymbol{X}^{[I+1]}$, set $I = I + 1$, and go back to step I3.

Updating $\boldsymbol{X}^{[I]}$ in Step I5 is done using a quasi-Newton method, such as BFGS or SR1 (symmetric rank-one update of Broyden method) [92]. Our iteration is designed to find a zero of the function $g(\boldsymbol{X})$ defined by (4.51). The Jacobian cannot be calculated directly since evaluation of this function requires applying Steps 2 through 6 of the algorithm. But in practice we have a good estimate of the Jacobian from the previous time step and typically three or four iterations of the quasi-Newton algorithm gives convergence along with an updated approximation to the Jacobian matrix.

Since this fully implicit method can be further used to add a nonzero mass density to the membrane, we discuss the algorithm in details as follows.

- The position of the membrane at t^{n+1} is $\boldsymbol{X}^{n+1}$ that is found by equation (4.29). If we replace $\boldsymbol{X}^{n+1}$ in (4.29) by the variable $\boldsymbol{X}$, the task of finding $\boldsymbol{X}^{n+1}$ becomes solving the following nonlinear equation

$$g(\boldsymbol{X}) = \boldsymbol{X} - \boldsymbol{X}^n - \frac{\triangle t}{2}\left(\mathcal{U}(\boldsymbol{X}) + \mathcal{U}(\boldsymbol{X}^n)\right) \tag{4.52}$$

The initial guess for the position of the membrane at t^{n+1}, $\boldsymbol{X}^{n+1}$, is

$$\boldsymbol{X}^{[0],\,n+1} = 2\boldsymbol{X}^n - \boldsymbol{X}^{n-1}. \tag{4.53}$$

Using this guess, we compute the normal and tangential force density by the follows.

$\boxed{\text{For massless membranes}}$

$$\begin{aligned} f_{\boldsymbol{n}}^{[0],\,n+1} &= \frac{\partial}{\partial s}\left(T^{[0],\,n+1}\boldsymbol{\tau}^{[0],\,n+1}\right)\cdot\boldsymbol{n} \\ f_{\boldsymbol{\tau}}^{[0],\,n+1} &= \frac{\partial}{\partial s}\left(T^{[0],\,n+1}\boldsymbol{\tau}^{[0],\,n+1}\right)\cdot\boldsymbol{\tau}. \end{aligned} \tag{4.54}$$

For membranes with mass

$$
\begin{aligned}
f_{\boldsymbol{n}}^{[0],\,n+1} &= \left(\frac{\partial}{\partial s}(T^{[0],\,n+1}\boldsymbol{\tau}^{[0],\,n+1}) - m(s)\frac{\boldsymbol{X}^{[0],\,n+1} - 2\,\boldsymbol{X}^{n} + \boldsymbol{X}^{n-1}}{\triangle t^2}\right)\cdot \boldsymbol{n} \\
f_{\boldsymbol{\tau}}^{[0],\,n+1} &= \left(\frac{\partial}{\partial s}(T^{[0],\,n+1}\boldsymbol{\tau}^{[0],\,n+1}) - m(s)\frac{\boldsymbol{X}^{[0],\,n+1} - 2\,\boldsymbol{X}^{n} + \boldsymbol{X}^{n-1}}{\triangle t^2}\right)\cdot \boldsymbol{\tau}.
\end{aligned}
\tag{4.55}
$$

- Solve the transport step

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \nabla^2\boldsymbol{u} + F_{\boldsymbol{\tau}}^{[0],\,n+1}, \tag{4.56}$$

where the singular force $F_{\boldsymbol{\tau}}^{[0],\,n+1}$ is spread to the fluid from the membrane by means of the immersed boundary method, namely,

$$F_{\boldsymbol{\tau}}^{[0],\,n+1} = \sum_k f_{\boldsymbol{\tau}}^{[0],\,n+1} D(\boldsymbol{X}_k)\Delta s.$$

Here $D(\boldsymbol{X}_k)$ is the discrete delta function.

- Use the IIM to solve the projection step

$$\nabla^2 p = \nabla\cdot\boldsymbol{u}^*, \tag{4.57}$$

with the jump condition $[\![p]\!] = f_{\boldsymbol{n}}^{[0],\,n+1}$ incorporated into the difference equation. $\boldsymbol{u}^*$ is the result from the transport step.

- Interpolate $\boldsymbol{U}^{[0],\,n+1}$ to the location $\boldsymbol{X}^{[0],\,n+1}$ to get $\mathcal{U}(\boldsymbol{X}^{[0],\,n+1})$. Let the initial guess of the Hessian matrix for $t=0$ to be an identity matrix, *i.e.* $H^{[0],\,1} = \mathrm{I}$, and set the initial guess of the Hessian matrix for each time step to be the one from the previous time step. *i.e.* $H^{[0],\,n+1} = H^n$.

Denote $g^{[k],\,n+1} = g(\boldsymbol{X}^{[k],\,n+1})$ and proceed through the iterative steps of a quasi-Newton method as follows:
For k=0, 1, 2, ... obtain $\boldsymbol{X}^{[k+1],\,n+1}$, $H^{[k+1],\,n+1}$ *from* $\boldsymbol{X}^{(k),\,n+1}$, $H^{(k),\,n+1}$

1. If $\| g^{[k],\,n+1} \| \leq \varepsilon$ then $\boldsymbol{X}^{[k],\,n+1} = \boldsymbol{X}^{n+1}$, $H^{n+1} = H^{[k],\,n+1}$, and stop, otherwise
2. Compute $\boldsymbol{s}^{[k],\,n+1} = H^{[k],\,n+1} g^{[k],\,n+1}$.
3. Determine $\boldsymbol{X}^{[k+1],\,n+1} = \boldsymbol{X}^{[k],\,n+1} - \lambda^k \mathbf{s}^{[k],\,n+1}$, where in practice the constant step length $\lambda^k = 1$ is chosen.
4. Compute the force density $f_{\boldsymbol{n}}^{[k+1],\,n+1}$ and $f_{\boldsymbol{\tau}}^{[k+1],\,n+1}$ as in (4.54) or (4.55).
5. Solve the transport and projection step as described in (4.56) and (4.57) to obtain $\mathcal{U}(\boldsymbol{X}^{[k+1],\,n+1})$. Set $\boldsymbol{p}^k = \boldsymbol{x}^{[k+1],\,n+1} - \boldsymbol{x}^{[k],\,n+1}$, $\boldsymbol{q}^k = g^{[k],\,n+1} - g^{[k+1],\,n+1}$.

6. Choose suitable values of γ^k and θ^k to compute the next Hessian matrix, $H^{[k+1],n+1} = \Psi(\gamma^k, \theta^k, H^{[k],n+1}, \boldsymbol{p}^k, \boldsymbol{q}^k)$,

where

$$\begin{aligned}\Psi(\gamma,\, \theta,\, H,\, \boldsymbol{p},\, \boldsymbol{q}) =& \gamma H + \left(1 + \gamma\,\theta \frac{\boldsymbol{p}^T H \boldsymbol{p}}{\boldsymbol{p}^T \boldsymbol{q}}\right) \frac{\boldsymbol{p}\boldsymbol{p}^T}{\boldsymbol{p}^T \boldsymbol{q}} - \gamma \frac{(1-\theta)}{\boldsymbol{q}^T H \boldsymbol{q}} H \boldsymbol{q} \cdot \boldsymbol{q}^T H \\ &- \frac{\gamma\,\theta}{\boldsymbol{p}^T\,\boldsymbol{q}} \big(\boldsymbol{p}\,\boldsymbol{q}^T\, H + H\,\boldsymbol{q}\,\boldsymbol{p}^T\big).\end{aligned} \tag{4.58}$$

end for

In (4.58), different choice of γ and θ leads to different method. Two common choices are

1. $\gamma = 1,\ \theta = 1$: the rank two method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS method).

2. $\gamma = 1,\ \theta = \boldsymbol{p}^T\boldsymbol{q}/\big(\boldsymbol{p}^T\boldsymbol{q} - \boldsymbol{q}^T H \boldsymbol{q}\big)$: the symmetric, rank-one method of Broyden (SR1 method).

In theory, the BFGS method requires that $\boldsymbol{p}^k$ and $\boldsymbol{s}^k$ satisfy the *curvature condition*

$$(\boldsymbol{p}^k)^T\, \boldsymbol{s}^k > 0. \tag{4.59}$$

However, this condition does not always hold for our function (4.52), and in this case we need to enforce (4.59) explicitly by imposing restrictions on the line search procedure, *i.e.* to choose an appropriate λ^k so that (4.59) is satisfied. Nevertheless, in practice λ^k is usually chosen to be constant one for simplicity. For our implementation, we also choose $\lambda^k = 1$ to avoid any complexities arising from choosing different λ^k. Consequently for our implementation, the BFGS method does not converge as well as the SR1 method which does not require the curvature condition (4.59).

Detailed discussions for quasi-Newton methods can be found, for example, in [74, 92].

4.7 Preservation of steady state

Our algorithm has the nice feature that certain steady states are exactly maintained. In particular, consider a circular stretched membrane enclosing some incompressible fluid, a *pressurized membrane.* This is a two-dimensional model for an inflated balloon, and is an exact steady state with $\boldsymbol{u} \equiv 0$ and a piecewise constant pressure having two values $p_{\text{in}} > p_{\text{out}}$. The pressure is larger inside the membrane and the jump in pressure equals the force exerted by the membrane, which is purely a normal force with constant magnitude around the membrane. A circular bubble with force exerted by surface tension has the same steady state.

With our IIM, this steady state is exactly maintained numerically. To see this, consider one step of the algorithm with $\boldsymbol{U}^n = 0$ and $\boldsymbol{u}^n = 0$. Since $\boldsymbol{F}_\tau = 0$, we obtain $\boldsymbol{U}^* = 0$ and $\boldsymbol{u}^* = 0$ after applying Steps 1–3. By our choice of correction terms, we find that $B^1_{i-1/2,j}$ and $B^2_{i,j-1/2}$ always take one of the values 0 or $\pm(p_{\text{in}} - p_{\text{out}})$. It can be verified that Step 4 then results in p_{ij} being piecewise constant with value p_{in} or p_{out}, so p is computed exactly at the grid points. Then the update terms in both (4.24) and (4.26) turn out to be exactly zero everywhere, since any jump in p is exactly cancelled by a correction term B, and so $\boldsymbol{u}^{n+1} = \boldsymbol{u}^* = 0$ and $\boldsymbol{U}^{n+1} = \boldsymbol{U}^* = 0$. Hence the steady state is maintained.

By contrast, the original IB method does not exactly maintain a steady state of this form. The incorporation of the normal force in the transport step leads to $\boldsymbol{u}^* \neq 0$, and using its discrete divergence on the right-hand side of the Poisson problem does not lead to a pressure that is exactly correct. While the resulting $\boldsymbol{U}^{n+1}$ may be small on a fine grid, there is typically some inward motion of the interface, a "leaking" of the fluid from the high pressure region. This is often observed with IB methods and is illustrated in some of the examples of Section 4.9. The improvement of Peskin and Printz [83] greatly reduces but does not eliminate this effect.

4.8 Relation to the blob projection method

We have briefly reviewed the blob projection method of Cortez and Minion in Section 2.3.3. It is interesting to compare our approach with their method, since they also avoid taking the divergence of discrete delta functions. The blob projection method is based on the idea that the divergence-free portion of the force $\boldsymbol{F}$ can be computed exactly and contributes to the time derivative of $\boldsymbol{u}$. This is seen by applying the projection operator $\mathbb{P}$ to the first equation of (4.1). Since $\nabla \cdot \boldsymbol{u} = 0$ at all times $\boldsymbol{u}_t$ is left alone while $\boldsymbol{F}$ is replaced by $\mathbb{P}\boldsymbol{F}$. In the blob projection method, the singular force $\boldsymbol{F}$ is discretized as a sum of discrete cutoff functions centered at points on the interface, as in the IB method, using a discrete delta function $d_h(\boldsymbol{x})$ of the form (4.20), for example. The regularized force $\mathbb{P}\boldsymbol{F}$ by the cutoff is computed analytically to give a "dipole-field" velocity. Such dipole-field velocity is nonlocal and nonzero at all grid points, and so a special fast algorithm discussed in [24] must be used to compute these sums efficiently.

There is presumably still an effect of smearing the normal force over discrete blobs, but with the better $d_h(\boldsymbol{x})$ used in [24] and fourth order accurate algorithms, very good results are reported. In particular, good volume conservation is obtained, although it does not appear that steady states of the form discussed in Section 4.7 can be exactly maintained. Each $\mathbb{P}d_h(\boldsymbol{x})$ based on the normal force gives a velocity update that is nonzero everywhere. Mathematically, integrating the velocity updates that would result from projecting an exact delta function at each point on the boundary should yield complete cancellation in the case where the interface is circular with a normal force of constant magnitude. Numerically, however, when this force is discretized by blobs and the update evaluated at discrete grid points, it is unlikely that exact cancellation will occur.

This method seems more difficult to implement efficiently than our approach. Also, the analytic form of $\mathbb{P}d_h(\boldsymbol{x})$ needed depends on the boundary conditions on $\boldsymbol{u}$ imposed on the computational domain, and so this approach may be limited to particular boundary conditions in simple geometry.

Philosophically, there are connections with our method that are worth noting. We update $\boldsymbol{u}^*$ using only the tangential component of the force. This component is more nearly divergence-free than the normal component, in the sense that a purely tangential force of constant magnitude is exactly divergence free and accelerates the fluid by dragging it along. If the magnitude of the tangential force varies then this it is not exactly divergence free, with the variation in force instead leading to variation in the pressure jump across the interface, as seen from the second jump condition in (4.4). The normal force goes entirely into the pressure jump. So by using only $\boldsymbol{F}_\tau$ in the transport step, we are essentially performing an approximate projection of the force, but in a way that maintains locality near the boundary. The errors in this projection are fixed in Steps E5 and E6, using the solution p to the nonlocal Poisson problem.

4.9 Numerical results

In this section we present some numerical results for several test problems with immersed elastic membranes.

Example 4.1. We first consider an example used by Cortez and Varela [25], who present an asymptotic solution that can be used for comparison purposes. A closed elastic membrane in an inviscid fluid is perturbed slightly from a circular steady state and allowed to oscillate. The tension is assumed to be constant along the membrane at each time t with value $T(t)$ given by some stiffness factor σ multiplied by the length of the membrane, which we compute using Simpson's rule on the cubic spline through the control points $\boldsymbol{X}_k(t)$. We take $\sigma = 1$. The force at any point s on the membrane is then specified to be of the form

$$\boldsymbol{f}(s,t) = T(t)\kappa(s,t)\boldsymbol{n}(s,t), \tag{4.60}$$

where $\boldsymbol{n}(s,t)$ is the unit normal and $\kappa(s,t)$ is the curvature. Note that since the force is in the normal direction (due to the fact that the fluid is inviscid and can slip in the tangential direction), we do not need to use discrete delta functions and the force is imposed entirely through the jump in pressure in the IIM algorithm.

The initial shape of the membrane is given by

$$r_0(\theta,\, t_0) = (1-\varepsilon^2/2)^{1/2} + \varepsilon\,\cos(k\theta), \tag{4.61}$$

which specifies the radius of the membrane at each angle θ. For $\varepsilon = 0$ this is just circle. We take $k = 2$ and $\varepsilon = 0.1$, and use N_b marker points $\boldsymbol{X}_k(t)$ that are initially uniformly spaced on the membrane.

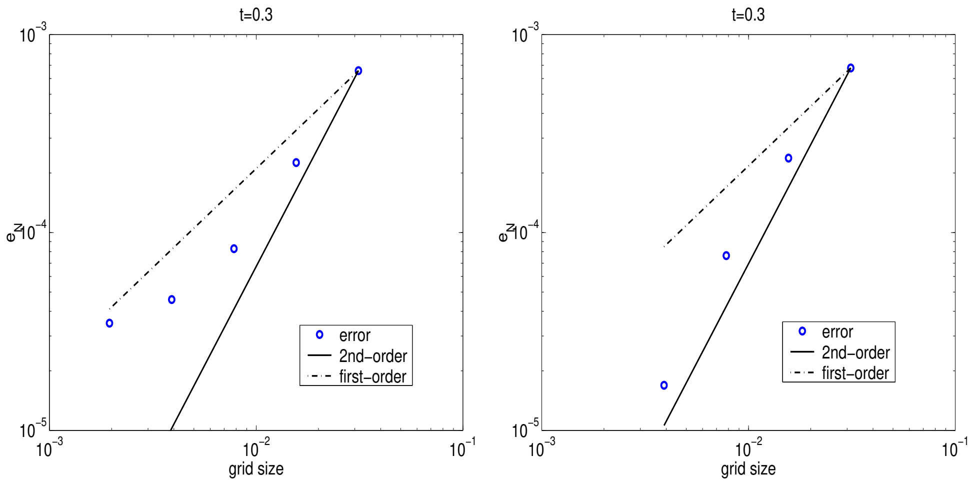
The asymptotic solution derived in [25] for small ε is then given by

$$r(\theta,\, t) = \left[1 - \frac{\varepsilon^2}{4}A^2(t)\right] + \varepsilon A(t)\cos(k\theta) + \varepsilon^2 B(t)\cos(2k\theta), \tag{4.62}$$

where the amplitudes $A(t)$ and $B(t)$ are given by (for $k = 2$)

$$A(t) \;=\; \cos\left(\sqrt{6}\pi\left(1 - \frac{223}{480}\varepsilon^2\right)t\right) \tag{4.63}$$

$$B(t) \;=\; \frac{3}{20} + \frac{1}{3}\cos(2\sqrt{6}\pi\, t) - \frac{29}{60}\cos(\sqrt{60}\pi\, t). \tag{4.64}$$



(a) (b)

Figure 4.2: (a) The error in the norm (4.65) at time $t = 0.3$, relative to the asymptotic solution (4.62). (b) The error relative to a reference solution on a 512×512 mesh.

