Simulation of Thin Elastic Solids in the Incompressible Viscous Flow using Implicit Interface Representation

by

Jae Hyung Kim

Submitted to the Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

Author .....
Aeronautics and Astronautics
January 18, 2010

Certified by .....
Mark Drela
Professor
Thesis Supervisor

Accepted by .....
Eytan H. Modiano
Chairman, Committee on Graduate Students

Simulation of Thin Elastic Solids in the Incompressible Viscous Flow using Implicit Interface Representation

by

Jae Hyung Kim

Submitted to the Aeronautics and Astronautics
on January 18, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis provides a numerical algorithm to solve fluid-structure interaction problems in the Cartesian grid. Unlike the typical Immersed Interfaced Method (IIM), we define thin non-stretchable solid interface with the Level Set function. In addition, we developed a partial differential equation which represents the bending rigidity of the interface. The interface is assumed very thin and has zero elastic stress when it is flat. The interface gives singular forces to the incompressible viscous fluid and the fluid solver handles discontinuities across the interface.

Instead of solving two dynamic systems (i.e., fluid and solid), we solve the fluid field only and solve a convection equation of interface with the local fluid velocity. This idea is valid because of viscous fluid (i.e., velocity is continuous across the interface) as we can see frequently in the IIM. The result shows that elastic interface vibrates and converges to an equilibrium state. The oscillatory motion of the interface depends on the viscosity of fluid, Young's modulus and thickness of interface. The results looks correct physically, and they match with the existing IIM results.

Thesis Supervisor: Mark Drela
Title: Professor

Acknowledgments

First and foremost, I would like to thank Dr. Jean-Christophe Nave and Professor Mark Drela, my thesis advi- sor, for his guidance, support, patience, enthusiasm and attention to detail. Owing to their leadership, I have learned to see a bigger picture, changing my way of thinking. None of the works in this thesis would have been possible without the help of my current/future advisor at MIT. I would like to thank Professor Klaus-Jurgen Bathe for his guidance, enthusiasm and dedication to this study, and to thank my colleagues of Finite Element Research Group at MIT. I would also like to thank Dr. Shingyu Leung for his valuable advices and help regarding the collaborative works.

Special thanks to my family, father, Yong-Gu Kim, my mother, Kyung-Hee Lee, my sisters, Hee-Jin Kim, and especially to my grandmother who died in Summer 2009, for their unlimited mental support during my life at MIT. Thanks to all of my friends for their mental support.

The help and consideration of all those mentioned above are sincerely appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we introduce a numerical method to solve the incompressible Navier-Stokes equations involving flexible interfaces which are represented by the level set function. One of the existing approach to solve fluid-structure interaction problems in the Cartesian grid is the Immersed Interface Method (IIM) in which the material interface is tracked explicitly with a finite number of Lagrangian markers. Even though this explicitly immersed interface model is easy to impose elastic membranes and the singular forces are determined directly from the Lagrangian markers, it requires more complicated interpolation to smear this singular forces or velocities on Cartesian grid. In order to simplify the numerical algorithm, the level set method is used to represent the interface in [10].

So far, in contrast, the level set method did barely appear to be proper to represent the interface such as elastic membranes. Therefore, in terms of fluid interactions, the level set method has been strictly used for curvature dependent problems such as soap bubbles or multi-phase simulations. However, in this paper we introduce a new method to represent the bending elasticity with the level set function so that the level set method can be coupled with the fluid motions involving deformable interfaces.

In particular, we shall evaluate the flapping dynamic equation and develop the singular forces. These normal and tangential force densities are considered as jump conditions which yield discontinuities of fluid variables across the interface. The difference with the typical IIMs is that we enforce the boundary conditions that the

viscous flow on the interface should move or accelerate with the interface. In other words, the velocity of the interface must have the same velocity of the interface for viscous flows. Thus, we can employ the velocity of the fluid to advect the interface, which does not require any temporal projection. If interfaces move with the local fluid velocity, two or more interfaces do not overlap each other as long as the flow field is incompressible even if we apply multiple number of interfaces in one fluid solver.

1.1 Background

In general, the fluid-structure interaction problems strongly relate not only to aerodynamic tunings with respect to the acoustic and structural design, but also to an effective control of large-scale vortical interactions with a flexible body in ocean engineering[3]. Elucidating a flapping problem has also become popular in sorts of engineering sense. There are several recent works that imply possible applications of flapping dynamics such as turbulence reduction (Shen et al.2003), energy generation (Allen and Smits 2001), and biomechanical propulsions (e.g. fish-like swimming or fluttering birds), etc.

While flapping phenomena are easily observed in daily life, the history of computational works on fluid-structure interaction is relatively short compared to problems without interaction. One of the general problems is a moving membrane excited by a viscous fluid. In 1972, Charles S. Peskin developed the Immersed Boundary Methods (IBMs) for the study of blood flow within the human heart [13], which differed substantially from the common CFD methods back then. The IBMs highly reduce the time for grid generation and do not require any grid transformation or deformation, since the boundary conditions can be controlled on Cartesian grids. The IBM yields a problem of how accurately boundary conditions can be imposed. But recent works provide sorts of inter/extrapolations methods so as to handle non-smooth or even discontinuous quantities across immersed boundaries [9]. The IBM sets a number of control points to represent the interface. The singular forces at the control points are spread to the Cartesian grid, which acts a forcing term to solve the Navier-Stokes

equations. However, it smears out sharp interface since it uses the discrete delta function.

On the other hand, the Immersed Interface Method (IIM), which was originally proposed by LeVeque and Li [8], combines the jump contribution into the finite difference schemes and can avoid the smearing sharp shape and maintain second order accuracy. Therefore, applying jump in pressure and in derivatives of both pressure and velocity allows one to solve the Navier-Stokes eqution with a high accuracy. The advantage of the IIMs is that we may set multiple membranes without any conflict between membranes. The main drawback of this method is that a spacial discretization of the Navier-Stokes equations and interpolation the fluid and force quantities onto the Cartesian grid are demanding when we use the Lagrangian marker. In this paper, we present the level set method coupled with the jump conditions from the IIM (See the next section).

Apart from the usage of the Cartesian grid system, there are methods that use the moving grid approach such as a coupled fluid-structure direct simulation (FSDS). Even though the FSDS, which employs body-fitted grid system, allows one to impose the boundary conditions easily, it instead requires grid transformation or grid clustering at the body ends at every time step so as to allow a smooth calculation around a sharp edge[2]. This demends expensive computation cost. In addition, the FSDS solves two dynamic systems using external forces between fluid and structure which are regarded as an input for both. In this case, the old solution must be projected to a newly generated grid to reduce the error. Moreover, it is more difficult to apply multiple number of membranes with the FSDS since the grid generation must reflect the information of shape and location of the interfaces.

1.2 Problem Statement

The objective of this research is to obtain viscous, incompressible flow solutions of fluid-structure interaction problems with an implicit imposition of forces obtained by the level set function. As in the immersed interface method, we use the Cartesian

grid and jump conditions from the singular force density. The jump in pressure and in the derivatives of both pressure and velocity contributes to the finite difference schemes only when the finite difference involves irregular points. This allows the non-smoothness or discontinuities of the solution of the Navier-Stokes equations. However, the velocity must be continuous across the interface for viscous flows, i.e.,

$$[\mathbf{V}]_{\Gamma(t)} = 0$$

Hence the velocity of the fluid at the interface surface, which is smooth, is nothing but the velocity of the interface. Therefore, the location and contour of the interface can be updated for the new time step by solving the advection equation of the level set function. This automatically satisfies the no-penetration condition.

In order to obtain the force density and jump conditions, we use the level set function whose zero level set represent the evolving interface front. Since the level set field is smooth over the entire domain, we can obtain the curvature of the interface or higher derivatives of the level set curve by using the surrounding level set values. In particular, for the bending elasticity of the interface, we introduce an identical partial differential equation using level set functions in Chapter 3. Once the force density is known from the level set field, the entire algorithm is similar to the IIM. The Figure 1 shows the overall rough algorithm which is used for this work. The biggest merit of this approach is that both fluid and structure are moving in the same velocity which governs the advection equations of the level set function, which makes the problem implicit. Since this approach imposes that the structure moves along with the local fluid velocity, we can avoid problems where two or several interfaces intersect or cross through each other.

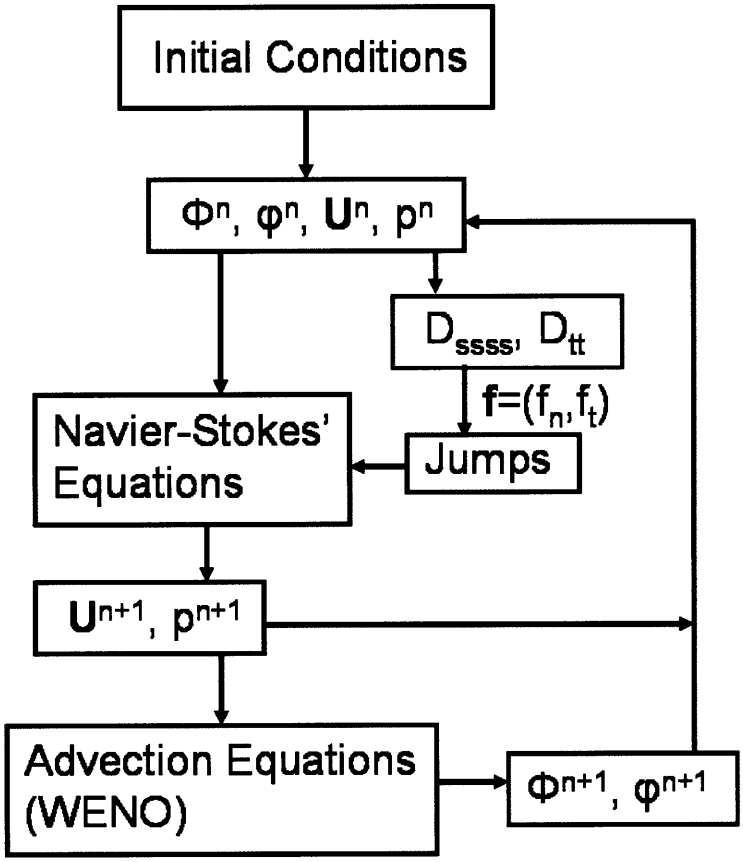


Figure 1-1: Overall Algorithm Map

1.3 Outline of This Thesis

This paper is organized as follows. In Chapter 2, we describe the governing equations of the fluid and structural model, and force density and jump conditions. We present the Navier-Stokes equations and the flapping equation. And the jump relations from the singular force are shown. Chapter 3 presents the level set method. We introduce the identical PDE representation which is equal to the bending elasticity of the interface. In Chapter 4, we states the numerical techniques. Multi-step method, generalized finite difference schemes, jump contribution, interpolation and WENO scheme are shown. Finally, Chapter 5 presents simulation results. Appendix has the exact expression of the identical PDE for bending elasticity.

Chapter 2

Interactive Dynamic Systems

2.1 Explicit Physical Models

We consider a problem of a thin membrane or flag motion which is excited by an incompressible viscous flow, or a problem of a fluid motion which is excited by forces from the interface. Either can be regarded as a fluid-structure interaction in which forces between two media and velocity of fluid govern the two dynamic system. Before entering the implicit physical system, the basic dynamic equations are shown in this section.

2.1.1 Structural Dynamic Model

In this work, the flag can be regarded as a typical cloth which has a low bending rigidity and the fluid-induced tension yields the main portion of the structural restoring force. The solid body is sufficiently thin so that its immersed volume can be neglected for numerical simulations. The governing equation of the flag motion can be derived by Hamilton's principle, linearizing around small body thickness and extensional strain. Zhu and Peskin [13] provided the final form of the equation, permitting arbitrary configuration and orientation of the body, which is described by

$$\rho_s h \frac{\partial^2 \mathbf{X}}{\partial t^2} - Eh \frac{\partial}{\partial s}\left[\left(1 - \left(\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial \mathbf{X}}{\partial s}\right)^{-1/2}\right)\frac{\partial \mathbf{X}}{\partial s}\right] + EI\frac{\partial^4 \mathbf{X}}{\partial s^4} = \mathbf{F}_{ext} \qquad (2.1)$$

where $\mathbf{X} = \{x(s,t), y(s,t)\}$ is the body position vector in Cartesian coordinate, s is the Lagrangian coordinate along the material line, ρ_s is the flag (solid) density, h is the flag thickness, and EI is the bending rigidity (E is the Young's modulus). $\mathbf{F}_{ext}$ transfers the net external force.

The boundary conditions at the leading edge are given as

$$\begin{aligned} &\mathbf{X} = 0 \\ &\mathbf{X}_s = 0 \qquad or \qquad \mathbf{X}_{ss} = 0 \end{aligned} \tag{2.2}$$

where the second restriction depends on the choice of whether it is a rotation free. The natural free-end boundary conditions that satisfy our flapping problem [3] are given as

$$\begin{aligned} &-Eh\left(1 - \left(\tfrac{\partial \mathbf{X}}{\partial s} \cdot \tfrac{\partial \mathbf{X}}{\partial s}\right)^{-1/2}\right) \tfrac{\partial \mathbf{X}}{\partial s} + EI \tfrac{\partial^3 \mathbf{X}}{\partial s^3} = 0 \\ &\tfrac{\partial^2 \mathbf{X}}{\partial s^2} = 0 \end{aligned} \tag{2.3}$$

at any free-end.

2.1.2 Fluid Dynamic Model

The continuity equation for viscous incompressible flow is

$$\nabla \cdot \mathbf{V} = 0 \tag{2.4}$$

and the Navier-Stokes' equations are given as

$$\rho \frac{\partial \mathbf{V}}{\partial t} + \rho \mathbf{V}(\nabla \cdot \mathbf{V}) = -\nabla p + \mu \nabla^2 \mathbf{V} + \rho \mathbf{g} + \rho \mathbf{f}(\mathbf{x}, t) \tag{2.5}$$

where

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma(t)} \mathbf{F}(s,t) \delta(\mathbf{x} - \mathbf{X}(s,t))\, ds$$

is the force acting on the fluid from the interface. The boundary condition on the interface is

$$\mathbf{V}_{fluid} = \mathbf{V}_{interface} \tag{2.6}$$

and an unbounded domain is applied for the boundaries of solver.

2.2 Implicit Physical Models

In this section, we introduce the implicit analytical algorithm to solve fluid-structure interaction problems. In typical approach such as FSDS ([2], [3]) , fluid and structural dynamic equations are solved separately with imposition of forces acting between two media. It is clear that the idea of this work is the same with the FSDS in that we evaluate the forces between structure and fluid. However, we solve Navier-Stokes' equation by evaluating singular force from the structural equation and solve the convection equation of the interface with fluid velocity. It is the reason of calling this method 'implicit'. As in the Immersed Interface Method (IIM) [6], we allow the solutions of the Navier-Stoke's equation to be non-smooth or discontinuous across the interface by changing the finite difference stencils only adjacent grid points of the interface.

2.2.1 Construction of Singular Forces

Remember that the Navier-Stokes equations and flapping equation have different dimensions. The fluid equation has a dimension of $[N/m^3]$ and structural equation has a dimension of $[N/m^2]$. Hence, in order to evaluate the interaction force correctly, let's rewrite the flapping equation as following:

$$\rho_s h \frac{\partial^2 \mathbf{X}}{\partial t^2} = Eh \frac{\partial}{\partial s} \left[\left(1 - \left(\frac{\partial \mathbf{X}}{\partial s} \cdot \frac{\partial \mathbf{X}}{\partial s} \right)^{-1/2} \right) \frac{\partial \mathbf{X}}{\partial s} \right] - EI \frac{\partial^4 \mathbf{X}}{\partial s^4} + \mathbf{F}_{ext} \tag{2.7}$$

Since acceleration $\frac{\partial^2 \mathbf{X}}{\partial t^2}$ in the left hand side is defined in Lagrangian manner, if we rewrite the flapping equation as

$$\frac{\partial^2 \mathbf{X}}{\partial t^2} = \frac{Eh}{\rho_s h}\frac{\partial}{\partial s}\left[\left(1 - \left(\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial \mathbf{X}}{\partial s}\right)^{-1/2}\right)\frac{\partial \mathbf{X}}{\partial s}\right] - \frac{EI}{\rho_s h}\frac{\partial^4 \mathbf{X}}{\partial s^4} + \frac{\mathbf{F}_{ext}}{\rho_s h} \tag{2.8}$$

, then right hand side of above equation can be considered as singular forces acting on the fluid. Note that this equation has a dimension of force per unit mass $[N/kg]$. The first two terms are contributed by the internal force of solid and the third term is related to the direct interaction forces which are pressure difference across the interface and viscous shear stress. Therefore, we can calculate the force on the interface surface, acting on the fluid, by evaluating the right hand side of the following equation.

$$\mathbf{F}(s,t) = \frac{Eh}{\rho_s h}\frac{\partial}{\partial s}\left[\left(1 - \left(\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial \mathbf{X}}{\partial s}\right)^{-1/2}\right)\frac{\partial \mathbf{X}}{\partial s}\right] - \frac{EI}{\rho_s h}\frac{\partial^4 \mathbf{X}}{\partial s^4} + \frac{\mathbf{F}_{ext}}{\rho_s h} \tag{2.9}$$

This should be applied in Equation (2.5). Instead of solving the two dynamic equation, we can introduce the normal and tangential singular force which acts on fluid as jump conditions. For viscous flow, the velocity of fluid on boundary must be same with the velocity of the interface. Therefore, the no-penetration condition can be guaranteed by using the velocity of fluid for the the interface behavior as far as the evaluation of singular forces is correct. We will introduce the singular force evaluation and advection method for the interface in Chapter 3. Once the singular force is determined, the force can be decomposed into normal and tangential component, i.e.,

$$\begin{aligned} f_n &= \mathbf{f}\cdot\mathbf{n} \\ f_\tau &= \mathbf{f}\cdot\tau \end{aligned} \tag{2.10}$$

where f_n, f_τ are the normal and tangential components of singular force, respectively. Additionally, we need to impose the no-slip condition. This can be represented in terms of tangential singular force:

$$f_\tau = -\nu \left[\tau \cdot \frac{\partial \mathbf{V}}{\partial \mathbf{n}} \right] \tag{2.11}$$

where $[\cdot]$ denotes jump condition. The details about relation between singular force and no-slip condition is represented in Chapter 4.3 along with the numerical technique. Thus, the tangential singular force is the sum of Equation (2.10) and (2.11).

2.2.2 Spatial Jump Conditions

Jump conditions are functions of the singular force density. Let τ and $\mathbf{n}$ be the unit tangential and normal vectors, which are defined as ([19], [18])

$$\tau = (\tau_x, \tau_y) = \frac{1}{J}\left(\frac{\partial x}{\partial s}, \frac{\partial y}{\partial s}\right), \qquad J \equiv \sqrt{\left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2} \tag{2.12}$$

$$\mathbf{n} = (n_x,\ n_y) = (\tau_y,\ -\tau_x) \tag{2.13}$$

Jump of a quantity ψ across the interface $[\psi]_\Gamma$ is defined as

$$[\psi]_\Gamma = (\psi)_{\Gamma^+} - (\psi)_{\Gamma^-} \tag{2.14}$$

and the jump conditions in two dimensional space are given in [19].

$$[\mathbf{V}] = 0, \qquad \left[\frac{\partial \mathbf{V}}{\partial x}\right] = -Re\tau_y \tau f_\tau, \qquad \left[\frac{\partial \mathbf{V}}{\partial y}\right] = Re\tau_x \tau f_\tau \tag{2.15}$$

$$[p] = f_n, \qquad \left[\frac{\partial p}{\partial x}\right] = \frac{\tau_x}{J}\frac{\partial f_n}{\partial s} + \frac{\tau_y}{J}\frac{\partial f_\tau}{\partial s}, \qquad \left[\frac{\partial p}{\partial y}\right] = \frac{\tau_y}{J}\frac{\partial f_n}{\partial s} - \frac{\tau_x}{J}\frac{\partial f_\tau}{\partial s} \tag{2.16}$$

$$\mathbf{r}_{u1} = -\frac{1}{J^2}\left(\frac{\partial (J\tau_x)}{\partial s}\left[\frac{\partial \mathbf{V}}{\partial x}\right] + \frac{\partial (J\tau_y)}{\partial s}\left[\frac{\partial \mathbf{V}}{\partial y}\right]\right) \tag{2.17}$$

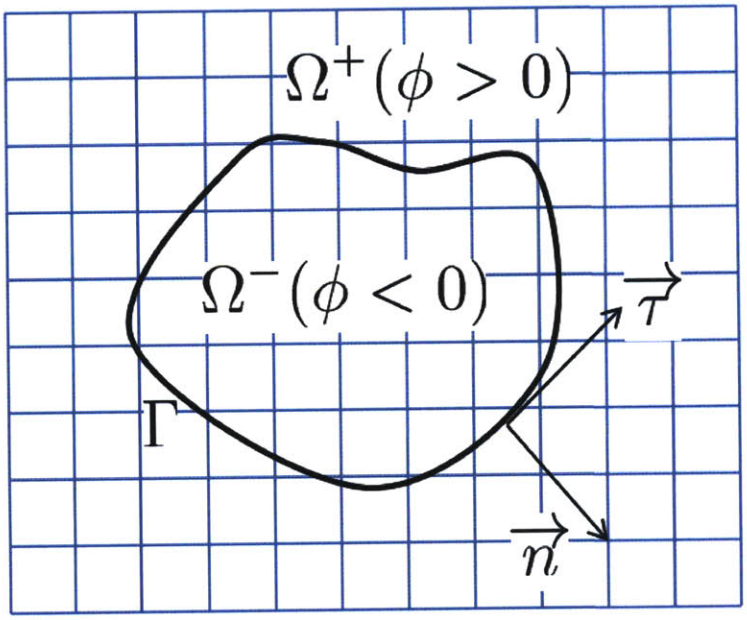


Figure 2-1: A domain with a immersed interface.

$$\mathbf{r}_{u2} = -\frac{1}{J}\left(Re\frac{\partial(f_\tau\tau)}{\partial s} + \frac{\partial n_x}{\partial s}\left[\frac{\partial \mathbf{V}}{\partial x}\right] + \frac{\partial n_y}{\partial s}\left[\frac{\partial \mathbf{V}}{\partial y}\right]\right) \tag{2.18}$$

$$\mathbf{r}_{u3} = Re[\nabla p] \tag{2.19}$$

$$\left[\frac{\partial^2 \mathbf{V}}{\partial x^2}\right] = \mathbf{r}_{u1}(\tau_x^2 - \tau_y^2) + \mathbf{r}_{u2}(2\tau_x\tau_y) + \mathbf{r}_{u3}(\tau_y^2) \tag{2.20}$$

$$\left[\frac{\partial^2 \mathbf{V}}{\partial y^2}\right] = \mathbf{r}_{u1}(\tau_y^2 - \tau_x^2) - \mathbf{r}_{u2}(2\tau_x\tau_y) + \mathbf{r}_{u3}(\tau_x^2) \tag{2.21}$$

$$r_{p1} = \frac{1}{J^2}\left(\frac{\partial^2 f_n}{\partial s^2} - \frac{\partial(J\tau_x)}{\partial s}\left[\frac{\partial p}{\partial x}\right] - \frac{\partial(J\tau_y)}{\partial s}\left[\frac{\partial p}{\partial y}\right]\right) \tag{2.22}$$

$$r_{p2} = \frac{1}{J}\left(\frac{\partial}{\partial s}\left(\frac{1}{J}\frac{\partial f_\tau}{\partial s}\right) - \frac{\partial n_x}{\partial s}\left[\frac{\partial p}{\partial x}\right] - \frac{\partial n_y}{\partial s}\left[\frac{\partial p}{\partial y}\right]\right) \tag{2.23}$$

$$r_{p3} = 2\left[\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right] - 2\left[\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right] \tag{2.24}$$

$$\left[\frac{\partial^2 p}{\partial x^2}\right] = r_{p1}(\tau_x^2 - \tau_y^2) + r_{p2}(2\tau_x\tau_y) + r_{p3}(\tau_y^2) \tag{2.25}$$

$$\left[\frac{\partial^2 p}{\partial y^2}\right] = r_{p1}(\tau_y^2 - \tau_x^2) - r_{p2}(2\tau_x\tau_y) + r_{p3}(\tau_x^2) \tag{2.26}$$

In dimensional Navier-Stokes' equations, Reynolds number is corresponding to $1/\nu$ which is the coefficient of the diffusion term. The form of $[a \cdot b]$ appears in r_{p3}. Note that $[a \cdot b] \neq [a] \cdot [b]$. The correct calculation is

$$[a \cdot b] = a^- \cdot [b] + b^- \cdot [a] + [a] \cdot [b] = a^+ \cdot [b] + b^+ \cdot [a] - [a] \cdot [b] \tag{2.27}$$

where a^-, b^-, a^+ and b^+ are obtained by interpolation [20].