To compare our computed solution to this, we compute $r_k(t)$ and $\theta_k(t)$ from the marker locations $\boldsymbol{X}_k(t)$ and measure the 2-norm of the error in radial position as

$$\frac{1}{N_b}\sqrt{\sum_{k=1}^{N_b}(r_k(t) - r(\theta_k(t), t))^2}. \tag{4.65}$$

The asymptotic solution is only accurate up to $O(\varepsilon^3) \sim 10^{-3}$ and so we do not expect the numerical solution to converge to the asymptotic solution as the grid is refined. Fig. 4.2(a) shows the error (relative to (4.62)) in this norm as the grid is refined, which illustrates good convergence on coarser grids but that the asymptotic solution evidently has an error of approximately 2×10^{-5} relative to the true solution. Fig. 4.2(b) shows the error in this same norm if the numerical solution on the finest (512×512) grid is used as a reference solution. Here we see nearly second order accuracy on all coarser grids.

We also compare the amplitudes $A(t)$ and $B(t)$ from (4.63) and (4.64) of the dominant Fourier modes against the computed results, obtained by applying a discrete Fourier transform to the computed function $r(\theta, t)$ determined by the marker points. These comparisons are shown in Figure 4.3. We see that the computed Fourier modes have good agreement with both $A(t)$ and $B(t)$, somewhat better than the results shown by Cortez and Varela [25]. They use an impulse method with a projected force $\mathbb{P}\boldsymbol{F}$ analogous to that used in the blob projection method. (Cortez [22] reports that better accuracy than shown in [25] has been obtained with better discrete delta functions.)

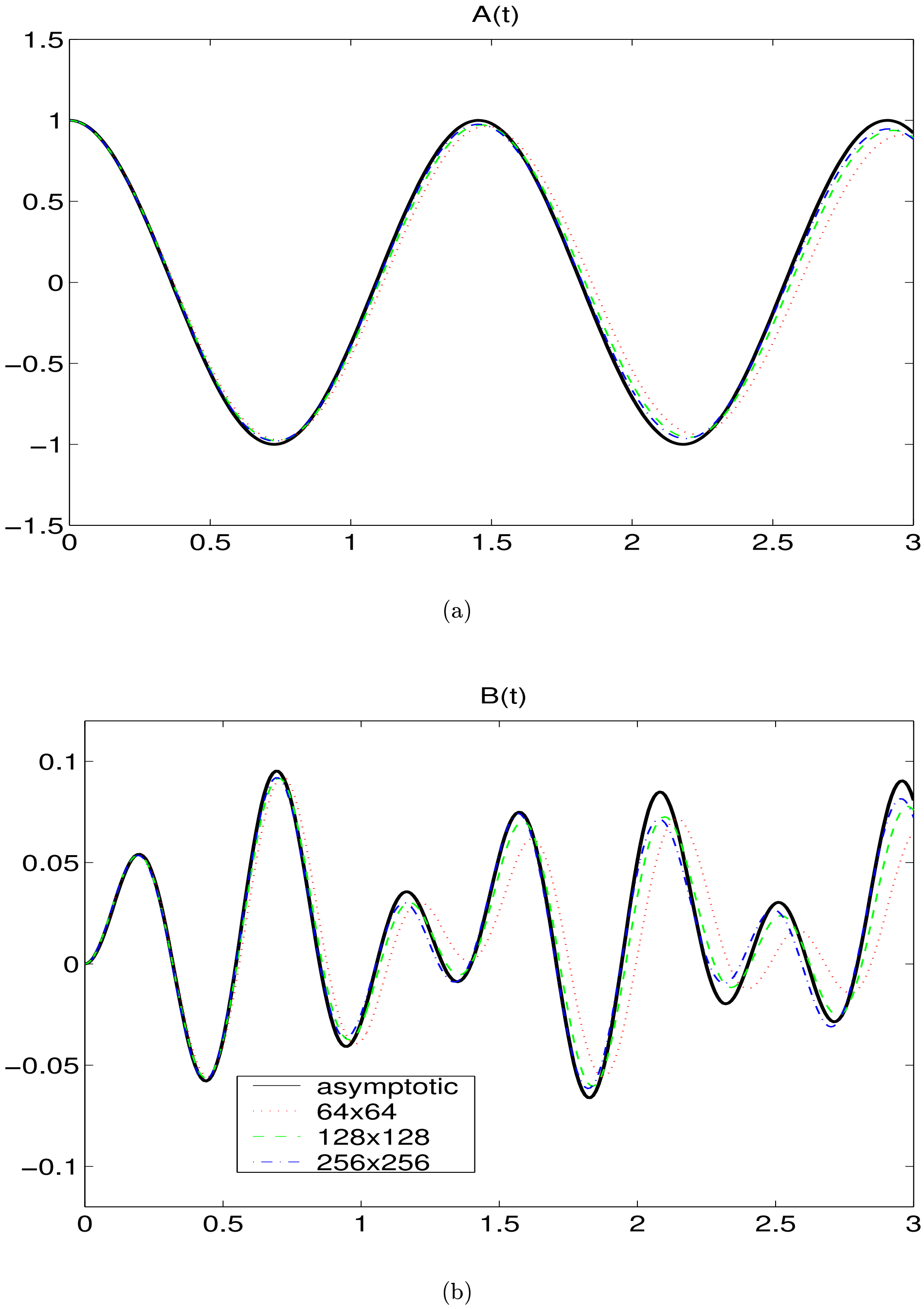


Figure 4.3: Comparison of the Fourier modes (4.63) and (4.64) from the asymptotic solution with those computed using the IIM.

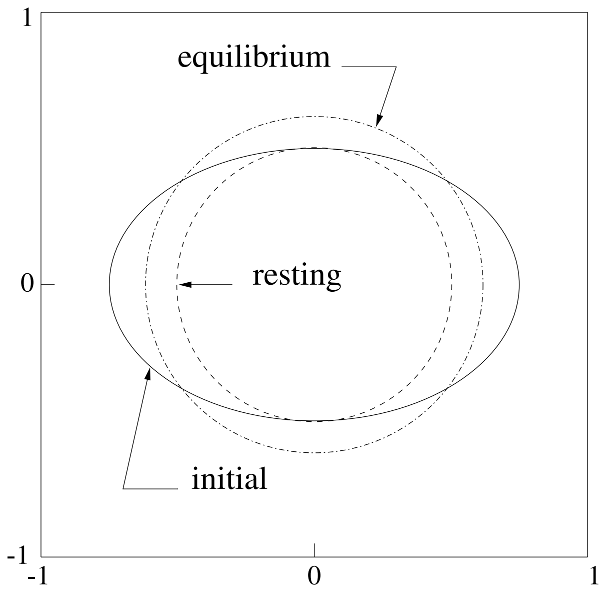


Figure 4.4: The interface at different states.

Example 4.2. The previous example was for an inviscid fluid with a simplified form of the interface force. In this example, we consider a stretched pressurized membrane immersed in a viscous fluid, with force density given by (4.9). We assume that the resting unstretched membrane is a circular elastic curve with radius $r_0 = 0.5$. The initial shape of the boundary is an ellipse with major and minor axes $a = 0.75$, $b = 0.5$ respectively. Due to the restoring force of the elastic boundary and the incompressible fluid inside the membrane, the ellipse should converge to an equilibrium circular steady state of the form discussed in Section 4.7, with radius $r_e = \sqrt{ab} \approx 0.61237$. When the Reynolds number is small, Re $= 1$ for instance, the membrane relaxes gradually to the equilibrium state as in Stokes flow, where similar examples have been used as computational tests in [64, 100]. When the Reynolds number is large, Re ≥ 100, the membrane will oscillate as it converges to the equilibrium state. Fig 4.4 shows three different states of the interface for Re $= 100$, the case considered in our computations. We solve on the domain $[-1, 1] \times [-1, 1]$ with $u = 0$ on all four boundaries, a closed box.

Fig. 4.5 shows nearly one cycle of the oscillation as computed on a 64×64 grid with $N_b = 64$. The solid line is computed by our IIM and the dotted one is computed using our own version of the original immersed boundary method. They are almost indistinguishable in a short time period, even though the grid is coarse. For the immersed boundary (IB) method, we again use the projection algorithm described above but with $\boldsymbol{F}_\tau$ replaced by the full force in (4.19), and with no correction terms in the projection step, i.e., $B^1_{i-1/2,j} = B^2_{i,j-1/2} = C_{ij} = 0$. We have not implemented the modified projection method introduced by Peskin and Printz [83], which improves the volume conservation of the original immersed boundary method (see below) but is more complicated.

To compare results over longer times, let r_x be the radius of the interface measured along x-axis and r_y the radius of the interface measured along y-axis. These should each oscillate and converge towards the common value $r_e = \sqrt{ab} \approx 0.61237$ as the oscillation is damped by viscosity and the membrane becomes more circular. We now use a 256×256 grid with $N_b = 256$. Fig 4.6 shows the evolution of r_x and r_y computed by both IIM and IB method. The solid line is from the IIM. It clearly shows that both r_x and r_y converge to r_e even when we blow up the figure near the true equilibrium position as shown in Fig. 4.6(b). By contrast, the dotted line computed from the IB method indicates that the circle shrinks linearly. The leaking of fluid is well-known for the original IB method [64, 83].

Since the area enclosed by the membrane should be exactly conserved, we can use the area based on the computed membrane position as one measure of the accuracy of the method. Fig. 4.7 shows that area is conserved with second-order accuracy using our method, and only to first order with our version of the IB method. Similar results were seen for Stokes flow in [64].

Figures 4.8 and 4.9 show slices of the pressure computed at two different times with the IIM and IB methods, respectively. These show that the two methods produce consistent results but that the pressure jump is more sharply captured with the IIM.

Figure 4.10 shows slices of the computed velocity with each method, with a grid refinement study. We see that results with the IIM are more accurate and smoother, particularly on the coarser grids.

Before moving on to the next example, we want to address an issue about the tangential force. The oscillation of membrane in this example is driven by the pressure difference across the interface. The force is mainly in the normal direction. Based on our calculation, the tangential force is about 1% of the normal force. However, even the tangential force is only a fraction of the normal force, it cannot be ignored. Figure 4.11 shows the incorrect numerical solutions computed by the algorithm, if the tiny tangential force is ignored.

Example 4.3. In this example we perform a similar calculation to Example 4.2. The results are compared with those from the explicit method. The main purpose of this test is to verify our implementation for the implicit algorithm. In addition, it also demonstrates the capability of using large time steps for the implicit method.

The boundary condition of the computational domain is doubly periodic. 128×128 mesh with markers $N_b = 128$ is used for the computation. Fig (4.12) (a) is computed by the explicit method with $\triangle t = 0.0075$. It shows a wriggling movement. The movement is the evidence of the instability of the explicit method for stiff problems. The CFL number is ≈ 0.547 at $t = 0.0525$. This relatively large CFL number is due to the instability, since when we reduce the time step to $\triangle t = 0.0025$, the method becomes stable and the CFL number is only 0.006 at $t = 0.00525$. Fig (4.12) (b) is the plot of the position of the interface at $t = 1$. The solid line is computed by the explicit method and the dotted one is by the implicit method. They are simply identical. The time step for the explicit method is $\triangle t = 0.0025$ and $\triangle t = 0.01$ for the implicit method. The inverse Hessian matrix in this example is approximated by the BFGS method. The CFL number is 0.03 for the explicit method and 0.117 for the implicit method at time $t = 1$. We expect that the CFL number to be ≤ 1 for the implicit method when we take $\triangle t = 0.05$.

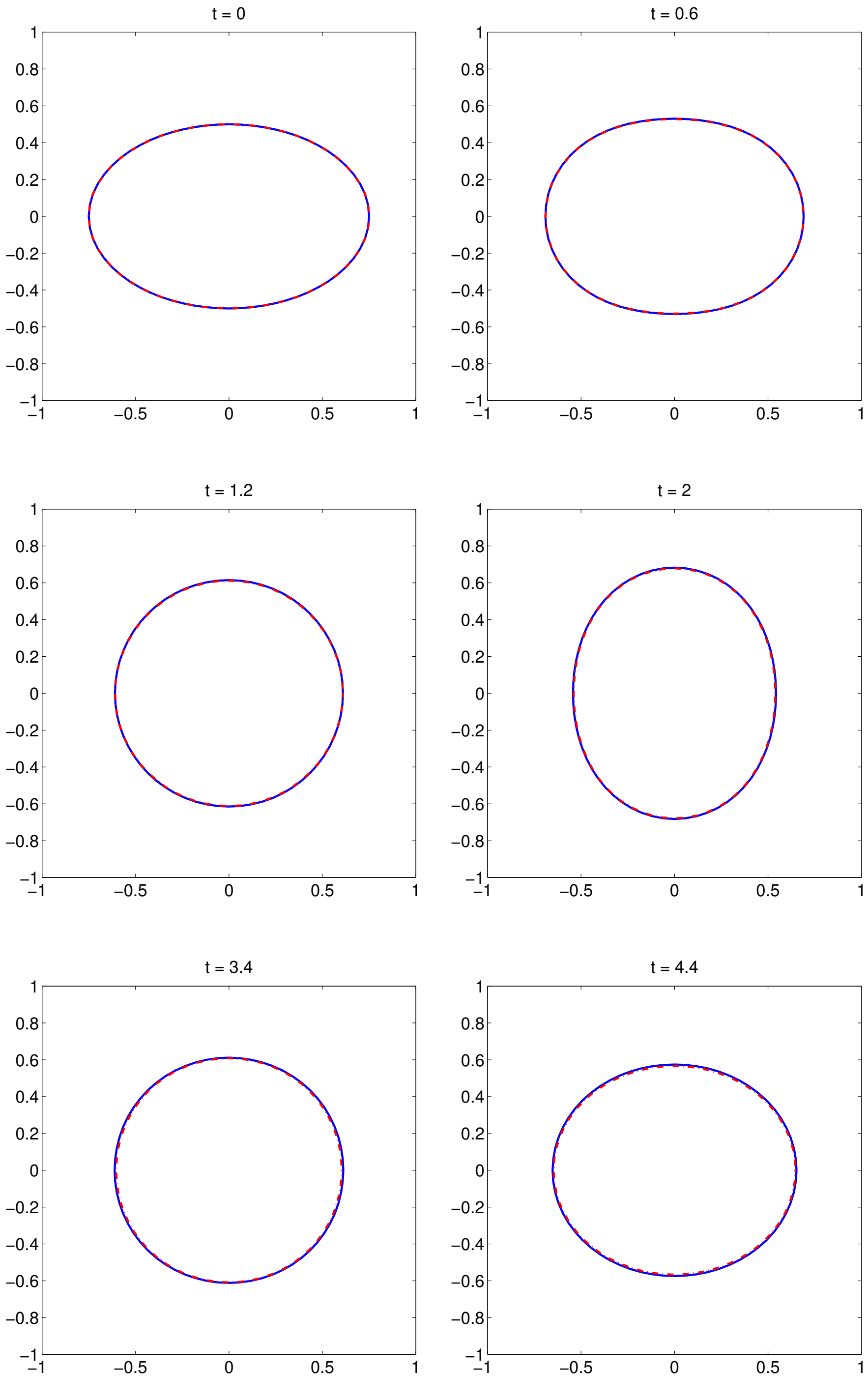
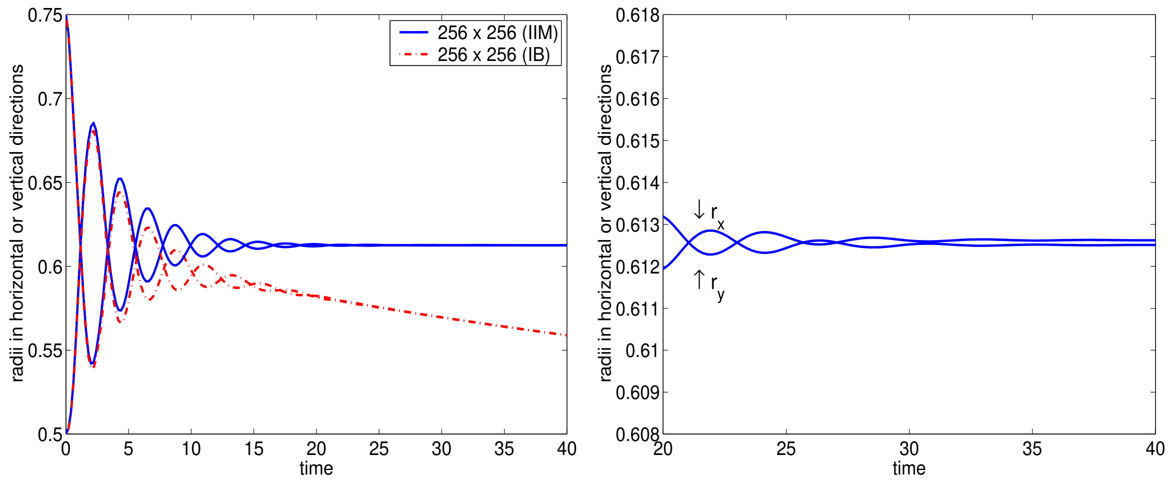


Figure 4.5: Nearly one cycle of the oscillation of the pressurized membrane in a box. For each frame, the solid line is the result from IIM and the dotted line is the one from IB method. 64×64 mesh with $N_b = 64$ is used



(a)

(b)

Figure 4.6: (a) The evolution of r_x and r_y. Solid line is from IIM. Dotted line is from IB method. (b) Blow-up of IIM results around the equilibrium. It shows that the interface converges to a circle with $r_e \approx 0.61237$

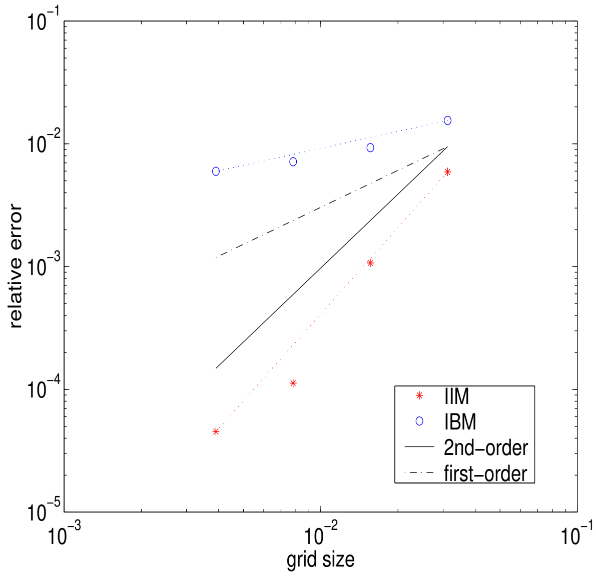


Figure 4.7: Grid refinement for the relative error of volume conservation at $t = 1$.