Chapter 3

Level Set Method and Linear Bending Elasticity

3.1 Level Set Method

Description of a flag contour with a level set function is an easy way to implement the immersed interfaced method. Let us define a level set function ϕ whose zero level set with evolving front represents the flag contour ([14], [17]).

$$\phi(\mathbf{X}(s,t),t) = 0 \tag{3.1}$$

In our case, the closed curve of the zero level set function splits the $x-y$ plane into two region, where the level set has positive and negative values outside and inside, respectively. By the chain rule,

$$\phi_t + \mathbf{X}'(s,t) \cdot \nabla\phi(\mathbf{X}(s,t),t) = 0 \tag{3.2}$$

By the speed function V_n which supplies the speed in the outward normal direction, this yields an evolution equation

$$\phi_t + V_n|\nabla\phi| = 0 \tag{3.3}$$

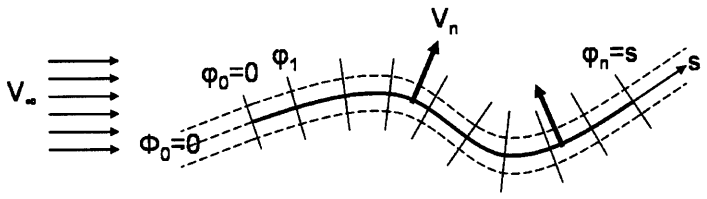


Figure 3-1: Level Set Representations in 2-dimensional space

where the outward normal unit vector $\mathbf{n} = \nabla\phi/|\nabla\phi|$ is used [16].

Now consider another global (not necessarily to be zero) level set function ψ such as

$$\psi(\mathbf{X}(s,t),t) = s \tag{3.4}$$

which indicates a coordinate along the material line s. The function ψ creates another level set which perpendicularly cross through the zero level set of ϕ. Paying attention to a particular flag point, s_o , defining the point as the origin (i.e. $s_o = s(x = 0) = 0$), yields a coordinate system which is satisfied along the interface. Thus, the (ϕ, ψ) creates the local coordinate system on the flag surface. Therefore, by the chain rule in terms of time, the tangential velocity to the zero level set function can be represented as

$$\psi_t + V_\tau|\nabla\psi| = 0 \tag{3.5}$$

where $\tau = \nabla\psi/|\nabla\psi|$ was used. The ψ plays a role as the velocity-correction in case that the material velocity, $\mathbf{V}$, is not perpendicular to the zero level set of ϕ. Even though s is a Lagrangian space along with the material curve, s can be described as a (reverse) function of x, y and t because $\{x, y\}$ are a function of s and t.

$$s = s(x, y, t) \tag{3.6}$$

Now we are ready to represent the elastic stress and acceleration of a flag in $x - y$ plane by employing ϕ and ψ. The details is shown in Chapter 4.

3.2 Description of the Bending Elasticity with the Level Set Functions

The goal of this section is to describe a new partial differential equation in terms of ϕ and ψ, which is equivalent to the two dimensional flapping equation (Eq. (2.1)).

3.2.1 Formulation of Bending Rigidity

In two dimensional problem, the unit normal and tangential vectors of the zero level set ϕ_0 can be expressed as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_x\mathbf{i} + \phi_y\mathbf{j}}{\sqrt{\phi_x^2 + \phi_y^2}} \tag{3.7}$$

$$\tau = \frac{\nabla\phi^T}{|\nabla\phi|} \tag{3.8}$$

where the orthogonal vector $\nabla\phi^T$ (i.e. $\nabla\phi\nabla\phi^T = 0$) was used. Note that the description of the tangential unit vector τ which is a function of ϕ instead of a function of ψ. As there are two possible tangential unit vectors which are orthogonal to the outward normal unit vector, we must choose one of them. In this work, the following vector was chosen.

$$\tau = \frac{-\phi_y\mathbf{i} + \phi_x\mathbf{j}}{\sqrt{\phi_x^2 + \phi_y^2}} \tag{3.9}$$

This a choice in which the curve is parameterized by s so that the interior is on the left in the direction of increasing s. As a result, it is obvious from the geometric relation that the $\partial\mathbf{X}/\partial s$ is nothing but the tangential unit vector τ. Let us rewrite the expression such as

$$\frac{\partial\mathbf{X}}{\partial s}(s,t) = \frac{-\phi_y(x(s,t),y(s,t),t)\mathbf{i} + \phi_x(x(s,t),y(s,t),t)\mathbf{j}}{\sqrt{\phi_x^2(x(s,t),y(s,t),t) + \phi_y^2(x(s,t),y(s,t),t)}} \tag{3.10}$$

and take partial derivatives three times in terms of s. Because the general form of the right-hand-side is a very complicated extension, let's define an operator $\mathbf{D}_{ssss}$ such that

$$\frac{\partial^4\mathbf{X}}{\partial s^4}(s,t) \Leftrightarrow \mathbf{D}_{ssss}[\phi(x,y,t)] \equiv f_1(\phi_x, \phi_y, \phi_{xx}, \phi_{yx}, \phi_{yy}, \phi_{xxx}, \cdots) \tag{3.11}$$

where the operator $\mathbf{D}_{ssss}$ is a function of ϕ_x, ϕ_y, ϕ_{xx}, ϕ_{yx}, ϕ_{yy}, ϕ_{xxx}, $\cdots$, etc. Calculating $\mathbf{D}_{ssss}[\phi(x,y,t)]$ requires a huge mathematical extension as well as a sig-

nificantly expensive computation cost. However, we not only can easily convert and automatically insert the final form into the calculation solver by using Mathematica software, but the complicated form will also be calculated only in the grids near the immersed interface, which does not bother calculation time much. See the Appendix for the final form of $\mathbf{D}_{ssss}[\phi(x,y,t)]$.

3.2.2 Formulation of Nonlinear Tension Term

We have left only with the second term of the flapping equation. Recall that

$$\frac{\partial \mathbf{X}}{\partial s}(s,t) = \frac{-\phi_y(x(s,t),y(s,t),t)\mathbf{i} + \phi_x(x(s,t),y(s,t),t)\mathbf{j}}{\sqrt{\phi_x^2(x(s,t),y(s,t),t) + \phi_y^2(x(s,t),y(s,t),t)}} \tag{3.12}$$

represents the unit vector in tangential direction, and then $\mathbf{X}_s \cdot \mathbf{X}_s$ becomes

$$\sqrt{\mathbf{X}_s \cdot \mathbf{X}_s} = 1 \tag{3.13}$$

Therefore, we do not have to take care about formulation of non-linear tension since it vanishes at least under cases where the fluid-structure interaction does not yield any stretching of interface over the simulation such as fluttering paper.

3.3 Verification of Formulation

In later part of this section, we will investigate whether the identical bending rigidity $\mathbf{D}_{ssss}$ have successfully formulated with Mathematica. If ϕ, ψ and $\mathbf{X}$ are given as a function of x, y and t, the $\mathbf{D}_{ssss}[\phi(x,y,t)]$ should correspond to the exact bending rigidity $\mathbf{X}_{ssss}$. Therefore, it is important to take proper level set functions which have an analytical expression of the position vector $\mathbf{X}$ with respect to s and t. Some of the easiest sets are used in the following subsections.

3.3.1 Test Example 1

Consider a circle level set function which grows with time, i.e.

$$\phi(x(s,t),\ y(s,t),t) = x^2 + y^2 - t^3 \tag{3.14}$$

The position vector that makes this level set zero is given by

$$\mathbf{X}(s,t) = \{x(s,t),\ y(s,t)\} = \{t^{3/2}cos(s/t^{3/2}),\ t^{3/2}sin(s/t^{3/2})\} \tag{3.15}$$

where s represents the distance along the zero level set, which starts from $\{t^{3/2}, 0\}$ in counterclockwise direction. This level set function is a proper choice because we have the analytical ψ function which is expressed as

$$\psi(x(s,t),\ y(s,t),t) = t^{3/2}arctan(y/x) \tag{3.16}$$

The identical partial derivatives can be calculated since ϕ and ψ are analytically known. Note that we should restrict the evaluation range within $s \in [0,\ \pi/2)$ to avoid singular points where the *arctan* values can be discontinuous and the numerical values jump when $y/x \to \infty$. The L_∞-norm of the errors between the directly calculated values and the identical PDEs shown in Appendix certificates an excellent match of the two approach, as shown in Table 1.

3.3.2 Test Example 2

Consider a non-stretchable level set function such as

$$\phi(x(s,t), y(s,t), t) = (x - t^2)^2 + (y - t^3)^2 - 4 \tag{3.17}$$

, which is a circle with radius of 2 and propagating along with $y = x^{3/2}$. The location vector which makes the level set zero is

$$\{x(s,t), y(s,t)\} = \{t^2 + 2cos(s/2), t^3 + 2sin(s/2)\} \tag{3.18}$$

where s is also defined as the distance, starting from $\{t^2 + 2,\ t^3\}$to the counter-clockwise direction along the zero level set. ψ is given by $s = s(x,y,t)$, i.e.

Table 3.1: Comparison of Direct Derivation and via New PDE

Infinity Norms	Test Example 1	Test Example 2
$\|\|\mathbf{X}_{ssss} - \mathbf{D}_{ssss}\|\|_{L_\infty}(x\ component)$	5.55112E-16	9.71445E-17
$\|\|\mathbf{X}_{ssss} - \mathbf{D}_{ssss}\|\|_{L_\infty}(y\ component)$	5.55112E-16	9.71445E-17

$$\psi(x(s,t), y(s,t), t) = 2arctan((y - t^3)/(x - t^2)) \tag{3.19}$$

As in Test Example 1, the identical PDE expression can be verified by plugging the analytical location vector, restricting the evaluation range to avoid singular points. The infinity norms of the error of each component are shown in Table 1. This proves that the identical PDE expressions are valid regardless of the stretchability of an interface. Note that the non-stretchable assumption is imposed in order to make the second term of structural equation zero.

Chapter 4

Numerical Techniques

In this section, sorts of numerical schemes are shown for Navier-Stoke's equations, approximation of bending elasticity, transport of level set functions, jump conditions and the Ghost Fluid Method (GFM) for the sake of usage of the finite difference method. For the simulation, the staggered grid was used, where the evaluation locations of pressure and velocities are shown in the following figure.

4.1 Discretization of the Incompressible Navier-Stokes' Equations

The incompressible Navier-Stokes' equations in two space dimensions are given by [15], which are

$$u_t = -\frac{1}{\rho}p_x - (u^2)_x - (uv)_y + \nu(u_{xx} + u_{yy}) \tag{4.1}$$

$$v_t = -\frac{1}{\rho}p_y - (uv)_x - (v^2)_y + \nu(v_{xx} + v_{yy}) \tag{4.2}$$

$$u_x + v_y = 0 \tag{4.3}$$

When the solutions u^n, v^n and p^n are known at time step n, we obtain the solutions

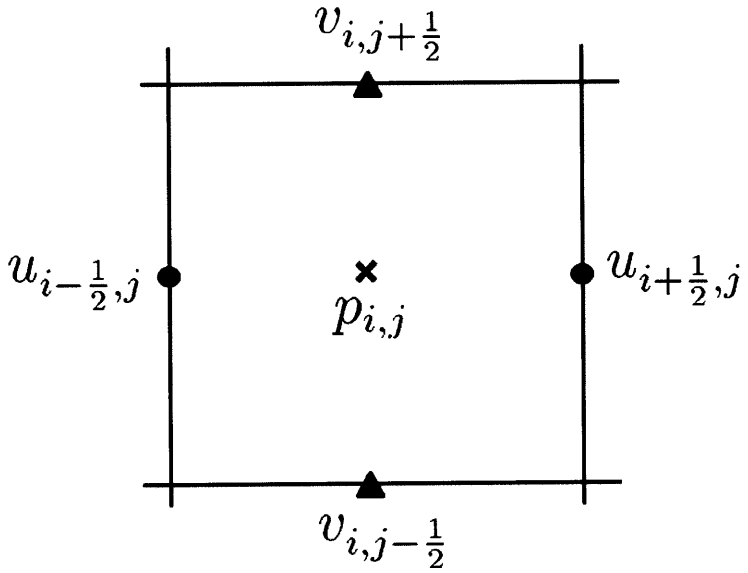


Figure 4-1: MAC grid

at time step n+1 by the following three step method.

Firstly, the nonlinear convection terms are solve by

$$\frac{u^* - u^n}{\Delta t} = -((u^n)^2)_x - (u^n v^n)_y + C_1^u \tag{4.4}$$

$$\frac{v^* - v^n}{\Delta t} = -((u^n v^n))_x - ((v^n)^2)_y + C_1^v \tag{4.5}$$

where the C_1^u and C_1^v are correction term from jump contribution, which is nonzero only the derivatives include irregular points. Jump contribution will be mentioned in Section 4.2.

Next, we can solve the viscosity terms implicitly as shown in [15] or explicitly. The explicit way is

$$\frac{u^{**} - u^*}{\Delta t} = \nu(u_{xx}^* + u_{yy}^*) + C_2^u \tag{4.6}$$

$$\frac{v^{**} - v^*}{\Delta t} = \nu(v_{xx}^* + v_{yy}^*) + C_2^v \tag{4.7}$$

, while the implicit way is

$$\frac{u^{**} - u^*}{\Delta t} = \nu(u_{xx}^{**} + u_{yy}^{**}) + C_2^u \tag{4.8}$$

$$\frac{v^{**} - v^*}{\Delta t} = \nu(v_{xx}^{**} + v_{yy}^{**}) + C_2^v \tag{4.9}$$

Note again that C_2^u and C_2^v are jump contributions.

The implicit approach yields a SPD(symmetric positive definite) linear system, while it does not allow us to include viscous jumps directly. To solve the inverse of the coefficient matrix of the linear system, we use the sparse Cholesky decomposition. On the other hand, we can easily apply no-slip boundary condition or stress jump with the explicit way, while it requires more restricted CFL condition which is proportional to the space interval squared. Both method will be used in this work following the

objectives of simulation.

Finally, the pressure correction is required to meet the incompressibility. Implicit pressure gradient is applied to solve

$$\frac{u^{n+1} - u^{**}}{\Delta t} = -\frac{1}{\rho}(p^{n+1})_x + C_3^u \tag{4.10}$$

$$\frac{v^{n+1} - v^{**}}{\Delta t} = -\frac{1}{\rho}(p^{n+1})_y + C_3^v \tag{4.11}$$

where the pressure p^{n+1} is from the conservation restriction, i.e,

$$-\nabla^2 p^{n+1} = -\frac{\rho}{\Delta t}\nabla \cdot \mathbf{u} + C_1^p + C_4^u + C_4^v \tag{4.12}$$

4.1.1 Second-Order Finite Difference Approximation

For the spatial discretization, we use standard central finite difference schemes. For the first derivatives, we use

$$(u_x)_{i+\frac{1}{2},j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x^2) \tag{4.13}$$

and for the second derivatives we use

$$(u_{xx})_{i,j} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + O(\Delta x^2) \tag{4.14}$$

Same schemes are used for the discretization of v and p as well. These schemes brings the differentiated values onto the correct locations on staggered grid except the step to solve non-linear convection term. In order to correct the locations of the convection term, we can use average between nearby points.

4.1.2 Transition to Upwinding for the Non-linear Convection Terms

The solution of non-linear convection step is not stable with the central differencing schemes when we use fast flows or large time step. Therefore, we take a linear combination between the central differencing and upwinding for the discretization of convection terms. In order to meet the stability condition of this hyperbolic behavior, we introduce a smoothing transition factor such as

$$\gamma = min(1.2\Delta t \cdot max(max_{i,j}|u_{i,j}|,\ max_{i,j}|v_{i,j}|),\ 1) \tag{4.15}$$

to apply

$$\frac{u^* - u^n}{\Delta t} = -\left((\bar{u}^h)^2 - \gamma|\bar{u}^h|\tilde{u}^h\right)_x^n - \left((\bar{u}^v\bar{v}^h) - \gamma|\bar{v}^h|\tilde{u}^v\right)_y^n + C_1^u \tag{4.16}$$

$$\frac{v^* - v^n}{\Delta t} = -\left((\bar{u}^v\bar{v}^h) - \gamma|\bar{u}^v|\tilde{v}^h\right)_x^n - \left((\bar{v}^v)^2 - \gamma|\bar{v}^v|\tilde{v}^v\right)_y^n + C_1^v \tag{4.17}$$

where $\bar{u}^h$ and $\bar{v}^h$ are the horizontally averaged value, $\bar{u}^v$ and $\bar{v}^v$ are the vertically averaged value, i.e.,

$$\bar{u}^h_{i,j} = \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2}, \qquad \bar{v}^h_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{v_{i,j+1/2} + v_{i-1,j+1/2}}{2} \tag{4.18}$$

$$\bar{u}^v_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{u_{i-1/2,j+1} + u_{i-1/2,j}}{2}, \qquad \bar{v}^v_{i,j} = \frac{v_{i,j+1/2} + v_{i,j-1/2}}{2} \tag{4.19}$$

and $\tilde{u}^h$, $\tilde{v}^h$, $\tilde{u}^v$ and $\tilde{v}^v$ are defined as

$$\tilde{u}^h_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{2}, \qquad \tilde{v}^h_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{v_{i,j+1/2} - v_{i-1,j+1/2}}{2} \tag{4.20}$$

$$\tilde{u}^v_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{u_{i-1/2,j+1} - u_{i-1/2,j}}{2}, \qquad \tilde{v}^v_{i,j} = \frac{v_{i,j+1/2} - v_{i,j-1/2}}{2} \tag{4.21}$$

This is to match the u^* and v^* locations onto the exact u and v locations in MAC

grid. This transition is the central differencing when $\gamma = 0$ and the upwinding when $\gamma = 1$([15]). The factor of 1.2 in gamma is from the empirical value, which gives a higher accuracy by tilting towards upwinding ([11]).

4.2 Discontinuities across the Interface

4.2.1 Generalized Central Finite Difference Formulas

In order to handle discontinuities of the solution of the Navier-Stokes' equations, we modify the standard discretization schemes only on grid points near the interface. As in the IIM, the generalized finite difference schemes can be employed to include the discontinuities. We assume that the discontinuity occurs at most once between two nearby grid points.

The highest order of a finite difference scheme for $v^{(n)}(x)$ with a discontinuity point at $x = \xi$ is $m - n + 1$, where m is the maximum number such that jump conditions $[v^{(l)}(\xi)]$ ($l = 0, 1, 2, \cdots, m$) are known (see [20]). Since we know jump conditions up to second order derivatives, the finite difference schemes which include jump conditions have second order for first derivatives and first order for second derivatives. Let's consider a case where the interface cuts the grid line between two grid points at $x = \xi$, $x_i < \xi < x_{i+1}$, $x_i \in \Omega^-$, $x_{i+1} \in \Omega^+$ as shown in Figure 4.2 (a). From Taylor series expansions, the following schemes hold for at least twice differentiable function $v(x)$:

$$v_x(x_i) = \frac{v_{i+1} - v_{i-1}}{2h} - \frac{1}{2h}\Sigma_{m=0}^{2}\frac{(h^+)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h^2) \tag{4.22}$$

$$v_x(x_{i+1}) = \frac{v_{i+2} - v_i}{2h} - \frac{1}{2h}\Sigma_{m=0}^{2}\frac{(h^-)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h^2) \tag{4.23}$$

$$v_{xx}(x_i) = \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} - \frac{1}{h^2}\Sigma_{m=0}^{2}\frac{(h^+)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h) \tag{4.24}$$

$$v_{xx}(x_{i+1}) = \frac{v_{i+2} - 2v_{i+1} + v_i}{h^2} + \frac{1}{h^2}\Sigma_{m=0}^{2}\frac{(h^-)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h) \tag{4.25}$$

where $h^+ = x_{i+1} - \xi$ (> 0), $h^- = x_i - \xi$ (< 0), and h is spacing interval in x direction. If x_i and x_{i+1} lie in Ω^+ and Ω^- respectively as shown in Figure 4.2 (b), the sign of the second term in the right hand side of the above equations must be modified.

Moreover, we define the h^+ and h^- using the level set function $\phi(x)$

$$h^+ = \frac{|\phi_{i+1}|h}{|\phi_i| + |\phi_{i+1}|}, \qquad h^- = -\frac{|\phi_i|h}{|\phi_i| + |\phi_{i+1}|} \tag{4.26}$$

and the Equation (4.22)-(4.25) can be compacted independent of the sign as followings:

$$v_x(x_i) = \hat{v}_{x,i}^{s.c.d} + sign(\phi_i)\frac{1}{2h}\Sigma_{m=0}^{2}\frac{(h^+)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h^2) \tag{4.27}$$

$$v_x(x_{i+1}) = \hat{v}_{x,i+1}^{s.c.d} - sign(\phi_{i+1})\frac{1}{2h}\Sigma_{m=0}^{2}\frac{(h^-)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h^2) \tag{4.28}$$

$$v_{xx}(x_i) = \hat{v}_{xx,i}^{s.c.d} + sign(\phi_i)\frac{1}{h^2}\Sigma_{m=0}^{2}\frac{(h^+)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h) \tag{4.29}$$

$$v_{xx}(x_{i+1}) = \hat{v}_{xx,i+1}^{s.c.d} + sign(\phi_{i+1})\frac{1}{h^2}\Sigma_{m=0}^{2}\frac{(h^-)^m}{m!}\left[\frac{\partial^m v(\xi)}{\partial x^m}\right] + O(h) \tag{4.30}$$

where the superscript s.c.d in $\hat{v}_x^{s.c.d}$ denotes the standard central difference scheme, and therefore the second terms in above expressions are the correction terms for discontinuities [6].

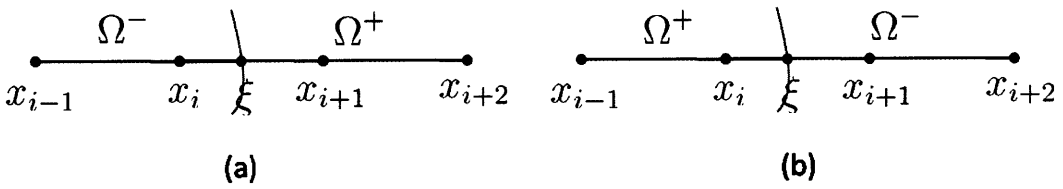


Figure 4-2: Cases of interface location. (a) $x_i \in \Omega^-$, $x_{i+1} \in \Omega^+$, (b) $x_i \in \Omega^+$, $x_{i+1} \in \Omega^-$

4.2.2 Jump Contributions

We discretize the Navier-Stokes' equations with standard central difference scheme except for the irregular grid points. To correct derivatives in irregular points, we shall add correction terms $\{C_i^u,\ C_j^v,\ C_1^p |\ i,\ j = 1, 2, 3\}$ which is nonzero only when irregular points should be taken care for spatial derivatives. For reduce the amount of pages, we evaluate only x component, and y component can be derived in the same way.

- Jump contribution associated with the convection term : C_1^u and C_1^v

  In order to compute the convection term, we calculate the averaged and differenced values as shown in Section 4.1.2. However, $\bar{u}$ is not just average of two quantities if the interface cuts the grid line of the two corresponding points. As seen in Figure 4-3 (a), consider a case where the interface ($x = d$) lies between a and b. And suppose that the point b is in Ω^+. Then the interpolated average value $u(b)$ is

$$\bar{u}_b^+ = \frac{u_a^+ + u_c^+}{2} \tag{4.31}$$

  where the superscript $+$ denotes the value evaluated at Ω^+ region. Hence u_a^+ is a ghost quantity and its values is determined by Taylor series expansion at $x = d$, i.e.,

$$u_a^+ = u_d^+ + h^- \left(\frac{du_b}{dx}\right)^+ + \frac{(h^-)^2}{2}\left(\frac{d^2u_b}{dx^2}\right)^+ + O(h^2) \tag{4.32}$$

  Since jump conditions are known up to second derivatives, we can replace the derivatives in above equation by the jump relations.

$$\mathrm{u}_a^+ = (u_d^- + [u_d]) + h^- \left(\left(\tfrac{du_b}{dx}\right)^- + \left[\tfrac{du_b}{dx}\right]\right) + \tfrac{(h^-)^2}{2}\left(\left(\tfrac{d^2u_b}{dx^2}\right)^- + \left[\tfrac{d^2u_b}{dx^2}\right]\right) + O(h^2)$$

$$= u_a^- + [u_d] + h^- \left[\frac{du_b}{dx}\right] + \frac{(h^-)^2}{2}\left[\frac{d^2u_b}{dx^2}\right] + O(h^2) \tag{4.33}$$

Therefore, the average is

$$\bar{u}_b^+ = \frac{u_a^- + u_c^+}{2} + \frac{1}{2}[u_d] + \frac{1}{2}h^- \left[\frac{du_b}{dx}\right] + \frac{(h^-)^2}{4}\left[\frac{d^2u_b}{dx^2}\right] + O(h^2) \tag{4.34}$$

and if point b lies in Ω^- the interpolated average at $x = b$ is

$$\bar{u}_b^- = \frac{u_a^+ + u_c^-}{2} - \frac{1}{2}[u_d] - \frac{1}{2}h^- \left[\frac{du_b}{dx}\right] - \frac{(h^-)^2}{4}\left[\frac{d^2u_b}{dx^2}\right] + O(h^2) \tag{4.35}$$

A compact expression for this case is

$$\bar{u}_b = \frac{u_a + u_c}{2} + sign(\phi(x_b))\left(\frac{1}{2}[u_d] + \frac{1}{2}h^- \left[\frac{du_b}{dx}\right] + \frac{(h^-)^2}{4}\left[\frac{d^2u_b}{dx^2}\right]\right) + O(h^2) \tag{4.36}$$

Next, when point d falls between b and c as shown in Figure 4-3 (b), same analysis gives the following interpolation scheme:

$$\bar{u}_b = \frac{u_a + u_c}{2} + sign(\phi(x_b))\left(\frac{1}{2}[u_d] + \frac{1}{2}h^+ \left[\frac{du_b}{dx}\right] + \frac{(h^+)^2}{4}\left[\frac{d^2u_b}{dx^2}\right]\right) + O(h^2) \tag{4.37}$$

Therefore, we can evaluate the average of velocity even when the quantities lie on different region across the interface.