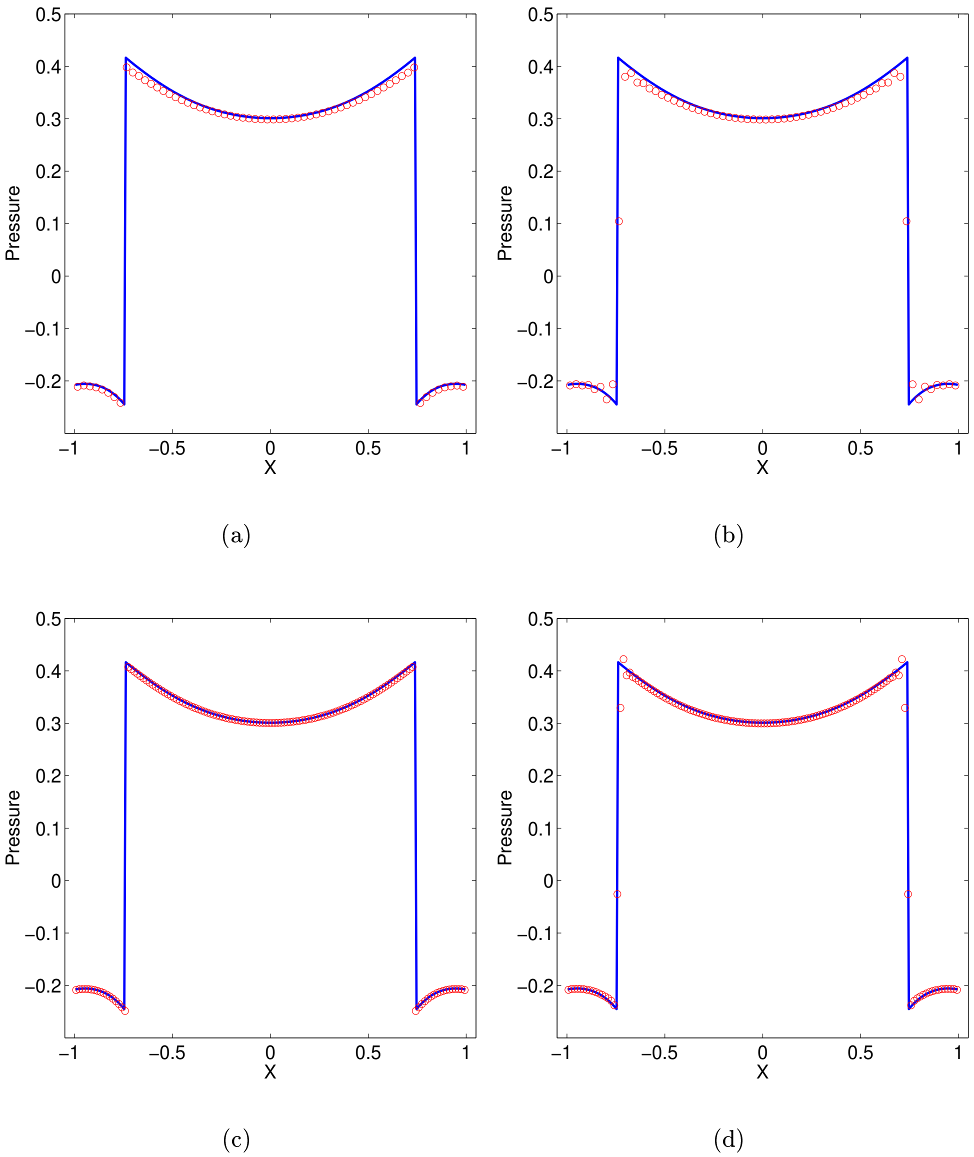


Figure 4.8: The pressure profile at $t = 0.2$. The slices of data are taken near $y = 0$. For all plots, the solid line is the result with IIM using 256×256 grid. In (a) and (b), the circles are results from IIM and IB Method using 64×64 grid respectively. (c) and (d) are results using 128×128 grid. We can see the smearing near the interface for results using IB Method, *i.e.* plots (b) and (d).

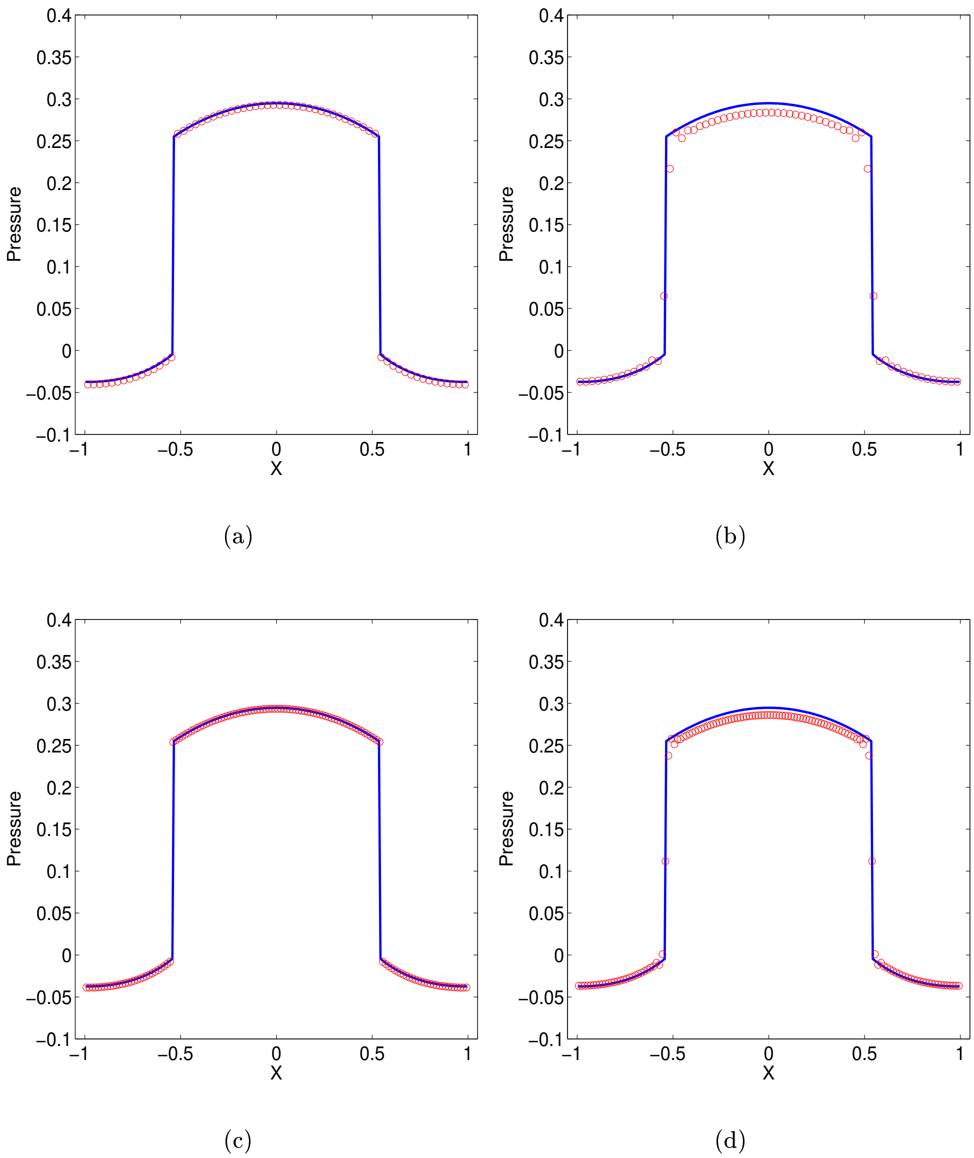
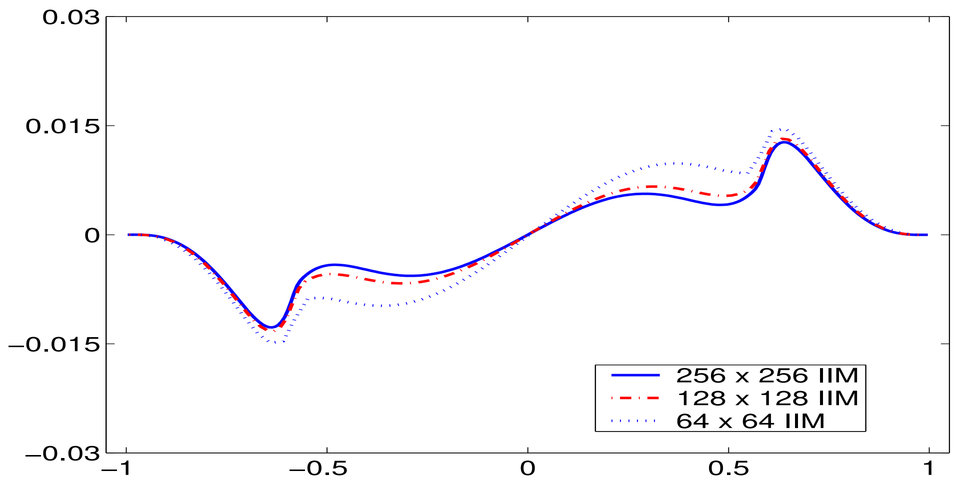
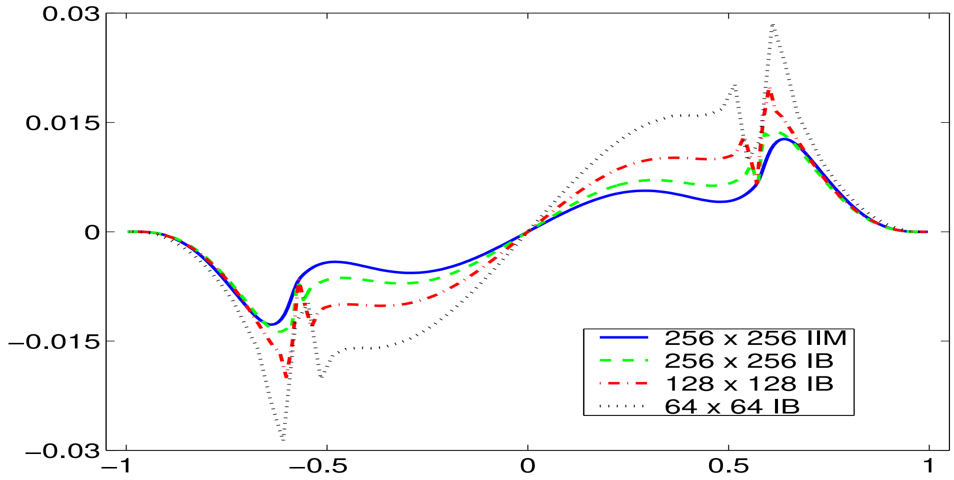


Figure 4.9: The same calculation as Fig 4.8 at later time $t = 2.2$. IB Method delivers weaker pressure inside the membrane. IIM gives sharp resolution for coarse grids.



(a)



(b)

Figure 4.10: (a) The $u-$ component velocity profile for the IIM at $y = -0.2656$ and $t = 4.2$, computed at three grid resolutions. (b) The same computation as in (a) using the IB method, along with the finest grid IIM result. The number of markers N_b used for each $N \times N$ mesh grid is $N_b = N$.

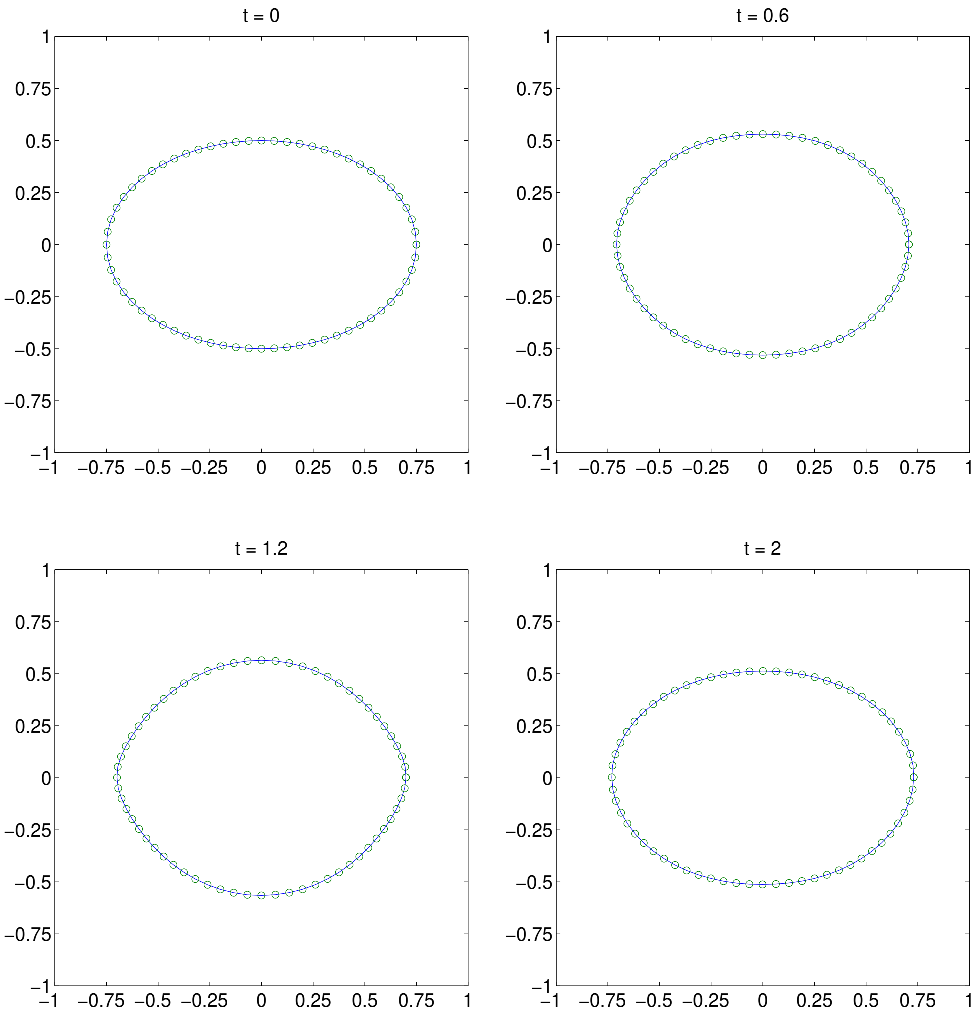


Figure 4.11: If the small tangential force in this example is ignored, the algorithm produces incorrect solutions compared with Figure 4.5. The tangential force is about 1% of the normal force.

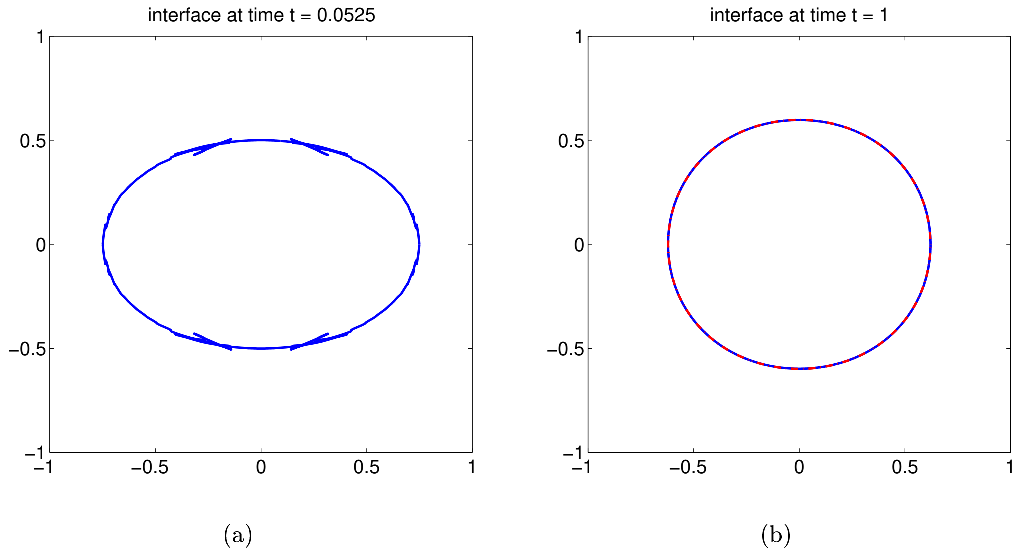


Figure 4.12: (a) An example of the instability of the explicit scheme with large time step. The time step is $\triangle t = 0.0075$. The interface starts having wriggling movement at $t = 0.0525$. 128×128 mesh grid is used. (b) The position of the interface at $t = 1$. The solid line is computed by the explicit method with $\triangle t = 0.0025$. The dotted line is computed by the implicit method with $\triangle t = 0.01$. The results are indistinguishable. The BFGS quasi-Newton method is used for the implicit scheme.

In [64], it is pointed out that in the first time step the quasi-Newton method can take too many iterations to converge. This is because the initial Hessian matrix is an identity matrix, an $O(\triangle t)$ approximation to the Jacobian matrix of $g(\boldsymbol{X})$. A good strategy for running the implicit IIM is to use a smaller time step initially and switch to the large time step after the convergence of the first time step. In this example, the implicit method uses a time step four times larger than that of the explicit method. The BFGS iterative method takes average $4 \sim 5$ iterations to converge for one time step. Within each time step, the implicit immersed interface method only computes the advection equation once, but solves elliptic equations for both diffusion and the projection-step in each iteration of the quasi-Newton method. It is well-known that solving the linear systems arising from the elliptic equations is the most costly part in the whole computation. So based on how many times the elliptic equations are solved in a given time period, we find that the computational cost for the implicit method is roughly the same as that for the explicit method.

Example 4.4. There are no existing test problems available for an immersed elastic membrane that possesses a nonzero mass density in the current literature. We present two examples of membranes with mass in this section. It is our priority to investigate the convergence and accuracy of this method in the future.

In this example, we consider the test problem similar to Example 2 but now the membrane has mass. Because of the effect of the viscosity, if we follow one particular marker on the interface and plot the radius, measured between the marker and the center of the interface, versus time, we can see a damped oscillation that is similar to the underdamping case in a damped mechanical system. In the underdamping case, the envelope of the oscillation depends on $\mathrm{e}^{-\alpha t}$, where $\alpha \propto \frac{1}{m}$ is the real part of the roots of the corresponding characteristic equation. The frequency of the oscillation $\omega \propto \frac{1}{\sqrt{m}}$, where m is the mass of the moving body. These imply that a heavy membrane has a large inertia and the membrane should oscillate in a low frequency. The envelope of the oscillation is expected to decay slowly. Figure 4.13 shows the damped oscillations of r_x and r_y for three different mass densities. r_x is the radius of the interface measured along x-axis and r_y is the radius of the interface measured along y-axis. The membrane with mass density $m(s) = 0.85$ has the longest period and a relatively slowly decaying rate. The massless membrane only experiences a few cycles of oscillation before it decays to a state near the equilibrium. Figure 4.13 shows the result using a 128×128 grid. The result computed from a finer grid is similar to Figure 4.13. Before we move on to the next example, we use a mesh refinement study to show convergence of this method. Figure 4.14 shows that the membrane position for three different grids is consistent and has good agreement.

Example 4.5. In [108], the experiment is designed to have one end of a silk thread pinned to a flowing soap film. For the computer simulation, this can be achieved by assuming that one marker has zero velocity at all time. In other words, when we solve the nonlinear equation (4.52) we assume that the jth marker has the property:

1. zero velocity at all time, or $U_j^{[k+1],n+1} = 0, \forall\, k,\, n,$
2. same position at all time, or $X_j^{[k+1],n+1} = X_j^0$ and $Y_j^{[k+1],n+1} = Y_j^0, \forall\, k,\, n.$

The second condition is imposed through all steps of the implicit method. The first condition and the new interface positions for the rest of the markers are obtained by the quasi-Newton iterations as usual.

Figure 4.15 is an example of a contracting elastic membrane with one fixed end point and the other a free end. The initial state is that the stretched curve has a maximum tension near the center of the curve and the tension decreases gradually towards the ends. Figure 4.15 (a) shows that the membrane contracts towards the center initially. Because the left end of the membrane is fixed, at later times the right end contracts towards the left end. We assume that the membrane is a straight line and hence the tension is only in the x-direction. Each marker has mass $m_k = 2$ and Reynolds number is $Re = 1$.

Example 4.6. As a final example, we put a pressurized circular membrane in the center of a lid-driven cavity, with zero initial fluid velocity. When the “lid” starts moving, the viscosity will drive the fluid motion. The membrane will also be driven and distorted by the flow, although the membrane tension tends to keep the membrane circular. This problem is computed with Reynolds number $Re = 5$ and stiffness constant $T_0 = 1$. The lid is moving from left to right with speed 1 and the other three walls are fixed with no-slip boundary conditions.

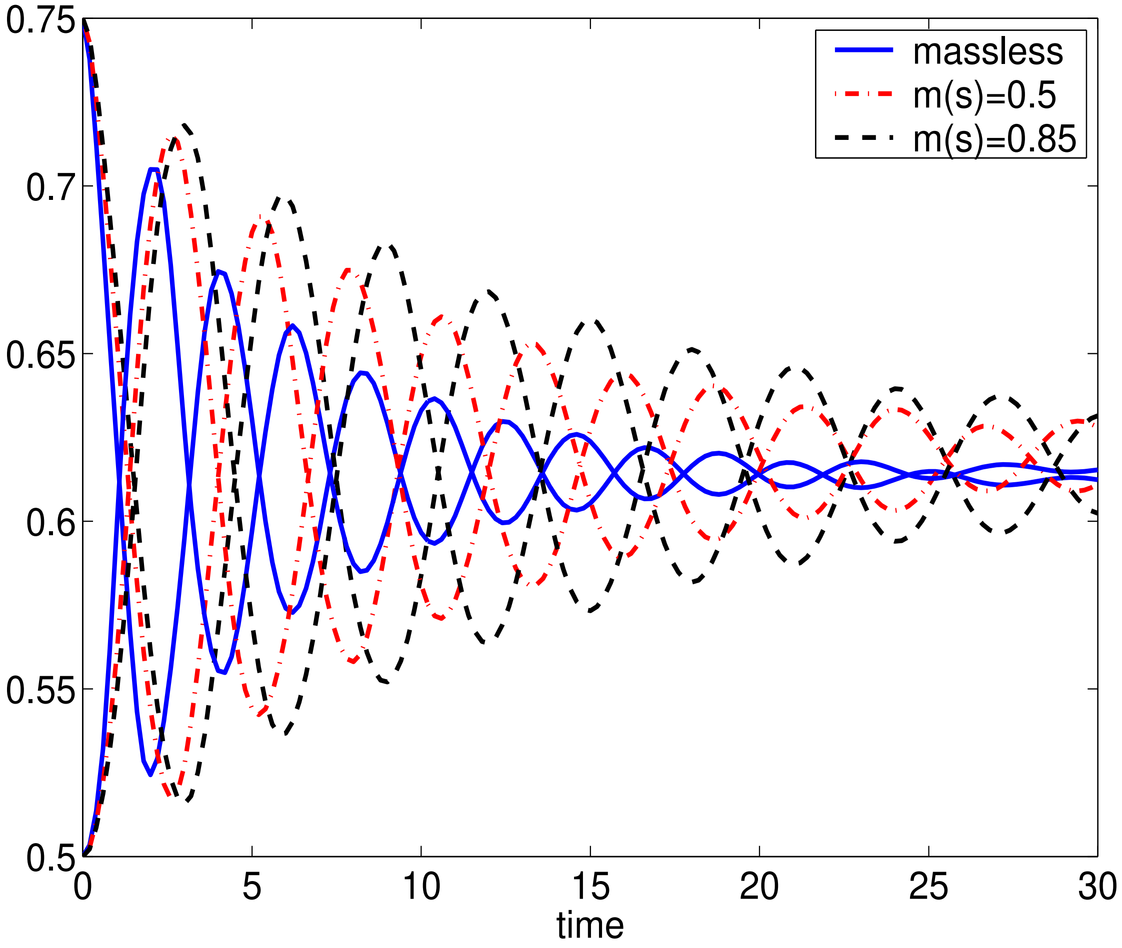


Figure 4.13: The damped oscillations for the membrane with different mass densities. 128×128 mesh with $N_b = 128$ is used. The oscillation is the evolution of r_x and r_y.

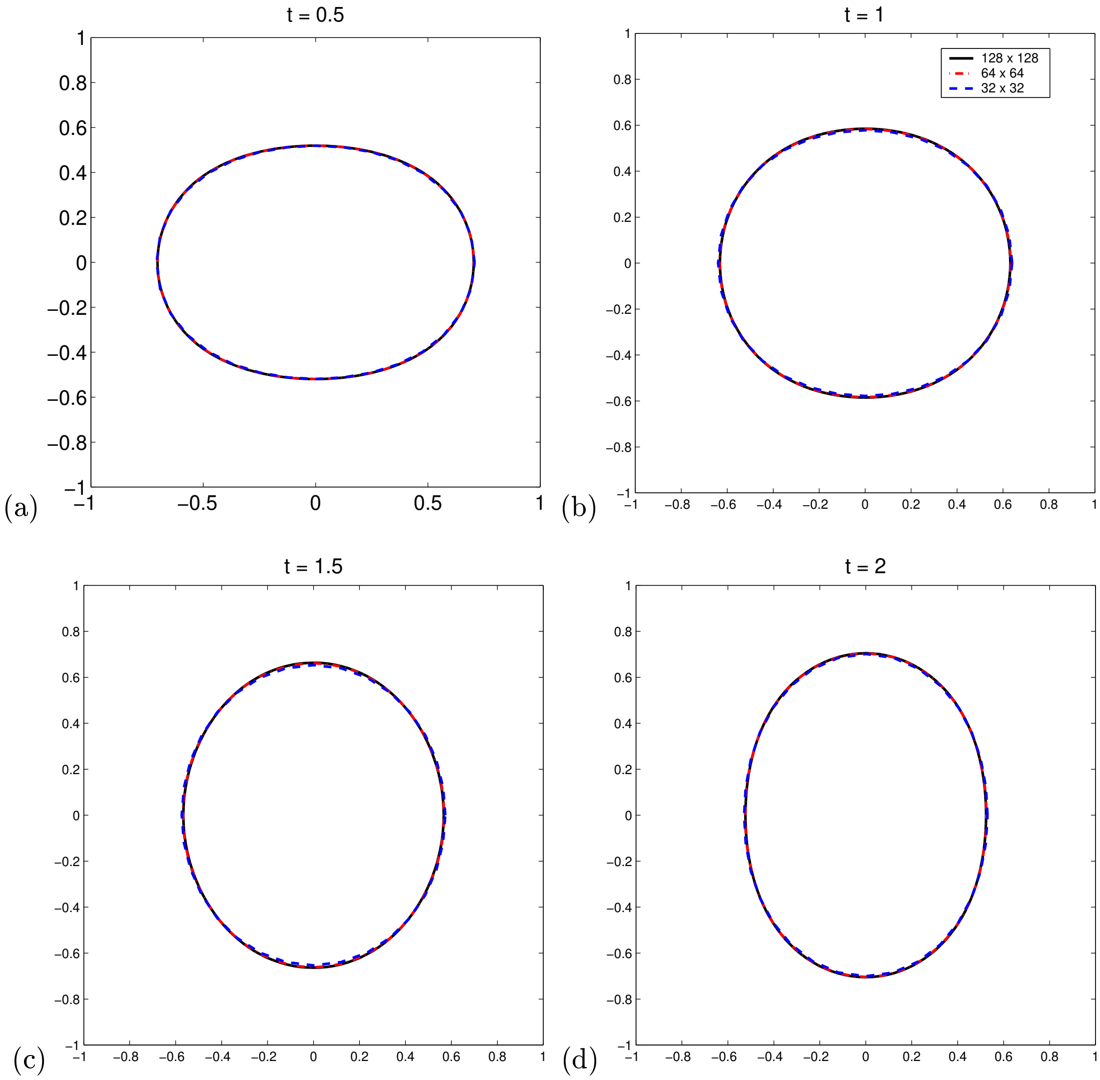


Figure 4.14: Results for the membrane with a constant mass density on three different grids, 128×128, 64×64, and 32×32. Results at four different times are shown.

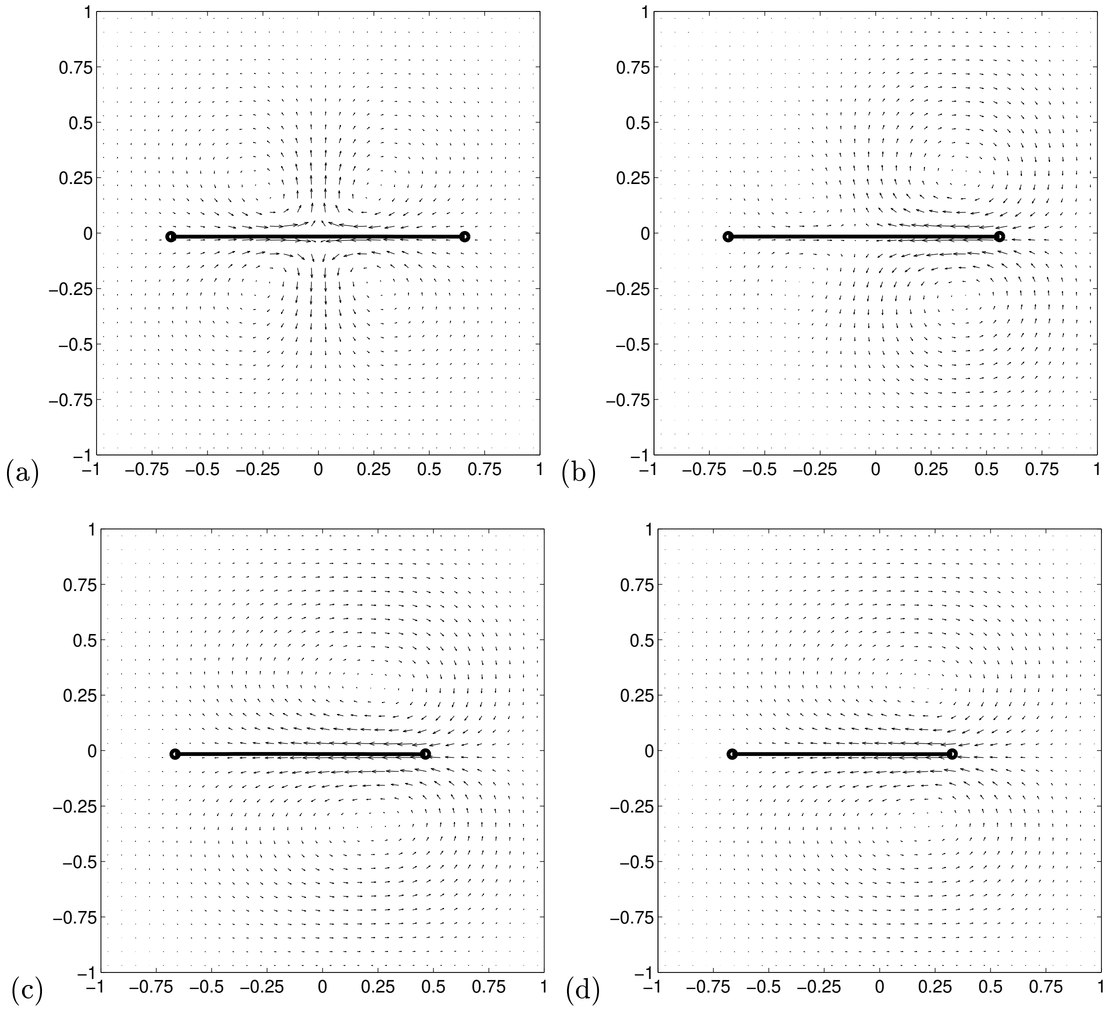


Figure 4.15: The dynamics of a contracting elastic line. From (a) to (d), the time is $t = 0.1, 0.7, 1.5$, and 5 respectively .

Figure 4.16 shows the solution at several times, as computed with the IIM on a fine grid (256×256). In order to show the fluid motion inside and outside the balloon, we also solve a scalar advection equation for some tracer that doesn't affect the flow but helps to visualize it. Three different values are used in the initial data to give the three colors. These values, representing three passive tracers, get smeared out a little bit near the sharp discontinuities by diffusion. The four small circles represent 4 markers on the membrane to indicate the rotation of the membrane.

Figure 4.17 shows the interface location at several times, as computed with each method on two different grids, 128×128 and 256×256. The IIM result on the finer grid is also displayed with the IB method results to make comparison easier. They compare well at early times but at later times there is more discrepancy. The area enclosed by the membrane is conserved with the IIM, while some shrinkage is apparent with the IB method.

We have shown a case at low Reynolds number, but our method also works well at higher Reynolds numbers. We have also experimented with different membrane stiffness factors and radii. Varying these parameters influences the dynamics and particularly the extent to which membrane is deformed as it moves with the fluid. Figure 4.18 is the case when the Reynolds number is 100. The stiffness constant and the radius remain the same as before. Because fluid at high Reynolds number is less viscous and has a weaker driving force than fluid at low Reynolds number, the balloon moves slower than the one seen in Figure 4.16 and it maintains fairly circular shape at all time.

We now let the Reynolds number be 5 again, but raise the stiffness constant to 20. Although the fluid has the same driving force as that in Figure 4.16, the membrane has a stronger resistance to the force and is deformed to a smaller extent than that in Figure 4.16. Figure 4.19 shows that the balloon is moving as fast as that in Figure 4.16, but maintains good circular shape.

Finally, we reduce the radius to 0.3 and keep the Reynolds number to be 5 and the stiffness constant to be 1. Figure 4.20 shows that the balloon moves faster than the one seen in Figure, which has radius 0.5. The balloon in this example is also deformed to a smaller extent than that in Figure 4.16.

4.10 Conclusions

We have proposed a relatively simple modification to the classic immersed boundary method, and shown that it gives better resolution and second-order accuracy for some test problems. The proper jump in pressure is easily built into the elliptic equation that must be solved in the projection step. This avoids the need to introduce unphysical discrete dipoles in intermediate steps. We have implemented this along with our version of the projection method, which is based on the high-resolution finite volume methods of CLAWPACK, but the ideas could also be applied to other projection methods as well. Our comparisons with the IB method are based on using our own code also for that method, so that the only difference in the two algorithms is the manner in which the normal component of force and jumps in pressure are handled. We believe that this illustrates the advantages of our suggested approach.