In addition to the average, the interpolation of $\tilde{u}$ is required as well. Using the similar way with Taylor series expansion, we can obtain the differenced value $\tilde{u}_b$. If point d falls between a and b,

$$\tilde{u}_b = \frac{u_c - u_a}{2} - sign(\phi(x_b))\left(\frac{1}{2}[u_d] + \frac{1}{2}h^- \left[\frac{du_b}{dx}\right] + \frac{(h^-)^2}{4}\left[\frac{d^2u_b}{dx^2}\right]\right) + O(h^2) \tag{4.38}$$

and if point d falls between b and c,

$$\tilde{u}_b = \frac{u_c - u_a}{2} + sign(\phi(x_b)) \left(\frac{1}{2}[u_d] + \frac{1}{2} h^+ \left[\frac{du_b}{dx} \right] + \frac{(h^+)^2}{4} \left[\frac{d^2 u_b}{dx^2} \right] \right) + O(h^2) \tag{4.39}$$

Using the accurately averaged and differenced values $\bar{u}$ and $\tilde{u}$, the convection term can be computed. The jump contributions associated with convection (C_1^u and C_1^v) are not represented in here.

- Jump contribution associated with the diffusion term : C_2^u and C_2^v

  The jump contribution induced by the diffusion term appears in generalized central difference scheme. When the interface cuts the grid line at $x = \xi$ ($x_i < \xi < x_{i+1}$), the second derivatives at x_i and x_{i+1} should be modified. Thus, the correction terms associated with $\nu\, u^*_{xx}$ are

$$(C_2^{u_{xx}})_{i,j} = sign(\phi_{i,j}) \frac{\nu}{h^2} \Sigma_{m=0}^{2} \frac{(h_x^+)^m}{m!} \left[\frac{\partial^m u(\xi)}{\partial x^m} \right] \tag{4.40}$$

$$(C_2^{u_{xx}})_{i+1,j} = sign(\phi_{i+1,j}) \frac{\nu}{h^2} \Sigma_{m=0}^{2} \frac{(h_x^-)^m}{m!} \left[\frac{\partial^m u(\xi)}{\partial x^m} \right] \tag{4.41}$$

  and if the interface cuts the grid line at $y = \eta$ ($y_j < \eta < y_{j+1}$) the correction terms associated with νu^*_{yy} are

$$(C_2^{u_{yy}})_{i,j} = sign(\phi_{i,j}) \frac{\nu}{h^2} \Sigma_{m=0}^{2} \frac{(h_y^+)^m}{m!} \left[\frac{\partial^m u(\eta)}{\partial y^m} \right] \tag{4.42}$$

$$(C_2^{u_{yy}})_{i,j+1} = sign(\phi_{i,j+1}) \frac{\nu}{h^2} \Sigma_{m=0}^{2} \frac{(h_y^-)^m}{m!} \left[\frac{\partial^m u(\eta)}{\partial y^m} \right] \tag{4.43}$$

  , hence the sum of these are the total correction in terms of x component of diffusion term.

- Jump in pressure and pressure gradient : C_1^p

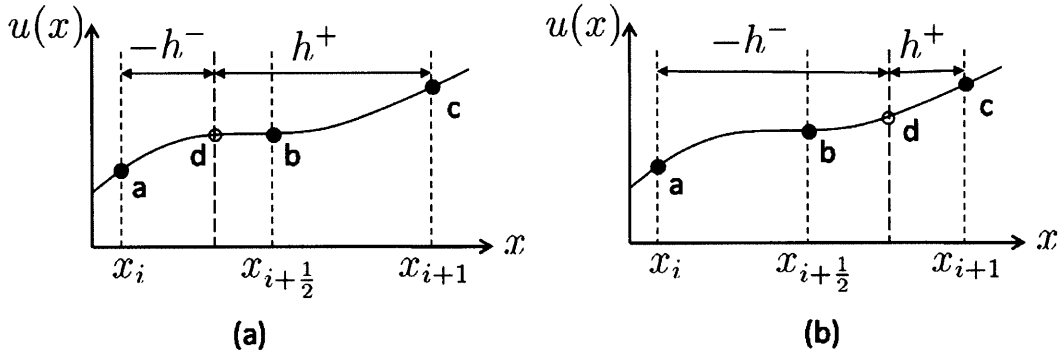


Figure 4-3: Interpolation to obtain average and differenced values.

Recall that the force density and its derivatives along the interface are related to the pressure jump and jump in pressure gradient [21]. In the multi-step method, we solve a Poisson equation to obtain pressure. Once jump condition is known, jumps in pressure or pressure gradient can be easily applied only by the Ghost Fluid Method (GFM) and changing the right-hand side of the linear system ([21]). Using the central differencing scheme, the pressure correction step in our work

$$\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u} \tag{4.44}$$

yields a linear system, i.e.,

$$\underline{A}\underline{p} = \underline{f} \tag{4.45}$$

where $\underline{A} \in R^{n_x n_y \times n_x n_y}$ is the coefficient matrix, $\underline{p} \in R^{n_x n_y \times 1}$ is unknown and $\underline{u} \in R^{n_x n_y \times 1}$ is source matrix from the right-hand side of the Poisson equation. For the simplicity, let's consider one dimension case ($p_{xx} = f$). For example, assume that the interface exists between a particular j and $j+1$ location, and the jumps in pressure is given as

$$[p]_{\partial\Omega} = a_\Gamma \tag{4.46}$$

And suppose that p_{j+1} exists in Ω^+ and p_j exists in Ω^- region and the interface $\partial\Omega$ is defined as a zero level set which values exist on the same location as the level set. Using the linear interpolation, the jump of pressure on the interface becomes

$$[p]_{\partial\Omega} \equiv a_\Gamma = \frac{a_{\Gamma,j}|\phi_{j+1}| + a_{\Gamma,j+1}|\phi_j|}{|\phi_j| + |\phi_{j+1}|} \tag{4.47}$$

Then discretized froms at j and $j+1$ locations are

$$\left(\frac{p_{j+1}^{+}-p_{j}^{-}}{\Delta x}-\frac{p_{j}^{-}-p_{j-1}^{-}}{\Delta x}\right)/\Delta x=f_{j} \tag{4.48}$$

$$\left(\frac{p_{j+2}^{+}-p_{j+1}^{+}}{\Delta x}-\frac{p_{j+1}^{+}-p_{j}^{-}}{\Delta x}\right)/\Delta x=f_{j+1} \tag{4.49}$$

Plugging the jumps into the above equations gives

$$\left(\frac{p_{j+1}^{-}-p_{j}^{-}}{\Delta x}-\frac{p_{j}^{-}-p_{j-1}^{-}}{\Delta x}\right)/\Delta x=f_{j}+\frac{a_{\Gamma}}{\Delta x^{2}} \tag{4.50}$$

$$\left(\frac{p_{j+2}^{+}-p_{j+1}^{+}}{\Delta x}-\frac{p_{j+1}^{+}-p_{j}^{+}}{\Delta x}\right)/\Delta x=f_{j+1}-\frac{a_{\Gamma}}{\Delta x^{2}} \tag{4.51}$$

in which we can see that what we have to do is to update right-hand side only. Now consider a jump of normal derivative of pressure, i.e.,

$$\left[\frac{dp}{dx}\right]_{\partial\Omega}=b_{\Gamma} \tag{4.52}$$

Using the subcell resolution

$$\theta=-h^{-}/h=\frac{|\phi_{j}|}{|\phi_{j}|+|\phi_{j+1}|} \tag{4.53}$$

and denoting the pressure value at interface p_I, the jump conditions becomes

$$\left(\frac{p_{j+1}-p_{I}}{(1-\theta)\Delta x}\right)-\left(\frac{p_{I}-p_{j}}{\theta\Delta x}\right)=b_{\Gamma} \tag{4.54}$$

where the first and second terms are the derivatives of left and right side respectively. Thus, pressure at the interface is

$$p_{I}=p_{j+1}\theta+p_{j}(1-\theta)-b_{\Gamma}\theta(1-\theta)\Delta x \tag{4.55}$$

and the derivatives on the left and right sides of the interface can be written as

$$\frac{p_I - p_j}{\theta \Delta x} = \frac{p_{j+1} - p_j}{\Delta x} - b_\Gamma (1 - \theta) \tag{4.56}$$

$$\frac{p_{j+1} - p_I}{(1 - \theta) \Delta x} = \frac{p_{j+1} - p_j}{\Delta x} + b_\Gamma \theta \tag{4.57}$$

Therefore, this approximation allows one to update the right hand side of the linear system, i.e.,

$$p_{xx,j} = f_j + \frac{a_\Gamma}{(\Delta x)^2} + \frac{b_\Gamma (1 - \theta)}{\Delta x} \tag{4.58}$$

$$p_{xx,j+1} = f_{j+1} - \frac{a_\Gamma}{(\Delta x)^2} + \frac{b_\Gamma \theta}{\Delta x} \tag{4.59}$$

This idea can be extended to multi-dimensional, multi-phase problems [21].

For example, we represent two dimensional calculations. For the given level set function such as

$$\phi(x, y) = (x - 0.5)^2 + (y - 0.5)^2 - 0.3^2 \tag{4.60}$$

whose zero level set is a circle with radius of 0.3, we apply a jump condition

$$a_\Gamma = x^2 + y + 1 \tag{4.61}$$

Similarly, with the same level set function and jump condition of

$$a_\Gamma = sin(10x) + y + 1 \tag{4.62}$$

are applied. The following figure represents the solutions of pressure field.

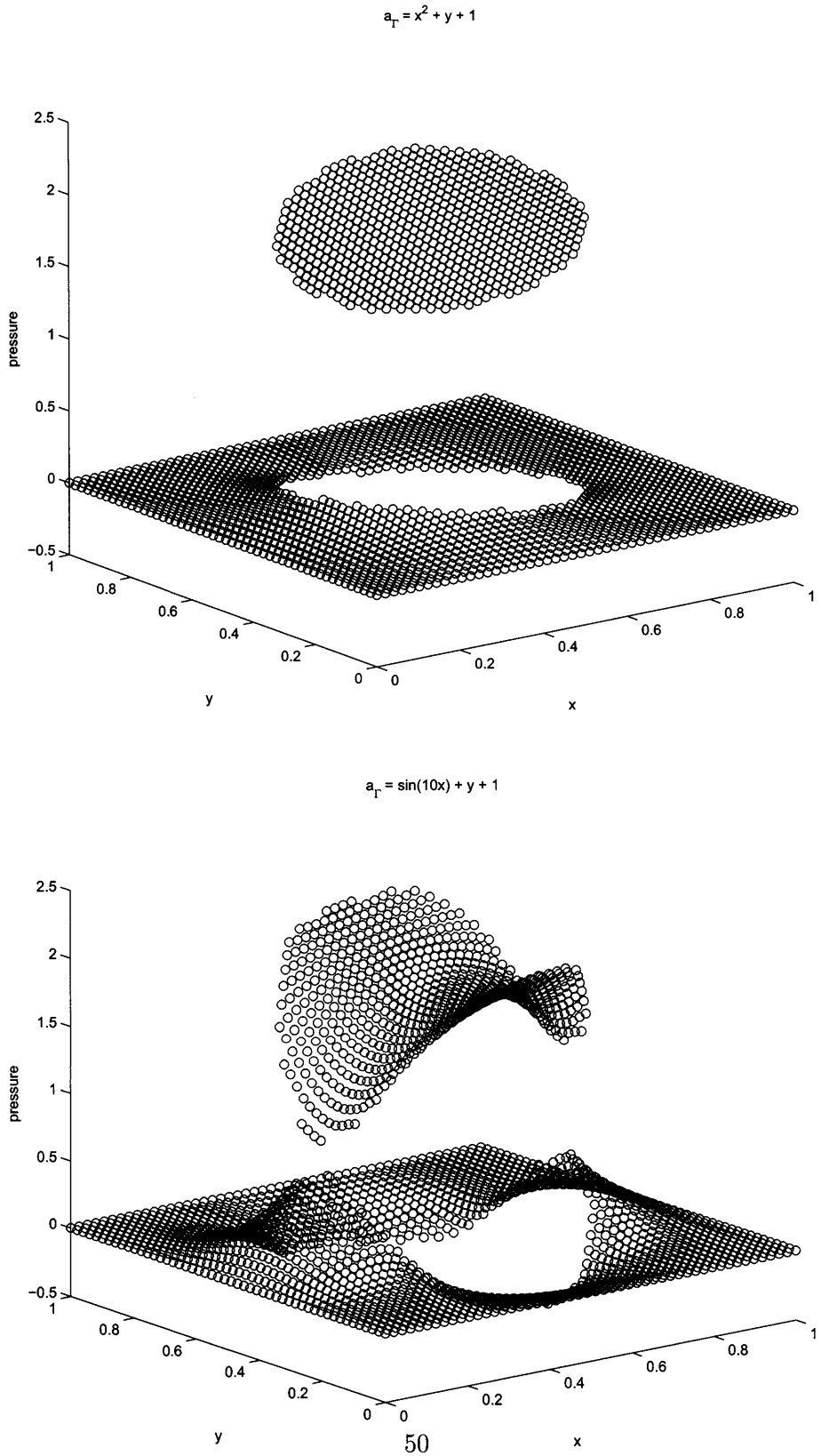


Figure 4-4: 2D Pressure Jump across a Closed Interface. (a) case 1 : $a_\Gamma = x^2 + y + 1$ (b) case 2 : $a_\Gamma = sin(10x) + y + 1$