We should note that our approach is applicable only to problems where there is a sharp

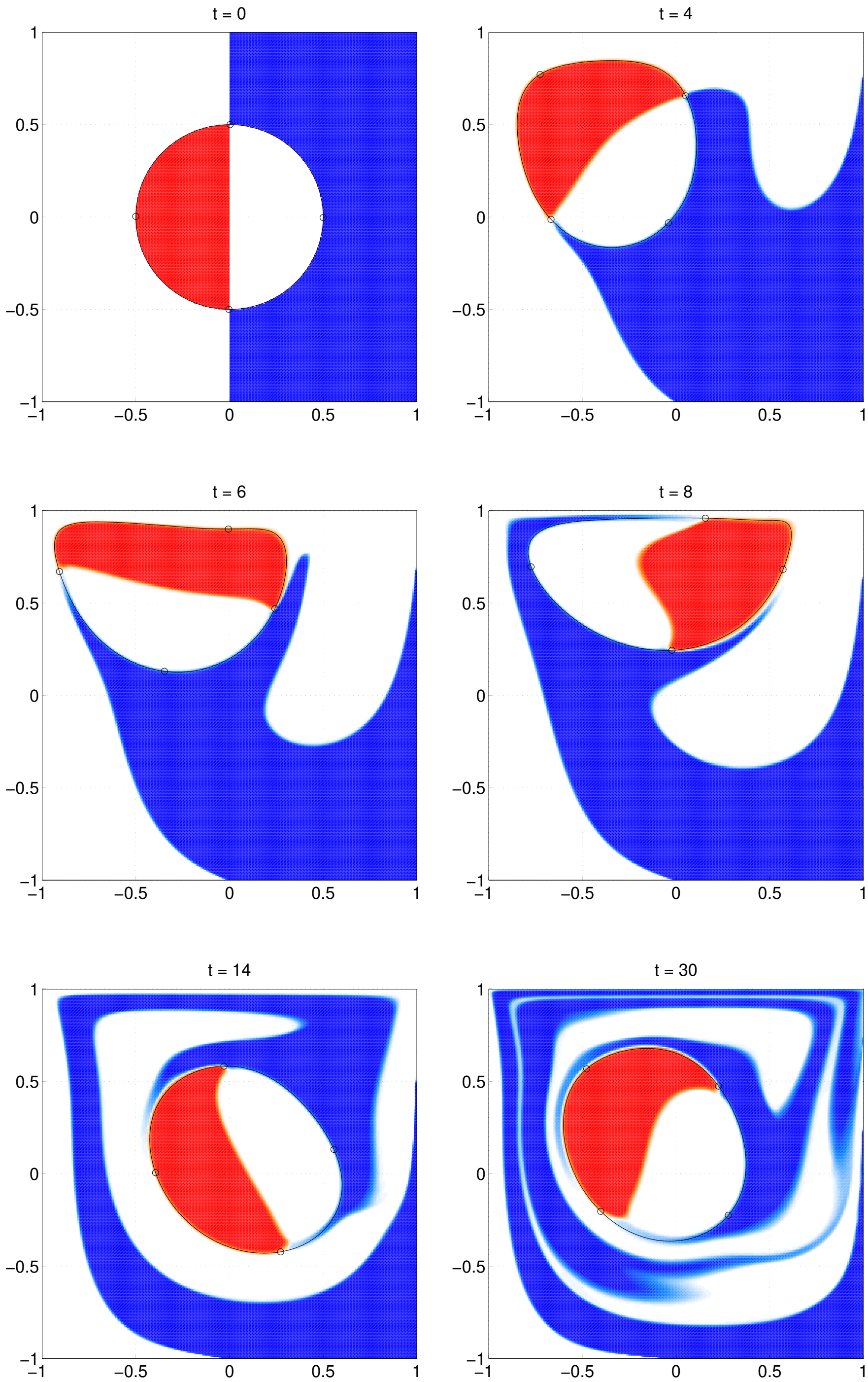


Figure 4.16: Stretched membrane in a lid-driven cavity. The Reynolds number is $Re = 5$ and the stiffness constant $T_0 = 1$. Radius is 0.5.

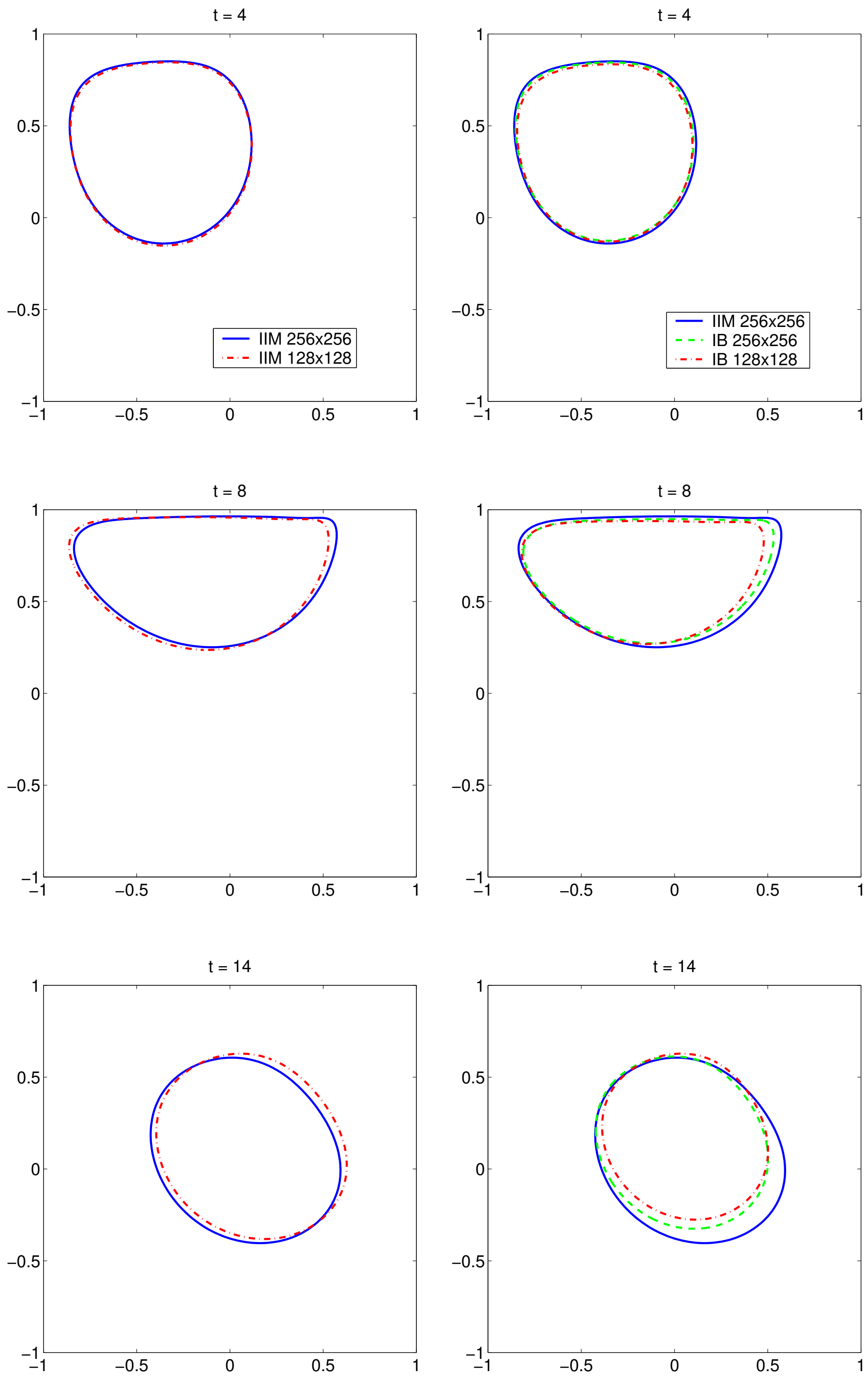


Figure 4.17: Membrane location at different times.

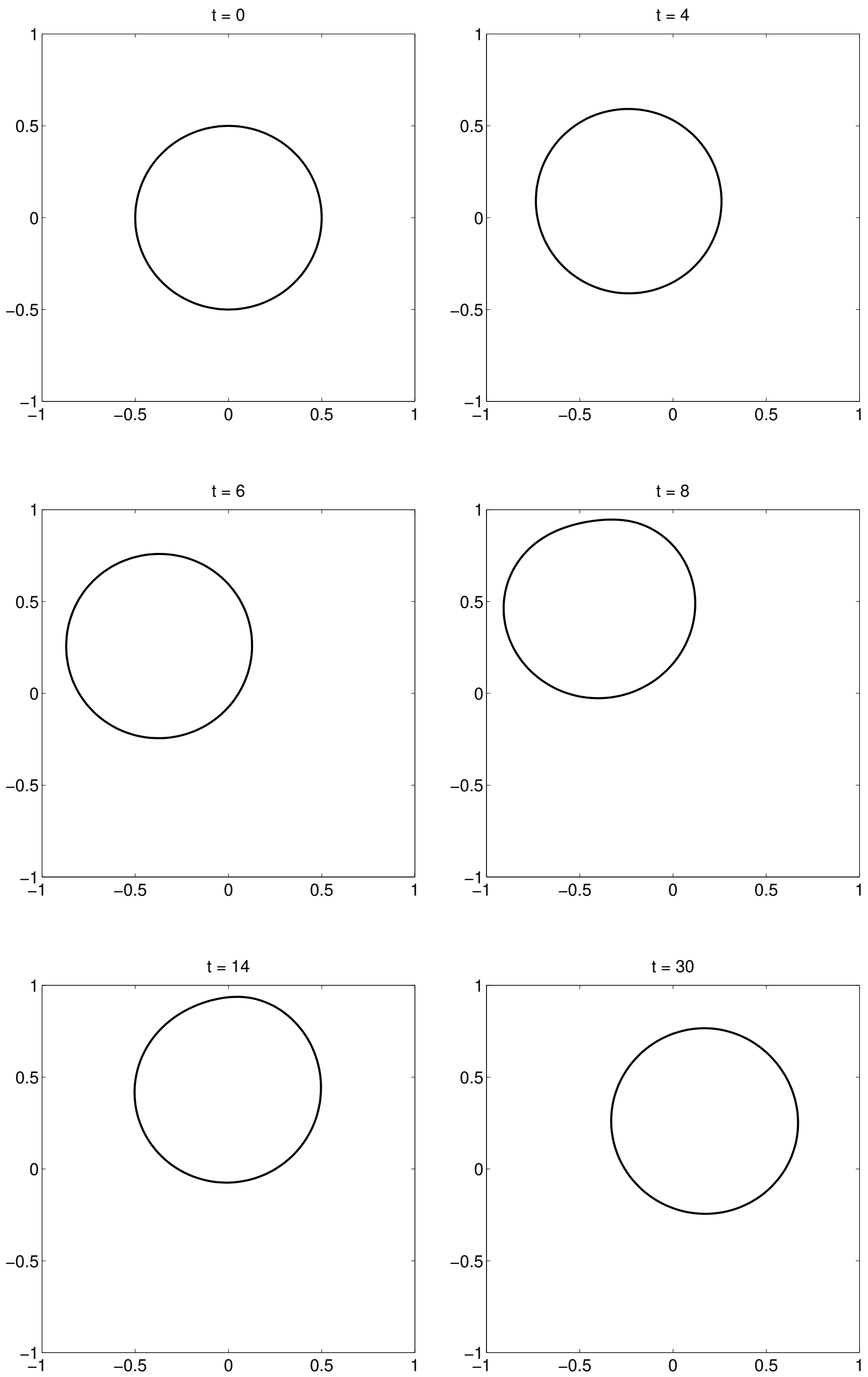


Figure 4.18: Membrane location at different times. $Re = 100$ and $T_0 = 1$. Radius is 0.5.

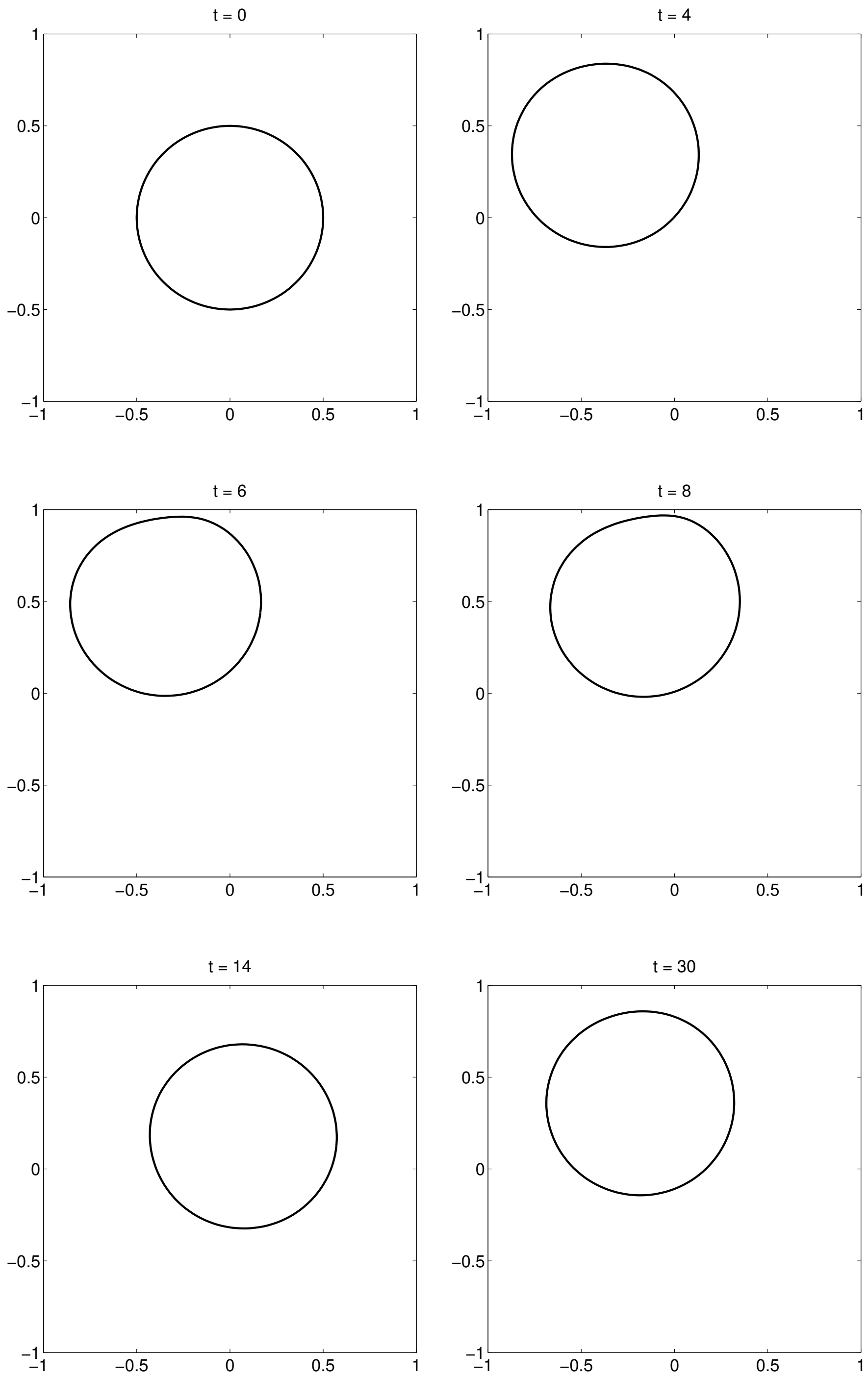


Figure 4.19: Membrane location at different times. $Re = 5$ and $T_0 = 20$. Radius is 0.5.

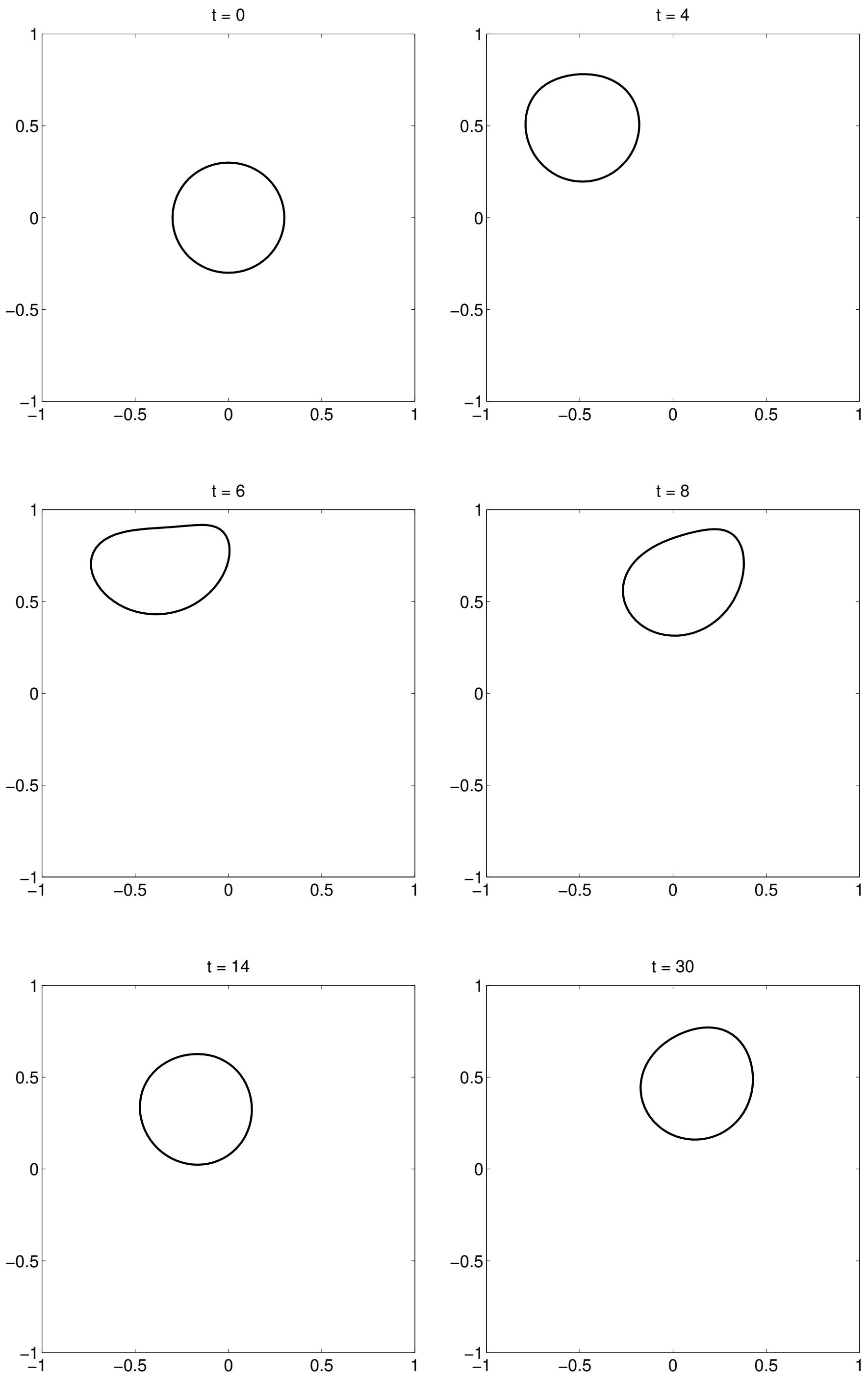


Figure 4.20: Membrane location at different times. $Re = 5$ and $T_0 = 1$. Radius is 0.3.

interface across which a jump in pressure is expected, *i.e.*, our membrane is assumed to be infinitely thin. This assumption is not valid in all applications where the IB method is used. For the three-dimensional heart model currently being studied by Peskin and McQueen [81], for example, the heart wall is not a sharp interface but rather has finite width and is built up from layers of fibers wrapping around. For an application of this sort the original IB method may be best, since the force really should be spread over a finite region. However, for many applications where the IB method is used a sharp interface is assumed, and we expect that the IIM approach would give better results.

We have proposed some basic tools for developing numerical simulations for fluid problems with immersed interfaces, including adding a nonzero mass density to an interface and pinning an object in a flowing channel. These are elements for us to simulate some interesting laboratory experiments, such as the one in [108]. Our ultimate goal is to understand the physics of the interface phenomena between fluid and flexible bodies via efficient and accurate numerical simulations.

In the future we hope to develop more general versions of this algorithm to handle cases where fluids with different densities and viscosities are on each side of the immersed boundary. Incorporating the pressure jump into the elliptic solve may be even more crucial in these cases.

Chapter 5

THE MATHEMATICAL BACKGROUND OF THE JUMP CONDITIONS

In this chapter we discuss the boundary conditions, or jump conditions, of an elastic membrane immersed in viscous incompressible fluid. The motivation of this chapter is to understand the mathematical background of jump conditions. Since jump conditions are the crucial elements in the implementation of our immersed interface method developed in the previous chapters, analyzing the underlying mathematical background of the jump conditions can help us to design an appropriate numerical algorithm.

5.1 The fluid model with singular forces

For the material interface problem or the free surface problem with surface tension, the interface can be replaced by a set a control points that exert singular forces to the fluid.

Using summation convection, the momentum equation with a singular force can be written in tensor notation as

$$\rho \frac{D u_i}{D t} = \frac{\partial \sigma_{ij}}{\partial x_j} + \boldsymbol{F}_i \tag{5.1}$$

where

$$\frac{D}{D t} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla \tag{5.2}$$

is the material derivative, and

$$\sigma_{ij} = -p\, \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{5.3}$$

is the fluid stress tensor.

The singular force $\boldsymbol{F}_i$ is expressed by

$$\boldsymbol{F}_i = \int_0^L f_i(s,t) \delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\, ds, \tag{5.4}$$

where f_i is the force density with respect to the measure ds, s the arc length function with $0 \leq s \leq L$, where L the length of the unstretched membrane, δ the two-dimensional Dirac delta function, and $\boldsymbol{X}$ the position of the markers on the interface.

Integrate (5.1) over a domain $\Omega(t)$ which moves with the fluid. Suppose that $\boldsymbol{n}$ is the outward normal of $\partial\Omega$, and dl is the measure for arc length on $\partial\Omega$. Then, by the incompressibility $\dfrac{D\rho}{Dt} = 0$, integrating the left hand side of (5.1) gives

$$\int_{\Omega(t)} \rho \frac{D u_i}{D t}\, d\boldsymbol{x} = \frac{d}{d t} \int_{\Omega(t)} \rho\, u_i\, d\boldsymbol{x}. \tag{5.5}$$

Using the Gauss divergence theorem, the first term on the right hand side of (5.1) is

$$\int_{\Omega(t)} \frac{\partial \sigma_{ij}}{\partial x_j}\, d\boldsymbol{x} = \int_{\partial\Omega(t)} \sigma_{ij}\, n_j\, dl. \tag{5.6}$$

Using (5.4) and by the definition of the Dirac delta function, integrating the second term on the right hand side of (5.1) gives

$$\begin{aligned} \int_{\Omega(t)} \int_0^L f_i(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\, ds\, d\boldsymbol{x} &= \int_0^L f_i(s,t) \int_{\Omega(t)} \delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\, d\boldsymbol{x}\, ds \\ &= \int_{\{s:\boldsymbol{X}(s,t)\in\Omega(t)\}} f_i(s,t)\, ds. \end{aligned} \tag{5.7}$$

After collecting terms, we have

$$\frac{d}{dt}\int_{\Omega(t)} \rho\, u_i\, d\boldsymbol{x} = \int_{\partial\Omega(t)} \sigma_{ij}\, n_j\, dl + \int_{\{s:\boldsymbol{X}(s,t)\in\Omega(t)\}} f_i(s,t)\, ds. \tag{5.8}$$

Let $\Omega(t)$ collapse to a closed thin strip $\Omega_\epsilon(t)$. The first term of the right hand side of (5.8) can be written as

$$\int_{\partial\Omega(t)} \sigma_{ij}\, n_j\, dl = \int_{\partial\Omega_\epsilon^+(t)} \sigma_{ij}\, n_j\, dl + \int_{\partial\Omega_\epsilon^-(t)} \sigma_{ij}\, n_j\, dl, \tag{5.9}$$

where $\Omega_\epsilon^+(t)$ is the outer boundary of the thin strip $\Omega_\epsilon(t)$ and $\Omega_\epsilon^-(t)$ is the inner boundary.

When the width of the strip approach zero, *i.e.* $\Omega_\epsilon(t) \longrightarrow \Gamma(t)$, the normal direction of the outer boundary is opposite to the one of the inner boundary. Equation (5.9) is then

$$\begin{aligned} \int_{\partial\Omega(t)} \sigma_{ij}\, n_j\, dl &= \int_{\Gamma^+} \sigma_{ij}^+\, n_j\, dl - \int_{\Gamma^-} \sigma_{ij}^-\, n_j\, dl \\ &= \int_\Gamma [\![\sigma_{ij} n_j]\!]\, dl \end{aligned} \tag{5.10}$$

where $[\![\cdot]\!]$ denotes the jump across the interface. The left hand side of (5.8) is zero, because the area $\Omega(t)$ approaches zero. Therefore when the arbitrary area $\Omega(t)$ collapses to a surface $\Gamma(t)$ that has codimension one and contains the set $\{s : \boldsymbol{X}(s,t) \in \Omega(t)\}$, equation (5.8) becomes

$$0 = \int_{\Gamma(t)} [\![\sigma_{ij} n_j]\!]\, dl + \int_{\{s:\boldsymbol{X}(s,t)\in\Omega(t)\}} f_i(s,t)\, ds, \tag{5.11}$$

or by the change of variables

$$\int_{\Gamma(t)} [\![\sigma_{ij} n_j]\!]\, dl = -\int_{\Gamma(t)} f_i(s,t)\frac{ds}{dl}\, dl. \tag{5.12}$$

Since the domain of integration is arbitrary, we have

$$[\![\sigma_{ij}]\!] n_j = -f_i\frac{ds}{dl} = -\frac{f_i}{\left|\partial \boldsymbol{X}/\partial s\right|}. \tag{5.13}$$

Taking normal and tangential components of (5.13), we obtain

$$n_i[\![\sigma_{ij}]\!]n_j = -\frac{\boldsymbol{f}\cdot\boldsymbol{n}}{\left|\partial \boldsymbol{X}/\partial s\right|}, \tag{5.14}$$

$$\tau_i[\![\sigma_{ij}]\!]n_j = -\frac{\boldsymbol{f}\cdot\boldsymbol{\tau}}{\left|\partial \boldsymbol{X}/\partial s\right|}. \tag{5.15}$$

5.2 Jump conditions

For an elastic membrane immersed in incompressible fluid, we assume the following:

1. There is no flow of fluid through the interface, and there is no fluid slip at the interface. The velocity is continuous across the interface. We have

$$[\![\boldsymbol{u}]\!] = 0, \tag{5.16}$$

   which implies the jump in the tangential derivative of $\boldsymbol{u}$ is zero. Thus,

$$[\![(\boldsymbol{\tau}\cdot\nabla\boldsymbol{u})_j]\!] = \left[\!\!\left[\tau_i\left(\frac{\partial\, u_j}{\partial\, x_i}\right)\right]\!\!\right] = 0. \tag{5.17}$$

   We also assume that the membrane (interface) is deforming continuously in fluid which means the interface is moving with the fluid and the velocity near the interface is normally not zero. Therefore even though we have the jump conditions (5.16) and (5.17), $\boldsymbol{u}^{\pm} \neq 0$ and $\tau_i\left(\dfrac{\partial\, u_j}{\partial\, x_i}\right)^{\pm} \neq 0$ in general. $\boldsymbol{u}^{+}$ means the velocity at the "+" side of the interface.

2. The fluid is incompressible which means

$$\nabla\cdot\boldsymbol{u} = 0,$$

   or in tensor notation,

$$\frac{\partial\, u_i}{\partial\, x_i} = 0. \tag{5.18}$$

   In addition, we have the tensor identity

$$n_i\, n_j + \tau_i\, \tau_j = \delta_{ij}. \tag{5.19}$$

We note that equations (5.17), (5.18), and (5.19) hold for both two dimensions and three dimensions.

In the following sections we show that the jump conditions for the elastic immersed interface can be derived from the above facts.

5.2.1 Uniform viscosity across the interface.

From (5.3), the fluid stress tensor, we have

$$
\begin{aligned}
n_i [\![\sigma_{ij}]\!] n_j &= n_i \left[\!\!\left[-p\,\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]\!\!\right] n_j \\
&= -[\![p]\!]\delta_{ij}\, n_i\, n_j + n_i \left[\!\!\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]\!\!\right] n_j \\
&= -[\![p]\!] + \mu \left[\!\!\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right]\!\!\right] (\delta_{ij} - \tau_i\,\tau_j).
\end{aligned}
\tag{5.20}
$$

The third equality is true because $\boldsymbol{n}$ is a unit normal and thus $n_i^2 = 1$. This implies $\delta_{ij}\, n_i\, n_j = 1$. Using this and (5.19) we have equation (5.20).

In addition, (5.18) implies

$$\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\delta_{ij} = 0, \tag{5.21}$$

and (5.17) implies

$$\left[\!\!\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right]\!\!\right] \tau_i\,\tau_j = 0. \tag{5.22}$$

(5.21) and (5.22) together imply

$$n_i [\![\sigma_{ij}]\!] n_j = -[\![p]\!]. \tag{5.23}$$

Similarly,

$$
\begin{aligned}
\tau_i [\![\sigma_{ij}]\!] n_j &= -[\![p]\!]\delta_{ij}\,\tau_i\, n_j + \left[\!\!\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]\!\!\right] \tau_i\, n_j \\
&= \mu\tau_i \left[\!\!\left[n_j \left(\frac{\partial u_i}{\partial x_j}\right)\right]\!\!\right] \\
&= \mu\boldsymbol{\tau} \cdot \left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \right]\!\!\right].
\end{aligned}
\tag{5.24}
$$

For the first equality in (5.24), it is easy to check that $\delta_{ij}\,\tau_i\, n_j = 0$, and we have

$$\left[\!\!\left[\mu \frac{\partial u_j}{\partial x_i} \right]\!\!\right] \tau_i\, n_j = 0$$

by (5.17).

When we compare (5.14) with (5.23), and (5.15) with (5.24), we have

$$[\![p]\!] = \frac{\boldsymbol{f}\cdot\boldsymbol{n}}{\left|\partial \boldsymbol{X}/\partial s\right|}, \tag{5.25}$$

$$\mu\boldsymbol{\tau} \cdot \left[\!\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \right]\!\!\right] = -\frac{\boldsymbol{f}\cdot\boldsymbol{\tau}}{\left|\partial \boldsymbol{X}/\partial s\right|}. \tag{5.26}$$

These are the jump conditions for an elastic immersed boundary in a uniform-viscosity fluid. One can find (5.25) and (5.26) in [83].

We continue simplifying (5.26). For this purpose, we can decompose $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}$ into two components as

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \cdot \boldsymbol{n}\right)\boldsymbol{n} + \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \cdot \boldsymbol{\tau}\right)\boldsymbol{\tau} \tag{5.27}$$

The following Lemma shows that the normal component of the normal derivative of the velocity is continuous.

Lemma 1 *For incompressible fluid, we have the jump condition*

$$\left[\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \cdot \boldsymbol{n}\right]\!\right] = 0$$

across an elastic membrane.

Proof: Because the proof is the same for either two dimensions or three dimensions, we show it in three-dimensional space. In three dimensions, let x, y, z be a system of Cartesian coordinates, and x^*, y^*, z^* be a system of local orthogonal curvilinear coordinates. For a given vector $\boldsymbol{v}$, it can be represented by

$$\boldsymbol{v} = v_1\,\boldsymbol{i} + v_2\,\boldsymbol{j} + v_3\,\boldsymbol{k} \quad \text{and} \quad \boldsymbol{v} = v_1^*\,\boldsymbol{i^*} + v_2^*\,\boldsymbol{j} + v_3^*\,\boldsymbol{k^*}, \tag{5.28}$$

for the two systems of coordinates. Suppose that $\boldsymbol{i^*} = \boldsymbol{n}$ is the normal, $\boldsymbol{j^*} = \boldsymbol{\tau}_1$ and $\boldsymbol{k^*} = \boldsymbol{\tau}_2$ the two tangential vectors. If these vectors are represented by

$$\begin{aligned} \boldsymbol{n} &= n_1\,\boldsymbol{i} + n_2\,\boldsymbol{j} + n_3\,\boldsymbol{k} \\ \boldsymbol{\tau}_1 &= \alpha_1\,\boldsymbol{i} + \alpha_2\,\boldsymbol{j} + \alpha_3\,\boldsymbol{k} \\ \boldsymbol{\tau}_2 &= \beta_1\,\boldsymbol{i} + \beta_2\,\boldsymbol{j} + \beta_3\,\boldsymbol{k}, \end{aligned} \tag{5.29}$$

it is easy to show that the transformation of the Cartesian $xyz-$ system to the Cartesian $x^*y^*z^*-$ system is of the form

$$\begin{aligned} x^* &= n_1\,x + n_2\,y + n_3\,z + b_1 \\ y^* &= \alpha_1\,x + \alpha_2\,y + \alpha_3\,z + b_2 \\ z^* &= \beta_1\,x + \beta_2\,y + \beta_3\,z + b_3, \end{aligned} \tag{5.30}$$

where b_1, b_2 , and b_3 are constants [56].

Suppose that the velocity in three dimensions is a function $\boldsymbol{u} = (u, v, w)$. The fluid is incompressible which means $\nabla \cdot \boldsymbol{u} = 0$, so by the chain-rule we have

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \\ &= \left(\frac{\partial u}{\partial x^*}, \frac{\partial v}{\partial x^*}, \frac{\partial w}{\partial x^*}\right) \cdot \left(\frac{\partial x^*}{\partial x}, \frac{\partial x^*}{\partial y}, \frac{\partial x^*}{\partial z}\right) \\ &+ \left(\frac{\partial u}{\partial y^*}, \frac{\partial v}{\partial y^*}, \frac{\partial w}{\partial y^*}\right) \cdot \left(\frac{\partial y^*}{\partial x}, \frac{\partial y^*}{\partial y}, \frac{\partial y^*}{\partial z}\right) \\ &+ \left(\frac{\partial u}{\partial z^*}, \frac{\partial v}{\partial z^*}, \frac{\partial w}{\partial z^*}\right) \cdot \left(\frac{\partial z^*}{\partial x}, \frac{\partial z^*}{\partial y}, \frac{\partial z^*}{\partial z}\right) = 0. \end{aligned} \tag{5.31}$$

In vector notation (5.31) is

$$\begin{aligned}\nabla\cdot\boldsymbol{u} &= (\nabla u\cdot\boldsymbol{n},\nabla v\cdot\boldsymbol{n},\nabla w\cdot\boldsymbol{n})\cdot\boldsymbol{n}\\ &+(\nabla u\cdot\boldsymbol{\tau}_1,\nabla v\cdot\boldsymbol{\tau}_1,\nabla w\cdot\boldsymbol{\tau}_1)\cdot\boldsymbol{\tau}_1\\ &+(\nabla u\cdot\boldsymbol{\tau}_2,\nabla v\cdot\boldsymbol{\tau}_2,\nabla w\cdot\boldsymbol{\tau}_2)\cdot\boldsymbol{\tau}_2=0.\end{aligned}\tag{5.32}$$

Because $[\![\nabla\cdot\boldsymbol{u}]\!]=0$, and by (5.17) we have $[\![(\nabla u\cdot\boldsymbol{n},\nabla v\cdot\boldsymbol{n},\nabla w\cdot\boldsymbol{n})\cdot\boldsymbol{n}]\!]=0$. □

Therefore in two dimensions, (5.26) can be written as

$$\mu\left[\!\!\left[\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right]\!\!\right]=-\frac{\boldsymbol{f}\cdot\boldsymbol{\tau}}{\left|\partial\boldsymbol{X}/\partial s\right|}\boldsymbol{\tau},\tag{5.33}$$

by (5.27) and Lemma 1.

We conclude the above discussion by the following theorem:

Theorem 1 *For two-dimensional uniform-viscosity fluid the pressure and the derivative of the velocity have the following jump conditions across an elastic immersed boundary:*

$$[\![p]\!]=\frac{\boldsymbol{f}\cdot\boldsymbol{n}}{\left|\partial\boldsymbol{X}/\partial s\right|},\tag{5.34}$$

$$\mu\left[\!\!\left[\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right]\!\!\right]=-\frac{\boldsymbol{f}\cdot\boldsymbol{\tau}}{\left|\partial\boldsymbol{X}/\partial s\right|}\boldsymbol{\tau},\tag{5.35}$$

where s is the arc length function for the resting membrane.