- Correction of Velocity : C_3^u and C_3^v

  Since the pressure is discontinuous across the interface, the derivative of pressure in corrector step overshoots. Therefore, we have to eliminate the effect of jump of pressure so that the fluid flows along independent pressure gradients in each domain ($\Omega^{\pm}$). We have to add velocity correction term $\{uc_k\}$, i.e.,

$$u_{j+1/2}^{n+1} = u_{j+1/2}^{**} - \frac{\Delta t}{\rho}(p_{j+1}^{n+1}|_{\Omega^+} - p_j^{n+1}|_{\Omega^-}) + \frac{\Delta t}{\rho}(C_3^u)_{j+1/2} \tag{4.63}$$

  where $(C_3^u)_{j+1/2}$ is defined as

$$(C_3^u)_{j+1/2} = sign(\phi_j)a_\Gamma/\Delta x \tag{4.64}$$

  Here, we present several examples of jump of pressure. Firstly, in one dimensional Poisson equation the following figure shows the solution of Equation (4.45) and the derivatives before and after velocity correction. Dirichlet boundary condition is used for this.

- Jump contribution associated with $\frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}*$: C_4^u, C_4^v

  In the MAC grid, the values of $(\rho/\Delta t)\nabla \cdot \mathbf{u}*$ lies on the center of the cell, i.e. locations for pressure. Therefore, interpolation of velocity location is not necessary. Assume that the interface falls between x_i and x_{i+1}. If the interface is closer to x_{i+1} than x_i, the jump contribution of u_x is

$$C_4^u = sign(\phi_i)\frac{1}{h}\left([u_x]_\Gamma(h^+) + \frac{1}{2}[u_{xx}]_\Gamma(h^+)^2\right) \tag{4.65}$$

  and if the interface is closer to x_i then

$$C_4^u = -sign(\phi_{i+1})\frac{1}{h}\left([u_x]_\Gamma(h^-) + \frac{1}{2}[u_{xx}]_\Gamma(h^-)^2\right) \tag{4.66}$$

  where the h^+ and h^- are defined as

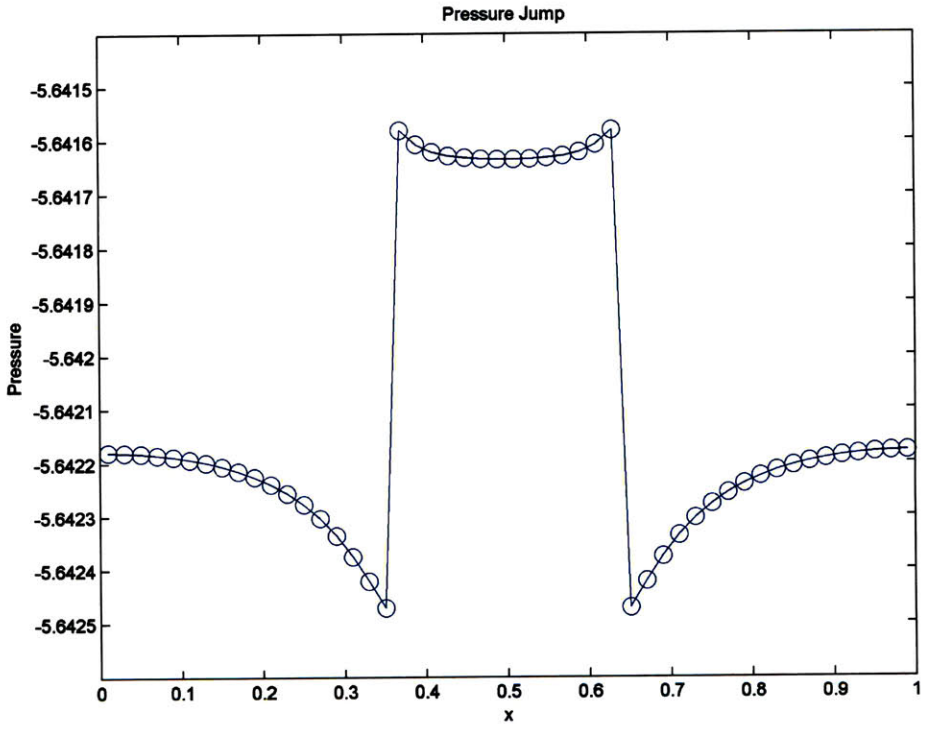


Figure 4-5: 1D Pressure Jump

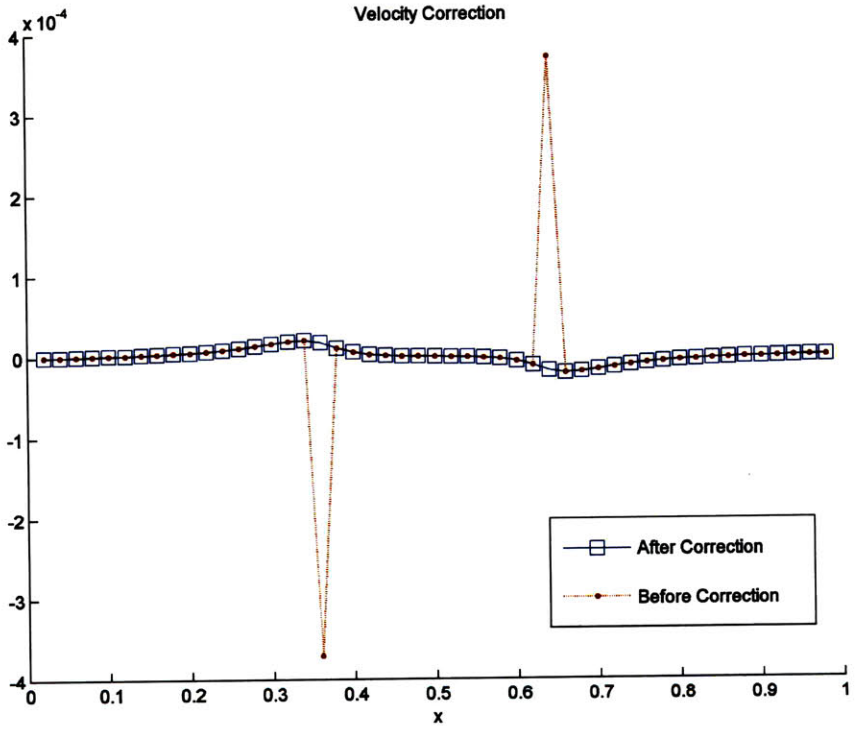


Figure 4-6: Velocity Correction.

$$h^+ = \frac{|\phi_{i+1}|h}{|\phi_i| + |\phi_{i+1}|}, \qquad h^- = -\frac{|\phi_i|h}{|\phi_i| + |\phi_{i+1}|} \tag{4.67}$$

In the same way, C_4^v can be determined.

4.3 No-Slip Boundary Condition

No-slip boundary condition can be imposed by adding tangential singular force f_τ which is related to the normal derivative of the velocity as shown in [18]. Since the tangential force comes from the viscous shear stress on the interface surface, its relation can be represented as

$$f_\tau = -\nu \left[\tau \cdot \frac{\partial \mathbf{u}}{\partial n}\right] = -\nu \left\{ \left(\tau \cdot \frac{\partial \mathbf{u}}{\partial n}\right)_{\Omega^+} - \left(\tau \cdot \frac{\partial \mathbf{u}}{\partial n}\right)_{\Omega^-} \right\} \tag{4.68}$$

The right hand side denotes the jump of the tangential component of the normal derivative of the fluid velocity.

The normal derivative of the velocity can be calculated with one-sided finite difference scheme shown in Figure 4-7. For example, the normal derivative of $\mathbf{u}$ in Ω^+ region is given as following:

$$\left(\frac{\partial \mathbf{u}}{\partial n}\right)_{\Omega^+} = \frac{-3\mathbf{u}_0 + 4\mathbf{u}_1 - \mathbf{u}_2}{2\delta l} + O(\delta l^2) \tag{4.69}$$

where δl is the distance between two adjacent points on the stencil, and typically $\delta l \geq \sqrt{\Delta x^2 + \Delta y^2}$ to make the evaluation points exist in different Cartesian-grid cells. The velocity $\mathbf{u}_1$, $\mathbf{u}_2$ are interpolated values from the surrounding velocity field. For $\mathbf{u}_0$, we have to give special treatment to make sure no-slip condition. Since the interface moves in normal direction when solving $\phi_t + \mathbf{u} \cdot \nabla\phi = 0$, fluid velocity on the interface surface must be the same with the interface velocity,

$$\mathbf{u}_0 = (\mathbf{u}|_\Gamma \cdot \mathbf{n})\mathbf{n} \tag{4.70}$$

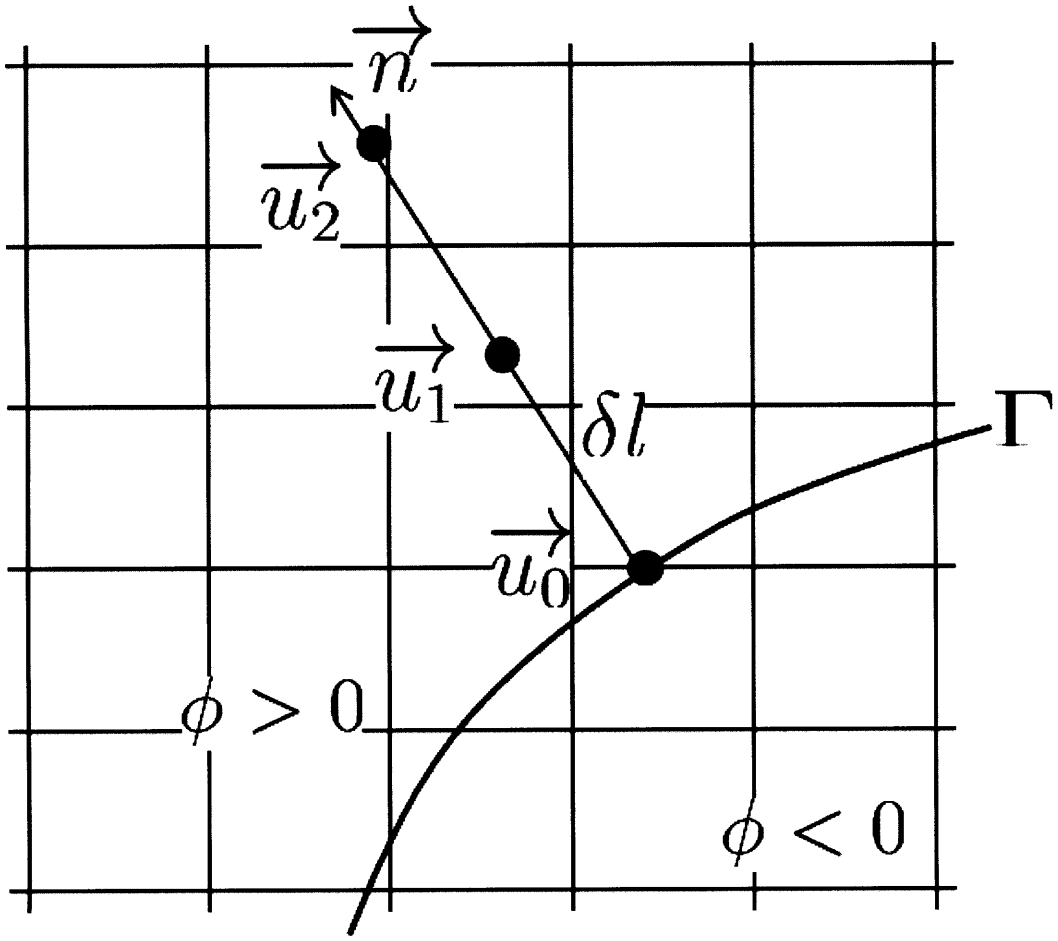


Figure 4-7: One-sided extrapolation using one-sided finite difference scheme

4.4 Finite Difference Approximation of Bending Elasticity

In this section, the capability of benchmark is shown between exact solutions and finite difference approximations for $\mathbf{D}_{tt}[\phi(x,y,t),\psi(x,y,t)]$ and $\mathbf{D}_{ssss}[\phi(x,y,t)]$. For the properly chosen $\phi(\mathbf{X},t)$ and $\psi(\mathbf{X},t)$, the exact solution can be obtained by plugging the exact location vector $\mathbf{X}$ into the $\mathbf{D}_{tt}[\phi(x,y,t),\psi(x,y,t)]$ and $\mathbf{D}_{ssss}[\phi(x,y,t)]$ that are a combination of spatial and temporal derivatives of $\phi(\mathbf{X},t)$ and $\psi(\mathbf{X},t)$. Therefore, it is important to select $\phi(\mathbf{X},t)$ and $\psi(\mathbf{X},t)$ which have an analytical location vector. On the other hand, for the approximations, the standard second-order central finite difference schemes are used over a structured grid.