We have shown the jumps in pressure and in the derivative of the velocity. Next we want to derive the jump in the derivative of the pressure. We begin our derivation by the following Lemma:

Lemma 2 *The jump in the material derivative of velocity is zero,*

$$\left[\!\!\left[\frac{D\boldsymbol{u}}{Dt}\right]\!\!\right]=[\![\boldsymbol{u}_t+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]\!]=0.\tag{5.36}$$

Proof: Since the velocity is continuous across the interface Γ, we have

$$\begin{aligned}u^+(x(t),y(t),t) &= u^-(x(t),y(t),t),\\ v^+(x(t),y(t),t) &= v^-(x(t),y(t),t).\end{aligned}$$

Taking the time derivative for both sides, we obtain

$$\begin{aligned}u_t^+ + u_x^+x_t^+ + u_y^+y_t^+ &= u_t^- + u_x^-x_t^- + u_y^-y_t^-,\\ v_t^+ + v_x^+x_t^+ + v_y^+y_t^+ &= v_t^- + v_x^-x_t^- + v_y^-y_t^-.\end{aligned}$$

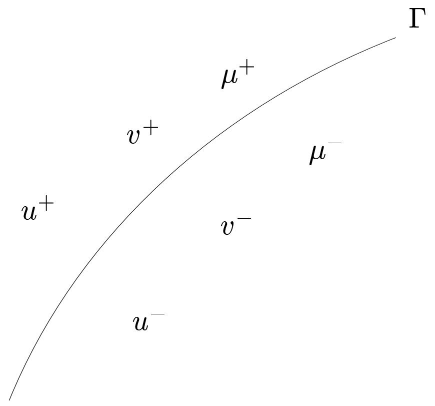


Figure 5.1: '+' sign is the outward direction of the interface Γ.

Since the interface has no-slip boundary condition, we have $\dfrac{\partial \boldsymbol{X}}{\partial t} = \boldsymbol{u}(x,\, y,\, t)$. Hence, we obtain

$$\begin{aligned} u_t^+ + u^+ u_x^+ + v^+ u_y^+ &= u_t^- + u^- u_x^- + v^- u_y^-\,, \\ v_t^+ + u^+ v_x^+ + v^+ v_y^+ &= v_t^- + u^- v_x^- + v^- v_y^-\,, \end{aligned}$$

which is the material derivative. Therefore, $\left[\!\left[\dfrac{D\boldsymbol{u}}{Dt} \right]\!\right] = 0.$ □

In addition to the jump conditions in Theorem 1, if the immersed interface is a closed curve then, with the help of Lemma 2, one can derive the jump condition for the derivative of the pressure as follows.

The momentum equation, (5.1), has the vector form

$$\rho(\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}) = -\nabla p + \mu\,\nabla^2\boldsymbol{u} + \int_0^L \boldsymbol{f}(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,ds. \tag{5.37}$$

Because the viscosity μ is uniform throughout the fluid, we have

$$\nabla\cdot(\mu\nabla^2\boldsymbol{u}) = \mu\nabla^2(\nabla\cdot\boldsymbol{u}) = 0, \tag{5.38}$$

by incompressibility. Therefore, applying the divergence operator $\nabla\cdot$ to (5.37), we obtain

$$\nabla\cdot(\rho\,\frac{D\boldsymbol{u}}{Dt}) = -\nabla^2 p + \nabla\cdot\int_0^L \boldsymbol{f}(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,ds. \tag{5.39}$$

In two-dimensional space, let $\phi(x,\, y)$ be an arbitrary smooth test function on a closed belt-shape domain $\Omega(t)$, which contains a closed curve Γ. Multiplying $\phi(x,\, y)$ to (5.39) and integrating it over the domain $\Omega(t)$ gives

$$\int_\Omega \nabla\cdot(\rho\,\frac{D\boldsymbol{u}}{Dt})\,\phi\,d\boldsymbol{x} = -\int_\Omega \nabla^2 p\,\phi\,d\boldsymbol{x} + \int_\Omega \nabla\cdot\int_0^L \boldsymbol{f}(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,ds\,\phi\,d\boldsymbol{x}. \tag{5.40}$$

Assuming that the density ρ is continuous across Γ, by Green's formulas (see for example [34] Appendix C) and the incompressibility, the left hand side of (5.40) is

$$\begin{aligned}
\int_\Omega \nabla\cdot(\rho\frac{D\boldsymbol{u}}{D\,t})\,\phi\,d\boldsymbol{x} &= -\int_\Omega \rho\frac{D\boldsymbol{u}}{D\,t}\cdot\nabla\phi\,d\boldsymbol{x}+\int_{\partial\Omega}\rho\frac{D\boldsymbol{u}}{D\,t}\phi\cdot\boldsymbol{n}\,dl\\
&= -\frac{d}{d\,t}\int_\Omega \rho\,\boldsymbol{u}\cdot\nabla\phi\,d\boldsymbol{x}+\int_{\partial\Omega}\rho\frac{D\boldsymbol{u}}{D\,t}\phi\cdot\boldsymbol{n}\,dl\\
&\to 0+\int_0^{L_c}\rho\left[\!\!\left[\frac{D\boldsymbol{u}}{D\,t}\right]\!\!\right]\phi\cdot\boldsymbol{n}\,dl,\qquad \text{as}\quad \Omega(t)\longrightarrow\Gamma\\
&= 0,\qquad \text{by}\quad \text{Lemma 2,}
\end{aligned}$$

where L_c is the length of Γ.

Similarly, using the Green's formulas twice, the first term of the right hand side of (5.40) is

$$\begin{aligned}
-\int_\Omega \nabla^2 p\,\phi\,d\boldsymbol{x} &= -\int_{\partial\Omega}\frac{\partial p}{\partial\boldsymbol{n}}\,\phi\,dl+\int_\Omega \nabla p\cdot\nabla\phi\,d\boldsymbol{x}\\
&= -\int_{\partial\Omega}\frac{\partial p}{\partial\boldsymbol{n}}\,\phi\,dl+\int_{\partial\Omega}p(\nabla\phi\cdot\boldsymbol{n})\,dl-\int_\Omega p\nabla^2\phi\,d\boldsymbol{x}\\
&\to -\int_0^{L_c}\left[\!\!\left[\frac{\partial p}{\partial\boldsymbol{n}}\right]\!\!\right]\phi\,dl+\int_0^{L_c}[\![p]\!]\phi_n\,dl-0,
\end{aligned}$$

as $\Omega(t)\longrightarrow\Gamma$.

By the definition of the derivative, in the sense of distributions (see [88] for example), the last term in (5.40) is

$$\begin{aligned}
\int_\Omega \nabla\cdot\left(\int_0^L \boldsymbol{f}\,\delta(\boldsymbol{x}-\boldsymbol{X})\,ds\right)\phi\,d\boldsymbol{x} &= -\int_\Omega\left(\int_0^L \boldsymbol{f}\,\delta(\boldsymbol{x}-\boldsymbol{X})\,ds\right)\cdot\nabla\phi\,d\boldsymbol{x}\\
&= -\int_0^L\left(\int_\Omega \boldsymbol{f}\,\delta(\boldsymbol{x}-\boldsymbol{X})\,d\boldsymbol{x}\right)\cdot\nabla\phi\,ds\\
&= -\int_0^L \boldsymbol{f}\cdot\nabla\phi\,ds.
\end{aligned}$$

After collecting terms and noting that

$$\begin{aligned}
[\![p]\!] &= (\boldsymbol{f}\cdot\boldsymbol{n})\,\frac{ds}{dl},\\
\boldsymbol{f}\cdot\nabla\phi &= (\boldsymbol{f}\cdot\boldsymbol{n})\,\phi_n+(\boldsymbol{f}\cdot\boldsymbol{\tau})\,\phi_s,
\end{aligned}$$

(5.39) becomes

$$\int_0^{L_c}\left[\!\!\left[\frac{\partial p}{\partial\boldsymbol{n}}\right]\!\!\right]\phi\,dl=-\int_0^L \hat{f}_2\,\phi_s\,ds, \tag{5.41}$$

as $\Omega(t)\longrightarrow\Gamma$. Again $\hat{f}_2=\boldsymbol{f}\cdot\boldsymbol{\tau}$ is the tangential force strength. Applying integration by parts to the right hand side of (5.41) and noting that Γ_r, the resting configuration of Γ, is a closed curve and ϕ is an arbitrary smooth function, we obtain

$$\int_0^{L_c}\left[\!\!\left[\frac{\partial p}{\partial\boldsymbol{n}}\right]\!\!\right]\phi(l)\,dl=\int_0^L\frac{\partial\hat{f}_2(s)}{\partial s}\phi(s)\,ds \tag{5.42}$$

where L_c is the length of Γ and L is the length of Γ_r. It is not straightforward to obtain a closed form for the jump condition $\left[\!\left[\frac{\partial p}{\partial \boldsymbol{n}}\right]\!\right]$ from (5.42). However, if we consider the force density $\overline{\boldsymbol{f}}$ that is the force per unit length of the current configuration, the singular force $\boldsymbol{F}$ is

$$\boldsymbol{F} = \int_0^{L_c} \overline{\boldsymbol{f}}(l,t)\delta(\boldsymbol{x} - \boldsymbol{X}(l,t))\, dl, \tag{5.43}$$

where dl is the measure for Γ.

With the singular force $\boldsymbol{F}$ parameterized by (5.43), equation (5.42) simply gives

$$\left[\!\left[\frac{\partial p}{\partial \boldsymbol{n}}\right]\!\right] = \frac{\partial \hat{\overline{f}}_2}{\partial l}, \tag{5.44}$$

where $\hat{\overline{f}}_2 = \overline{\boldsymbol{f}} \cdot \tau$.

We conclude the above discussion by the following theorem:

Theorem 2 *For incompressible fluid, if the density is continuous and the viscosity is uniform across the immersed interface Γ, which is a closed curve, then we have the jump condition*

$$\left[\!\left[\frac{\partial p}{\partial \boldsymbol{n}}\right]\!\right] = \frac{\partial \hat{\overline{f}}_2(l,t)}{\partial l}, \tag{5.45}$$

where $\hat{\overline{f}}_2(l,t)$ is the tangential force strength of the membrane and l is the arc length of the membrane.

Remark: The formulation of the jump conditions in Theorem 1 depends on how we parameterize the singular force $\boldsymbol{F}$. For free surface problems with contact discontinuities, the force is based on the curvature of the current configuration and there is no natural reference configuration to use as a global parameterization, the way *the resting* arc length along the unstretched membrane used in the elastic membrane case. The singular force is parameterized by the arc length for the current configuration l, instead of s the arc length for the resting configuration. The force density is integrated with respect to the measure dl instead of ds in all our derivations. There is no need to perform change of variables to ally the two different measures throughout our proofs. The jump conditions in Theorem 1 then reduce to

$$\begin{aligned} [\![p]\!] &= \hat{\overline{f}}_1, \\ \mu\left[\!\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right]\!\right] &= -\hat{\overline{f}}_2, \end{aligned} \tag{5.46}$$

where $\hat{\overline{f}}_1 = \overline{\boldsymbol{f}} \cdot \boldsymbol{n}$ and $\hat{\overline{f}}_2 = \overline{\boldsymbol{f}} \cdot \boldsymbol{\tau}$. The function $\overline{\boldsymbol{f}}$ is the force per unit length of the current configuration and l is the arc length along the current configuration of the membrane.

5.2.2 Discontinuous viscosity across the interface.

Suppose that the fluid has discontinuous viscosity and the viscosity is a piecewise constant function. This means the viscosity is a constant on each side of the interface and the jump across the interface is also a constant, *i.e.* $[\![\mu]\!] \equiv$ constant.

Again from the fluid stress tensor, the jump condition in the normal direction in two dimensions is

$$\left[\!\!\left[-p + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_i n_j \right]\!\!\right] = -\frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\left|\partial \boldsymbol{X} / \partial s\right|}, \tag{5.47}$$

or

$$\left[\!\!\left[p - 2\mu\left(\frac{\partial u_i}{\partial x_j}\right) n_i n_j \right]\!\!\right] = \frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\left|\partial \boldsymbol{X} / \partial s\right|}, \tag{5.48}$$

since $i\,, j$ are symmetric. In vector notation, (5.48) is

$$[\![p]\!] - [\![2\,\mu(\nabla u \cdot \boldsymbol{n},\, \nabla v \cdot \boldsymbol{n})]\!] \cdot \boldsymbol{n} = \frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\left|\partial \boldsymbol{X} / \partial s\right|}. \tag{5.49}$$

Note that Lemma 1 remains valid for discontinuous viscosity. Therefore (5.49) becomes

$$[\![p]\!] - 2\,[\![\mu]\!](\nabla u \cdot \boldsymbol{n},\, \nabla v \cdot \boldsymbol{n}) \cdot \boldsymbol{n} = \frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\left|\partial \boldsymbol{X} / \partial s\right|} \tag{5.50}$$

by Lemma 1.

In two-dimensional space, the tangential jump condition is

$$\left[\!\!\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \right]\!\!\right] \tau_i n_j = -\frac{\boldsymbol{f} \cdot \boldsymbol{\tau}}{\left|\partial \boldsymbol{X} / \partial s\right|}, \tag{5.51}$$

or in its vector form

$$\left[\!\!\left[\mu\boldsymbol{\tau} \cdot \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} + \mu(\boldsymbol{\tau} \cdot \nabla \boldsymbol{u})_j\, n_j \right]\!\!\right] = -\frac{\boldsymbol{f} \cdot \boldsymbol{\tau}}{\left|\partial \boldsymbol{X} / \partial s\right|}. \tag{5.52}$$

Because the interface is an elastic membrane, (5.52) can be rewritten as

$$\left[\!\!\left[\mu\boldsymbol{\tau} \cdot \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \right]\!\!\right] + [\![\mu]\!](\boldsymbol{\tau} \cdot \nabla \boldsymbol{u})_j\, n_j = -\frac{\boldsymbol{f} \cdot \boldsymbol{\tau}}{\left|\partial \boldsymbol{X} / \partial s\right|}, \tag{5.53}$$

by (5.17).

We have shown the jump conditions for an elastic membrane immersed in fluid in two dimensions. We expect that the parameterization for the singular force will complicate the derivation for the three-dimensional case.

5.2.3 Jump conditions for the interface between fluids

From the previous sections, we have derived the jump conditions for elastic interfaces in two dimensions. In general, the interface of two fluids should satisfy a no-slip condition (It is true for most two-fluid flow. See [86] p121.) Hence (5.17) is still valid. Therefore a similar set of jump conditions as equations (5.50) and (5.53) can be written down for the fluid interface separating different fluids with different viscosities provided μ is constant within each fluid. These jump conditions are useful when the immersed interface method is applied to a fluid interface problem.

We note the following for fluid interface problems.

1. The singular force consists of surface tension and is dependent on the curvature of the current configuration of the interface.

2. The interface is not an elastic material and cannot bear tangential forces.

Using Lemma 1, (5.20), (5.24), and (5.17) the three-dimensional multiphase flow has jump conditions in their vector forms as follows:

$$[\![p]\!] - 2\,[\![\mu]\!](\nabla u \cdot \boldsymbol{n},\, \nabla v \cdot \boldsymbol{n},\, \nabla w \cdot \boldsymbol{n}) \cdot \boldsymbol{n} = \gamma(s)\kappa, \tag{5.54}$$

$$\begin{aligned} &[\![\mu(\nabla u \cdot \boldsymbol{n},\, \nabla v \cdot \boldsymbol{n}, \nabla w \cdot \boldsymbol{n}) \cdot \boldsymbol{\tau}_1]\!] \\ &\qquad + [\![\mu]\!](\nabla u \cdot \boldsymbol{\tau}_1,\, \nabla v \cdot \boldsymbol{\tau}_1,\, \nabla w \cdot \boldsymbol{\tau}_1) \cdot \boldsymbol{n} = 0 \end{aligned} \tag{5.55}$$

$$\begin{aligned} &[\![\mu(\nabla u \cdot \boldsymbol{n},\, \nabla v \cdot \boldsymbol{n}, \nabla w \cdot \boldsymbol{n}) \cdot \boldsymbol{\tau}_2]\!] \\ &\qquad + [\![\mu]\!](\nabla u \cdot \boldsymbol{\tau}_2,\, \nabla v \cdot \boldsymbol{\tau}_2,\, \nabla w \cdot \boldsymbol{\tau}_2) \cdot \boldsymbol{n} = 0 \end{aligned} \tag{5.56}$$

where $\gamma(s)$ is the surface tension and κ is the mean curvature [28, 54].

5.3 Conclusions

We have derived the jump conditions for an elastic membrane immersed in fluid or a fluid-fluid interface. We classify the jump conditions and clarify the necessary conditions for each category. Although the derivation can be messy in three dimensions , we do not see great obstacles to prevent us from extending the derivation from two dimensions to three dimensions. However, since all our computations in this thesis are limited to two dimensions, we will not pursue the three-dimensional case here.

Chapter 6

CONCLUSIONS AND OUTLOOK

6.1 Thesis contribution

The contribution of this thesis includes two parts. The first is a new class of projection methods for incompressible Navier-Stokes equations. Although projection methods have been used for solving incompressible flow since the late 1960's, it is still a challenge to develop a method that is high-order accurate and robust for flow at high Reynolds numbers. Especially when the system contains discontinuous conserved quantities, such as density in a variable density flow, it is desirable to have a method that gives high resolution results for these quantities. The projection method developed in this thesis is not only robust for flow at high Reynolds numbers, $Re = 10^4$ for instance, it also gives high-resolution results for discontinuous conserved quantities. The main idea of this new version of the projection method is to use the high-resolution finite-volume method from *CLAWPACK* for the advection. These high-resolution methods were developed for hyperbolic systems with discontinuous conserved quantities. Therefore, for incompressible viscous fluid when the Reynolds number is large, the advection dominates and the solution typically has sharp gradients. Our method is desirable for such flow. A fractional step method is used to couple in the viscous terms using a Crank-Nicolson method, and a discrete Poisson problem is solved for the projection step. A finite-difference method is used for this step that is compatible with the finite volume method used for advection.