4.4.1 Matching of Bending Stress : $\frac{\partial^4 \mathbf{X}}{\partial s^4}$

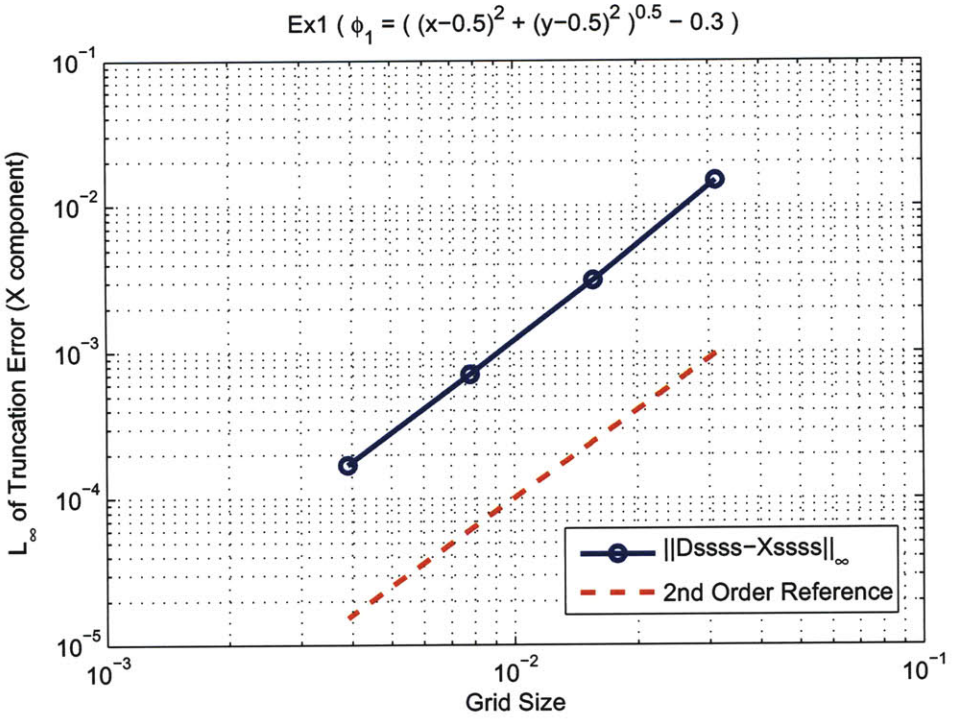
Basically, the level set function ϕ must be at least or more than C^4 if we want to estimate the convergence of $\mathbf{D}_{ssss}[\phi(x,y,t)]$ which contains fourth spatial derivatives. Two kinds of level set functions are investigated here; one is a fully C^∞ function such as

$$\phi_1(x,y) = \sqrt{(x-0.5)^2+(y-0.5)^2} - 0.3, \qquad (x,y) \in [0,1] \tag{4.71}$$

and the other one is C^∞ only in one dimension, i.e,

$$\phi_2(x,y) = xsin(x) - y, \qquad (x,y) \in [0,1] \tag{4.72}$$

where we dropped time variable for simplicity. The convergence calculations with different grid size are shown in Figure 4-8 and 4-9. As seen in the figures., the finite difference approximation for the elastic stress produced successfully for fully and semi C^∞ level set functions. The calculation results for different grid size are included in Figure 4-8 and 4-9., which represents an excellent convergence into second order.



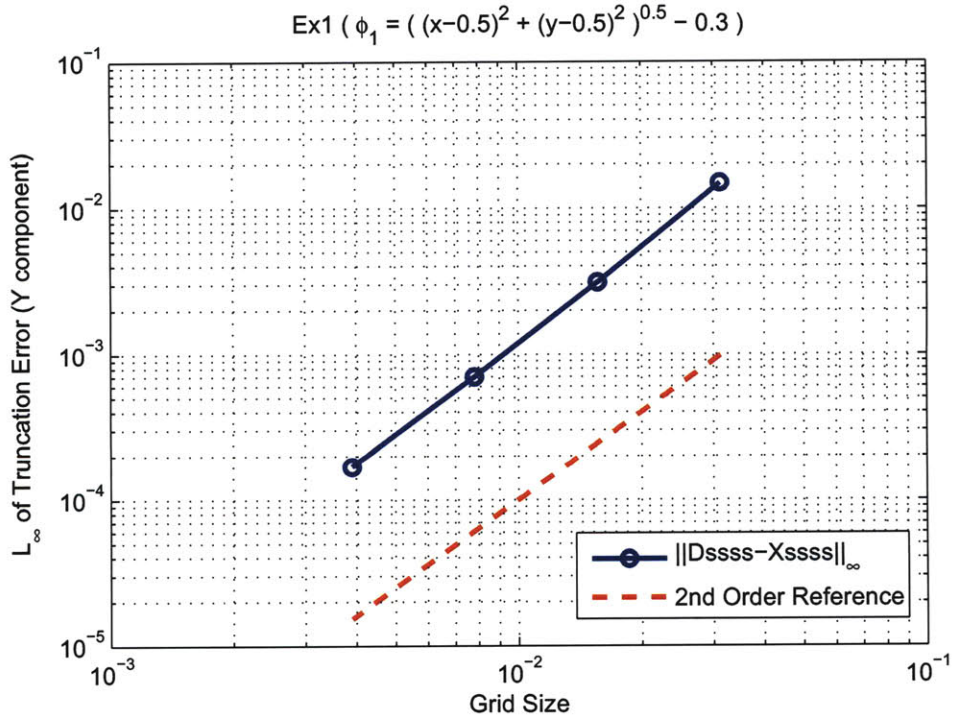


Figure 4-8: Convergence Study of L_∞-norm of Truncation Errors for $\phi_1(x,y) = \sqrt{(x-0.5)^2+(y-0.5)^2} - 0.3$

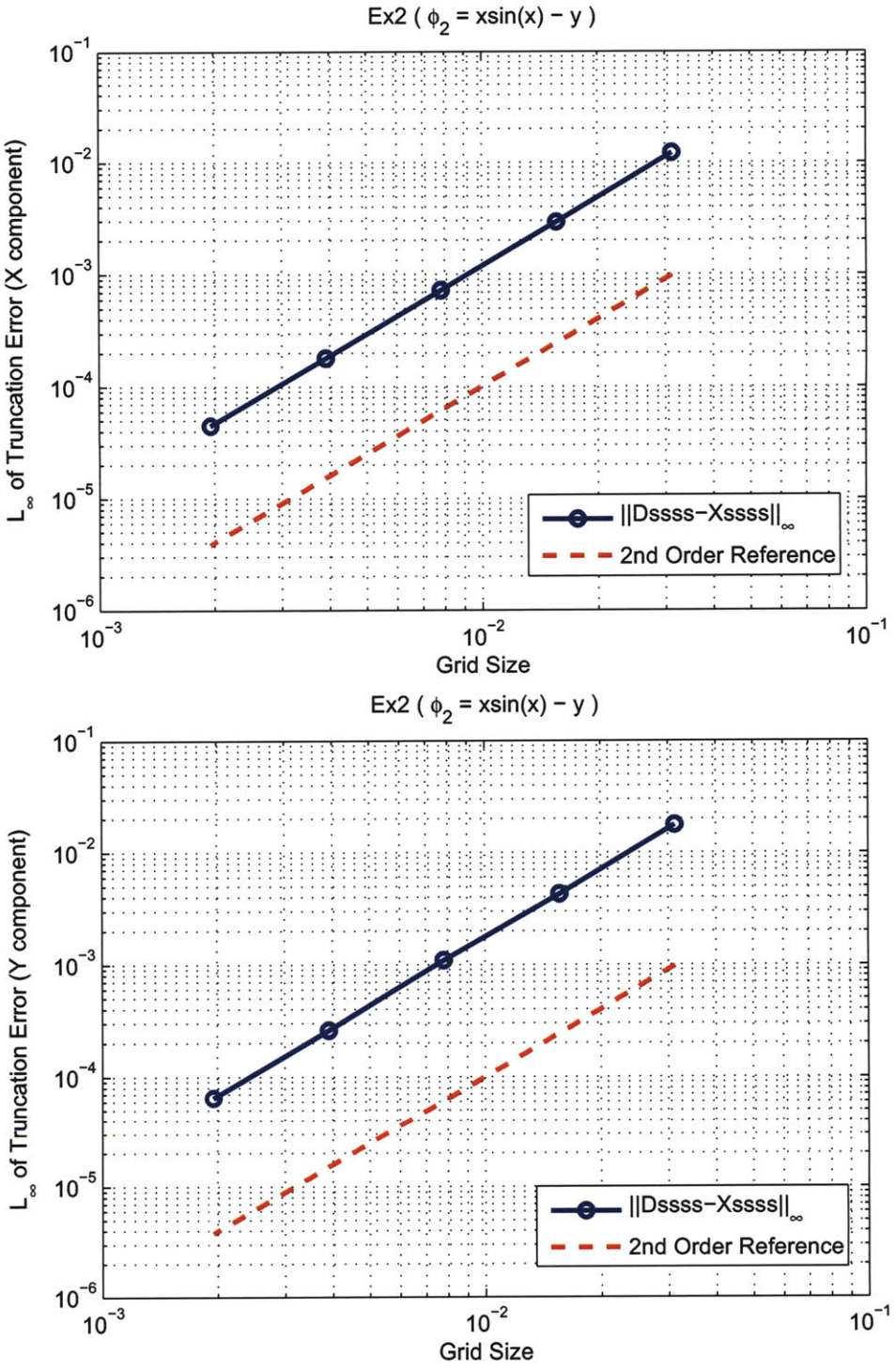


Figure 4-9: Convergence Study of L_∞-norm of Truncation Errors for $\phi_2(x, y) = xsin(x) - y$

4.5 Advection of Level Set Functions

We solve the advection equations of the level set functions to track the moving interface. Recall that the equation we should solve is

$$\phi_t + F|\nabla\phi| = 0 \tag{4.73}$$

The typical first and second order schemes for convex or non-convex speed functions are shown in [16]. However, they are not applicable in this work because we calculate the bending elasticity with ϕ which contains fourth derivatives. For instance, consider a second order fourth differencing of the level set ϕ at time step n.

$$(\phi_{xxxx})^n_{i,j} = \frac{\phi^n_{i-2,j} - 4\phi^n_{i-1,j} + 6\phi^n_{i,j} - 4\phi^n_{i+1,j} + \phi^n_{i+2,j}}{\Delta x^4} + O(\Delta x^2) \tag{4.74}$$

After one iteration, a second order upwind yields

$$\begin{aligned}(\phi_{xxxx})^{n+1}_{i,j} &= \frac{\phi^{n+1}_{i-2,j} - 4\phi^{n+1}_{i-1,j} + 6\phi^{n+1}_{i,j} - 4\phi^{n+1}_{i+1,j} + \phi^{n+1}_{i+2,j} + O(\Delta x^2)}{\Delta x^4} + O(\Delta x^2) \quad (4.75)\\ &= \frac{\phi^{n+1}_{i-2,j} - 4\phi^{n+1}_{i-1,j} + 6\phi^{n+1}_{i,j} - 4\phi^{n+1}_{i+1,j} + \phi^{n+1}_{i+2,j}}{\Delta x^4} + O\left(\frac{1}{\Delta x^2}\right) \quad (4.76)\end{aligned}$$

Therefore, we need at least fifth order accurate scheme to resolve accurate bending elasticity. Since Weighted Essentially Non-Oscillatory (WENO) scheme has one order of improvement in accuracy compared to ENO scheme ([22], [5], [23]) we use WENO scheme for spatial discretization of level set functions along with the TVD Runge-Kutta time discretization.