In this thesis we have shown two examples containing discontinuous conserved quantities, which are solved by our projection methods. One is an example of Rayleigh-Taylor instability. In this example, two layers of fluid are initially at rest in a gravity field with a heavy fluid on top of a lighter fluid. The heavy fluid accelerates into the light fluid under the action of gravity. Compared with the literature, we see that the interface of density in this problem is sharply captured by the high-resolution algorithm implemented in our method, which gives a better prediction of the large-time phenomena, especially for high Reynolds numbers. In addition, large time steps can be used in our method for this problem. The other example is Boussinesq equations modeling two-dimensional Rayleigh-Bénard convection. A fluid is heated from below and cooled from above in a periodic channel with rigid top and bottom. The numerical solution of the fluid temperature computed by our method matches the prediction of linear stability theory. For large Rayleigh number $Ra = 50,000$, the Bénard cell computed from our method does not lose the symmetric structure, while compared with the literature using the lattice Boltzmann method. This shows that our method is robust for problems with sharp gradients in their solutions.

The key contribution of this thesis is to use this new projection method to develop an immersed interface method for incompressible flow with elastic boundaries or surface tension. When an elastic boundary or surface tension is present in the flow, the solutions (or their derivatives) of the incompressible Navier-Stokes equations have jumps across the interface. Our immersed interface method splits the force into normal and tangential components. Only the tangential component is spread using discrete delta functions, while the normal component is used to determine jump conditions for the pressure, which are incorporated into the Poisson problem in the projection step. This gives sharp resolution of the solution across the interface. With this moderate modification, our method improves on the standard immersed boundary method in terms of the resolution, order of accuracy and volume conservation. Because such modification does not change the basic structures of the algorithms for the immersed boundary method, other existing immersed boundary method codes could perhaps adopt our new method.

The test problem of Cortez and Varela [25], who present an asymptotic solution for small amplitude oscillations is used to show the method is convergent and is nearly second order accurate when the force is in the normal direction. An example of viscous fluid with moving elastic membrane is presented in comparison with the immersed boundary method. It shows that our method is consistent with the immersed boundary method at early times, but gives high-resolution results across the interface. At later times our method continues to perform well and maintains volume conservation, while the standard immersed boundary method loses volume of fluid enclosed by the membrane linearly.

In the thesis we also present a fully implicit immersed interface method that allows us to use much larger time steps in case when the membrane is stiff. A simple modification of this implicit method also allows us to model an interface with mass. When the interface has nonzero mass density, the jump conditions used in the algorithm contains an inertial term of mass times acceleration. If a time-centered difference scheme is used to discretize the second order derivative, the interface position at the next time step $\boldsymbol{X}^{n+1}$, and the velocity of the membrane at the next time step $\mathcal{U}(\boldsymbol{X}^{n+1})$ are obtained by solving the nonlinear system arising from using the trapezoidal method for moving the control points. The nonlinear system is solved by a quasi-Newton method.

In addition to introducing new algorithms for incompressible flow with immersed interfaces, we also review and derive the jump conditions for elastic interfaces and for interfaces with surface tension in the last chapter. Some of these jump conditions are used in this thesis and others may be useful in further extending this work.

6.2 Future work

Future research work points in two directions from the current standing. One is improving and extending this new immersed interface method, the other is applying it to specific applications. We want to extend the method to a more realistic fluid model, including membranes with jumps in the viscosity or density. In Chapter 5, we have seen that when the viscosity is discontinuous across the interface, the jump condition for the pressure couples the jump in the viscosity and the gradients of velocity, and hence a careful study is required of how to implement the pressure jump conditions in the projection step. One idea to implement the pressure jump conditions for the membrane with discontinuous viscosity is to utilize the implicit scheme developed in Section 4.6: We set the velocity in the jump condition (5.54) as the variable $\mathcal{U}(\boldsymbol{X})$ in the nonlinear equation (4.52), and iterate the quasi-Newton method until (4.52) converges.

In order to validate our numerical solutions for such complicated systems, we also would like to investigate possible test problems such as extensions of the one in the paper of Cortez and Varela [25].

After determining how to implement the pressure jump conditions in our immersed interface method for fluids with discontinuous viscosity, we can extend our method to different types of interface problems, e.g. the free surface problems discussed in Chapter 1.

The projection method developed in this thesis has been shown capable of handling flow at moderately high Reynolds numbers. It would be of great interest to extend this method to three dimensions. A three-dimensional version of *CLAWPACK* is developed in [57]. As we mention in Chapter 3, 3-D versions of adaptive-mesh-refinement *CLAWPACK*, *AMRCLAW* and *BEARCLAW*, have been developed. Our ultimate goal is to develop a three-dimensional AMR projection method based on the principle of the projection method in Chapter 3.

There are many possible applications for the immersed interface method developed in this thesis. One of them is fluid flexible-body structures, such as simulating viscous flow in a stenotic elastic tube, in which pulsatile flow in an arterial stenosis is modeled. Such application would allow us to study pulsatile flow in stenotic arteries to obtain useful information which may lead early detection, prevention and diagnosis of various cardiovascular diseases, such as heart attack and stroke.

BIBLIOGRAPHY

[1] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. C. Welcome. A conservative adaptive projection method for the variable density Navier-Stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.

[2] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, 17(2):358–369, March 1996.

[3] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.

[4] J. B. Bell, P. Colella, and L. H. Howell. An efficient second-order projection method for viscous incompressible flow. In *Tenth AIAA Computational Fluid Dynamics Conference*, June 1991.

[5] J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *J. Comput. Phys*, 101:334–348, 1992.

[6] M. J. Berger and R. J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35:2298–2316, 1998.

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

[8] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.

[9] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible navier-stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.

[10] F. H. Busse. Non-linear propertites of thermal convection. *Report on the Progress in Physics*, 41(12):1929–1967, 1978.

[11] T. Buttke. *Vortex flows and related numerical methods*, volume 395 of *NATO ASI Series C*, chapter Velocity methods: Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow, pages 39–57. edited by J. T. Beal, G. H. Cottet and S. Huberson, Kluwer Academic Publishers, 1993.

[12] D. Calhoun. *Cartesian grid method for solving the streamfunction-vorticity equations in irregular geometries.* PhD thesis, University of Washington, 1999.

[13] D. Calhoun. A cartesian grid method for solving the streamfunction-vorticity equations in irregular regions. *J. Comput. Phys.*, to appear, 2002.

[14] D. Calhoun and R. J. LeVeque. A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *J. Comput. Phys.*, 156:1–38, 2000.

[15] S. Chandrasekhar. *Hydrodynamic and Hydrmagnetic.* Oxford University Press, 1961.

[16] S. G. Chefranov. The dynamics of point vortex dipoles and spontaneous singularities in three-dimensional turbulent flows. *Sov. Phys. JETP*, 66:85, 1987.

[17] U. Chia, K. N. Chia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48:387–411, 1982.

[18] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:742–762, 1968.

[19] A. J. Chorin. Numerical solutionof incompressible flow problems. *Studies in Numerical Analysis*, 2:64–71, 1968.

[20] A. J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comp.*, 23:341–353, 1969.

[21] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[22] R. Cortez. personal communcation.

[23] R. Cortez. An impulse-base approximation of fluid motion due to the boundary forces. *J. Comput. Phys.*, 123:341–353, 1996.

[24] R. Cortez and M. Minion. The blob projection method for immersed boundary problems. *J. Comput. Phys.*, 161:428–453, 2000.

[25] R. Cortez and D. A. Varela. The dynamics of an elastic membrane using the impulse method. *J. Comput. Phys.*, 138:224–247, 1997.

[26] J Crank. *Free and moving boundary problems.* Oxford science publications. Claredon press · Oxford, 1984.

[27] R. Dillon, L. Fauci, A. Fogelson, and D. Gaver. Modeling biofilm processes using the immersed boundary method. *J. Comput. Phys.*, 129:57–73, 1996.

[28] D. D.Joseph and Y. Y. Renardy. *Fundamental of two fluid dynamics: Mathematical theorey and applications*, volume 1. Springer-Verlag, 1993.

[29] W. E and J.-G. Finite difference schemes for incompressible flows in the velocity-impulse density formulation. *J. Comput. Phys.*, 130:67–76, 1997.

[30] W. E and J.-G. Liu. Gauge method for viscous incompressible flows. unpublished 1996.

[31] W. E and J.-G. Liu. Projection method I: convergence and numerical boundary layers. *SIAM J. Numer. Anal.*, 4:1017–1057, 1995.

[32] W. E and J.-G. Liu. Projection method II: Godunov-Ryabenki analysis. *SIAM J. Numer. Anal.*, 33:1597–1621, 1996.

[33] D. Enright, R. P. Fedkiw, J. Feziger, and I. Mitchell. A hybrid partical level method for improved interface capturing. *preprint*, 2002.

[34] L. C. Evans. *Partial differential equations.* Providence, R.I. : American Mathematical Society, 1998.

[35] P. Fast. *Dynamics of the interfaces in non-Newtonian Hele-Shaw flow.* PhD thesis, New Youk University, 1999.

[36] P. Fast and W. D. Henshaw. Time-accurate computation of viscous flow around deforming bodies using overset grids. *AIAA*, 2002. 15th AIAA computational fluid dynamics conference, accepted.

[37] P. Fast and M. Shelley. An overset grid method for free surface flow. Technical report, LLNL Report, UCRL-JC-136484, 1999.

[38] L. J. Fauci and C. S. Peskin. A computational model of aquatic animal locomotion. *J. Comput. Phys.*, 85-108, 1988.

[39] R. P. Fedkiw, T. Aslam, A. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (The ghost fluid method. *J. Comput. Phys.*, 152:457–492, 1999.

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

[41] Y. Frigneau, J.-L. Guermond, and L. Quartapelle. Approximation of variable density incompressible flows by means of finite elements and finite volumes. *Communications in Numerical Methods in Engineering*, 17:893–902, 2001.

[42] A. Y. Gelfgat. Different modes of Rayleigh-Bénard instability in two- and three-dimensional rectangular enclosures. *J. Comput. Phys.*, 156:300–324, 1999.

[43] A. Greenbaum. *Iterative methods for solving linear system.* Frontiers in applied mathematics. SIAM, 1997.

[44] P. M. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part I: Theory. *International Journal for Numerical Methods in Fluid*, 11:587–620, 1990.

[45] P. M. Gresho and S. T. Chan. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part II: Implememtation. *International Journal for Numerical Methods in Fluid*, 11:621–659, 1990.

[46] J.-L. Guermond and L. Quartapelle. A projection FEM for variable density incompressible flows. *J. Comput. Phys.*, 165:167–188, 2000.

[47] X. He, S. Chen, and G. D. Doolen. A novel thermal model for the lattice Boltzmann method in incompressible limit. *J. Comp. Phys.*, 146:282–300, 1998.

[48] T. Hou. Numerical solutions to free boundary problems. *Acta Numerica*, 4:335–415, 1995.

[49] T. Hou, S. Osher, Z. Li, and H. Zhao. A hybrid method for moving interface problems with application to the hele-shaw flow. *J. Comput. Phys.*, 134:236–252, 1997.

[50] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Removing stiffness from interfacial flows with surfce tension. *J. Comput. Phys.*, 114:312–338, 1994.

[51] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. The long-time motion of vortex sheets with surface tension. *Phys. Fluids*, 114:1933–1950, July 1997.

[52] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Boundary integral methods for multicomponent fluids and multiphase materials. *J. Comput. Phys.*, 169:302–362, 2001.

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

[54] M. Kang, R. P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Compt. Phys.*, 15:323–360, 2000.

[55] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, 59:308–323, 1985.

[56] E. Kreyszig. *Advanced engineering mathematics.* New York : Wiley, 7th edition, 1993.

[57] J.-O. Langseth and R. J. LeVeque. A wave propagation method for three-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 165:126–166, 200.

[58] L. Lee and R. J. LeVeque. An immersed interface method for incompressible Navier-Stokes equations. *Submitted to J. Comput. Phys.*, 2002.

[59] R. J. LeVeque. CLAWPACK software. `http://www.amath.washington.edu/~claw`.

[60] R. J. LeVeque. *Numerical methods for conservation laws.* Birkhäuser, Basel, 1990.

[61] R. J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.

[62] R. J. LeVeque. Wave propagation algorithms for multi-dimensional hyperbolic systems. *J. Comput. Phys.*, 131:327–353, 1997.

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

[64] R. J. LeVeque and Z. Li. Immersed interface method for stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.

[65] R. J. LeVeque and S. Mitran. Wave-propagation methods and software for complex applications. Proceeding of Third International Symposium on Finite Volumes for Complex Applications, June 2002. Porquerolles, France.

[66] R. J. LeVeque and J. Oliger. Numerical methods based on additive splitting for hyperbolic partial differential equations. *Math. Comp.*, 40, 1983.

[67] Z. Li. Immersed interface method for moving interface problems. *Numerical Algorithms*, 14:269–293, 1997.

[68] Z. Li and M. C. Lai. The immersed interface method for the Navier-Stokes equation with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.

[69] X.-D. Liu, R. P. Fedkiw, and M. Kang. A boundary bondition capturing method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 160:151–168, 2000.

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

[71] A. A. Mayo and C.S. Peskin. An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. *Contemp. Math.*, 141:261–277, 1993.

[72] M. L. Minion. A projection method for locally refined grids. *J. Comput. Phys.*, 127:158–178, 1996.

[73] Q. Nie. The nonlinear evolution of vortex sheets with surface tension in axisymmetric flows. *J. Comput. Phys.*, 174:438–459, 2001.

[74] J. Nocedal and S. J. Wright. *Numerical Optimization*. Spriger, 1999. Springer Series in Operations Research.

[75] V. I. Oseledets. On a new way of writing the Navier-Stokes equation: The Hamiltonian formalism. *Commun. Moscow Math. Society*, 44:210–211, 1989.

[76] S. Osher and R. P. Fedkiw. Level set methods: an overview and some recent results. *J. Comput. Phys.*, 169:463–502, 2001.

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

[78] C. S. Peskin. *Lectures on Mathematical Aspects of Physiology*, volume 19. American Mathematical Society, 1981.

[79] C. S. Peskin. The immersed boundary method. *Acta Numerica*, pages 1–39, 2002.

[80] C. S. Peskin and D. M. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 105:113–132, 1980.

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

[82] C. S. Peskin and D. M. McQueen. A general method for the computer simulation of bilogical systems interacting with fluids. In *Symposia of the Society for Experimental Biology*, volume 19. C. P. Ellington and T. J. Pedley eds., Cambridge, 1995.

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

[84] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *International Journal for numerical methods in fluids*, 30:775–793, 1999.

[85] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *Internation Journal for Numerical Methods in Fluid*, 30:775–793, 1999.

[86] C. Pozrikidis. *Introduction to theoretical and computational fluid dynamics.* Oxford University, 1997.

[87] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130:267–282, 1997.

[88] M. Renardy and R. C. Rogers. *An introduction to partial differential equations.* Springer-Verlag, 1992.

[89] J. Shen. On error estimates of some higher order projection and penalty-projection methods for Navier-Stokes equations. *Numer. Math.*, 62:49–73, 1992.

[90] J. Shen. On error estimates of the projection methods for the Navier-Stokes equations: second-order schemes. *Math. Comp.*, 65:1039–1065, 1996.

[91] J. M. Stockie and S. I. Green. Simulating the motion of pulp fibres using the immersed boundary method. *J. Comput. Phys.*, 147:147, 1998.

[92] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, second edition, 1993.

[93] J. C. Strikwerda and Y. S. Lee. The accuracy of the fractional step method. *SIAM J. Numer. Anal.*, 37:37–47, 1999.

[94] G. I. Taylor. On the decay of vortices in a viscous fluid. *Philos. Mag.*, 46:671–674, 1923.

[95] R. Temam. Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des fractionaries II. *Arc. Rational Mech. Anal.*, 33:377–385, 1969.

[96] R. Temam. Remark on the pressure boundary condition for the projection method. *Theoret. Comput. Fluid Dynamics*, 3:181–184, 1991.

[97] A. N. Tikhonov and A. A. Samarskii. Homogeneous difference scheme. *USSR Comput. Math. and Phys.*, 1:5–67, 1962.

[98] A.-K. Tornberg and B. Engquist. The segment projection method for interface tracking. *preprint*, 2002.

[99] G. Tryggvason. Numerical simulations of the Rayleigh-Taylor instability. *J. Comput. Phys.*, 75:253–282, 1988.

[100] C. Tu and C. S. Peskin. Stability and instability in the computation of flows with moving immersed boundaries. *SIAM J. Sci. Statist. Comput.*, 13:70–83, 1992.

[101] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992.

[102] J. Van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. Comput.*, 7:870–891, 1986.

[103] A. Wang and K.-J. Bathe. Displacement/pressure based mixed finite element formulations for acoustic fluid-strucure interaction problems. *International Journal for Numerical Methods in Engineering*, 40:2001–2017, 1997.

[104] A. Wiegmann and K. Bube. The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. *SIAM J. on Numerical Analysis*, 35:177–200, 1998.

[105] A. Wiegmann and K. Bube. The explicit jump immersed interface method: Finite difference methods for pde with piecewise smooth solutions. *SIAM J. on Numerical Analysis*, 37:827–862, 2000.

[106] C. Zhang. *Immersed interface methods for hyperbolic systems of partial differential equations with discontinuous coefficients*. PhD thesis, University of Washington, 1996.

[107] C. Zhang and R. J. LeVeque. The immersed interface method for acoustic wave equations with discontinuous coefficients. *Wave Motion*, 25:237–263, 1997.

[108] J. Zhang, S. Childress, A. Libchaber, and M. Shelley. Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature*, 408:835–838, December 2000.

[109] L. Zhu and C. S. Peskin. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. *J. Comput. Phys.*, 2002. in press.

VITA

Name: Long Lee
Citizenship: Taiwan

Education

Ph.D. in Applied Mathematics, 2002
University of Washington, Seattle.

M.A. in Applied Mathematics, 1998
University of Maryland, College Park.

M.S. in Geophysics, 1990
National Central University, Taiwan.

B.S. in Engineering, 1988
National Taiwan University, Taiwan.