4.5.1 Weighted Essentially Non-Oscillatory (WENO) Scheme

In this subsection, we present fifth order WENO scheme from Liu et al.([22]) to discretize spatial derivative of level set function. WENO scheme can be obtained by taking interpolating polynomial which approximates the spatial differentials and building a convex combination of them. Here the fifth order WENO is represented

([12]).

Let us consider an one dimensional function ϕ whose differencing ϕ_x is what we want to approximate. Firstly, define a table of differences of $\{\phi_j\}$ such as

$$v_1 = (\phi_{i-2} - \phi_{i-3})/\Delta x \tag{4.77}$$
$$v_2 = (\phi_{i-1} - \phi_{i-2})/\Delta x \tag{4.78}$$
$$v_3 = (\phi_i - \phi_{i-1})/\Delta x \tag{4.79}$$
$$v_4 = (\phi_{i+1} - \phi_i)/\Delta x \tag{4.80}$$
$$v_5 = (\phi_{i+2} - \phi_{i+1})/\Delta x \tag{4.81}$$

Next, we define the indicator of the smoothness IS_j as

$$IS_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2 \tag{4.82}$$
$$IS_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \tag{4.83}$$
$$IS_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_1 - 4v_4 + v_5)^2 \tag{4.84}$$

And the weights w_j for the convex combination are determined as

$$w_j = \frac{\alpha_j}{\alpha_1 + \alpha_2 + \alpha_3} \qquad j = 1, 2, 3 \tag{4.85}$$

where the α_j are decided by the ENO property, i.e,

$$\alpha_1 = 0.1(IS_1 + \epsilon)^{-2} \tag{4.86}$$
$$\alpha_2 = 0.6(IS_2 + \epsilon)^{-2} \tag{4.87}$$
$$\alpha_3 = 0.3(IS_3 + \epsilon)^{-2} \tag{4.88}$$

where, by the convention, ϵ is added in the denominator to reduce the sensitivity

in case that the indicator of the smoothness is near zero. We use $\epsilon = 10^{-5}$ in this work.

Then we obtain the upwind scheme for $(\phi_x)_j$, i.e,

$$(\phi_x)_j = w_1\left(\frac{1}{3}v_1 - \frac{7}{6}v_2 + \frac{11}{6}v_3\right) + w_2\left(-\frac{1}{6}v_2 + \frac{5}{6}v_3 + \frac{1}{3}v_4\right) + w_3\left(\frac{1}{3}v_3 + \frac{5}{6}v_4 - \frac{1}{6}v_5\right) \tag{4.89}$$

For the downwind scheme, we can define the finite differences, v_k, in reverse way and apply the same idea. In addition to x direction, we can use an analogy for multi-dimension.

4.5.2 TVD Runge-Kutta Time Discretization

High order time discretizations are introduced by Shu and Osher in [23]. For the usage of the WENO scheme, here we use fourth order TVD Runge-Kutta time discretization. For the simplicity of the notation, we define the spatial discretization operator $L_j^n = L_j(\phi^n) \equiv F|\nabla\phi|$ and the time discretization becomes

$$\begin{aligned}
\phi_j^{(0)} &= \phi_j^n && (4.90)\\
\phi_j^{(1)} &= \phi_j^{(0)} + L_j^{(0)} && (4.91)\\
\phi_j^{(2)} &= \frac{1}{2}\phi_j^{(0)} + \frac{1}{2}\phi_j^{(1)} - \frac{1}{4}L_j^{(0)} + \frac{1}{2}L_j^{(1)} && (4.92)\\
\phi_j^{(3)} &= \frac{1}{9}\phi_j^{(0)} + \frac{2}{9}\phi_j^{(1)} + \frac{2}{3}\phi_j^{(2)} - \frac{1}{9}L_j^{(0)} - \frac{1}{3}L_j^{(1)} + L_j^{(2)} && (4.93)\\
\phi_j^{n+1} &= \frac{1}{3}\phi_j^{(1)} + \frac{1}{3}\phi_j^{(2)} + \frac{1}{3}\phi_j^{(3)} + \frac{1}{6}L_j^{(1)} + \frac{1}{6}L_j^{(3)} && (4.94)
\end{aligned}$$

4.5.3 Reinitialization

We shall make a use of WENO scheme for the re-initialization process to solve the re-initialization equation

$$\phi_t + sign(\phi_o)(|\nabla\phi| - 1) = 0 \tag{4.95}$$

where sign(ϕ_o) is the sign of initial ϕ, initial value in this equation.

Chapter 5

Simulation Results

5.1 2D Balloon Simulation

We investigate balloon simulations to check the capability of normal force jump. The force difference between inside and outside of a balloon holds the shape of it. The pressure difference depends on the curvature of the interface and the correlation to the pressure jump is given as

$$f_n = [p]_\Gamma = \sigma\kappa \tag{5.1}$$

where σ is the surface tension coefficient and κ is the signed curvature. The curvature is given as a function of level set ϕ [4]. The initial shape is given as an ellipse which has larger major axis in x direction. During the iterations, the geometry of a balloon reaches to an equilibrium state. The following figure shows the results of velocity vector, pressure and streamline field at different times. We set $\sigma = 0.01$ and $\mu = 0.002$ for this simulation. The interface oscillates to the equilibrium state as it converges. The figure XX shows the temporal evolution of major radii r_x and r_y with $\mu = 0.002$ and $T_o = 35$.

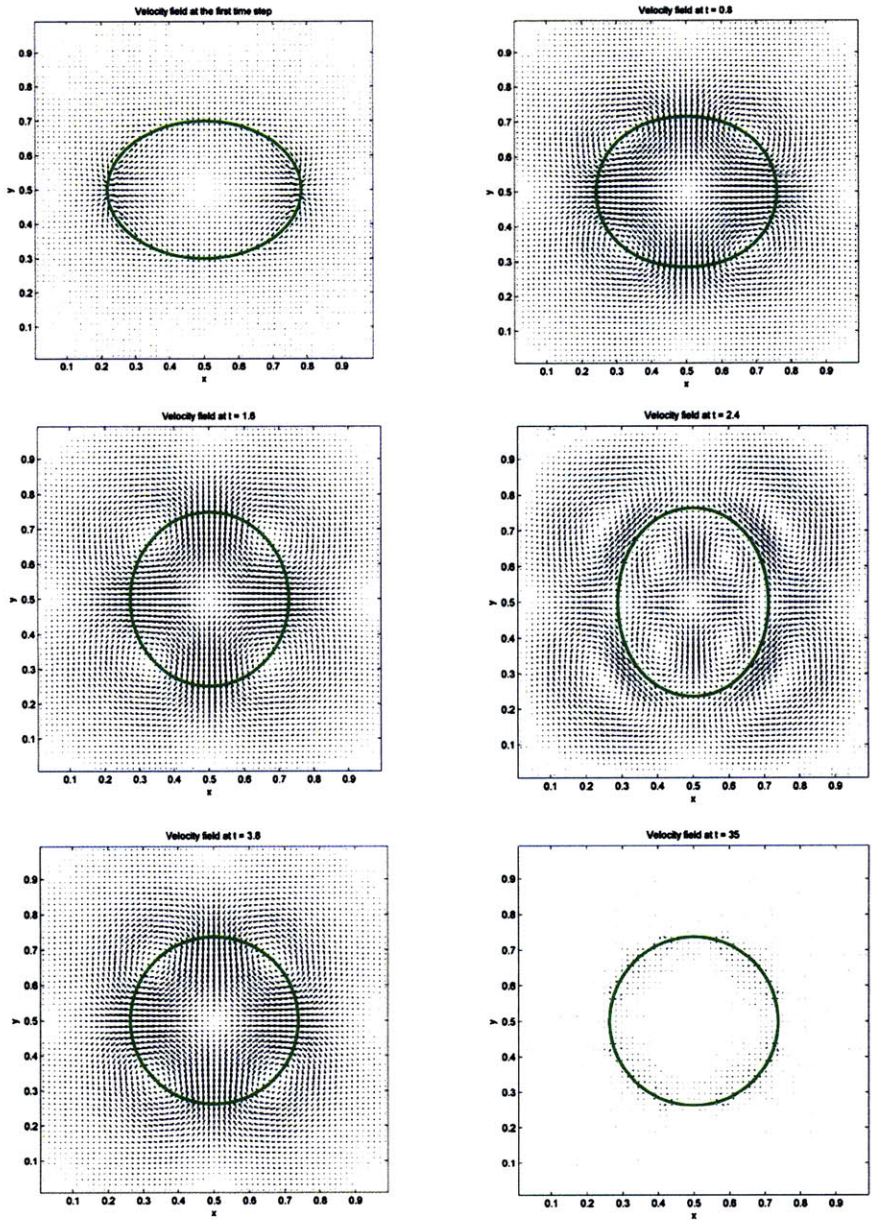


Figure 5-1: Vector field at different times

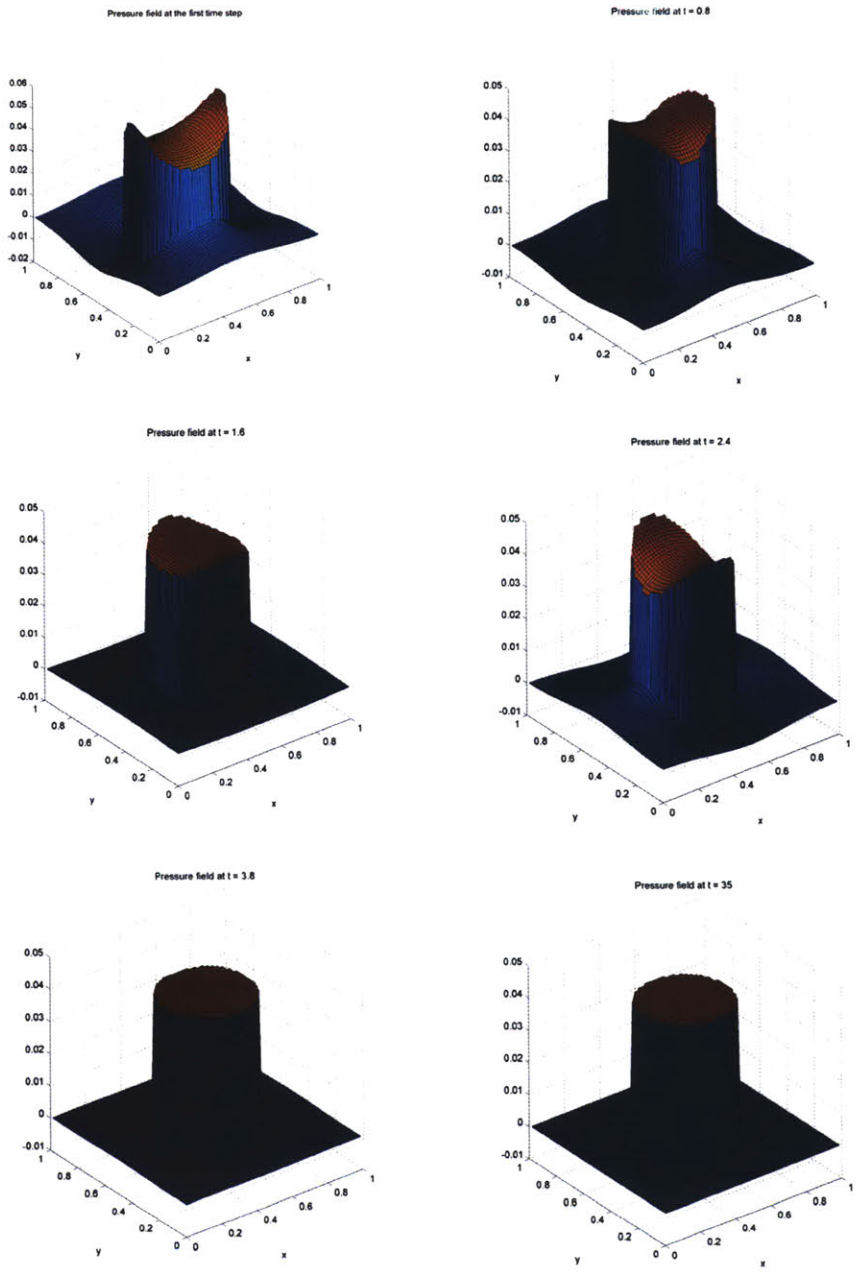


Figure 5-2: Pressure field at different times

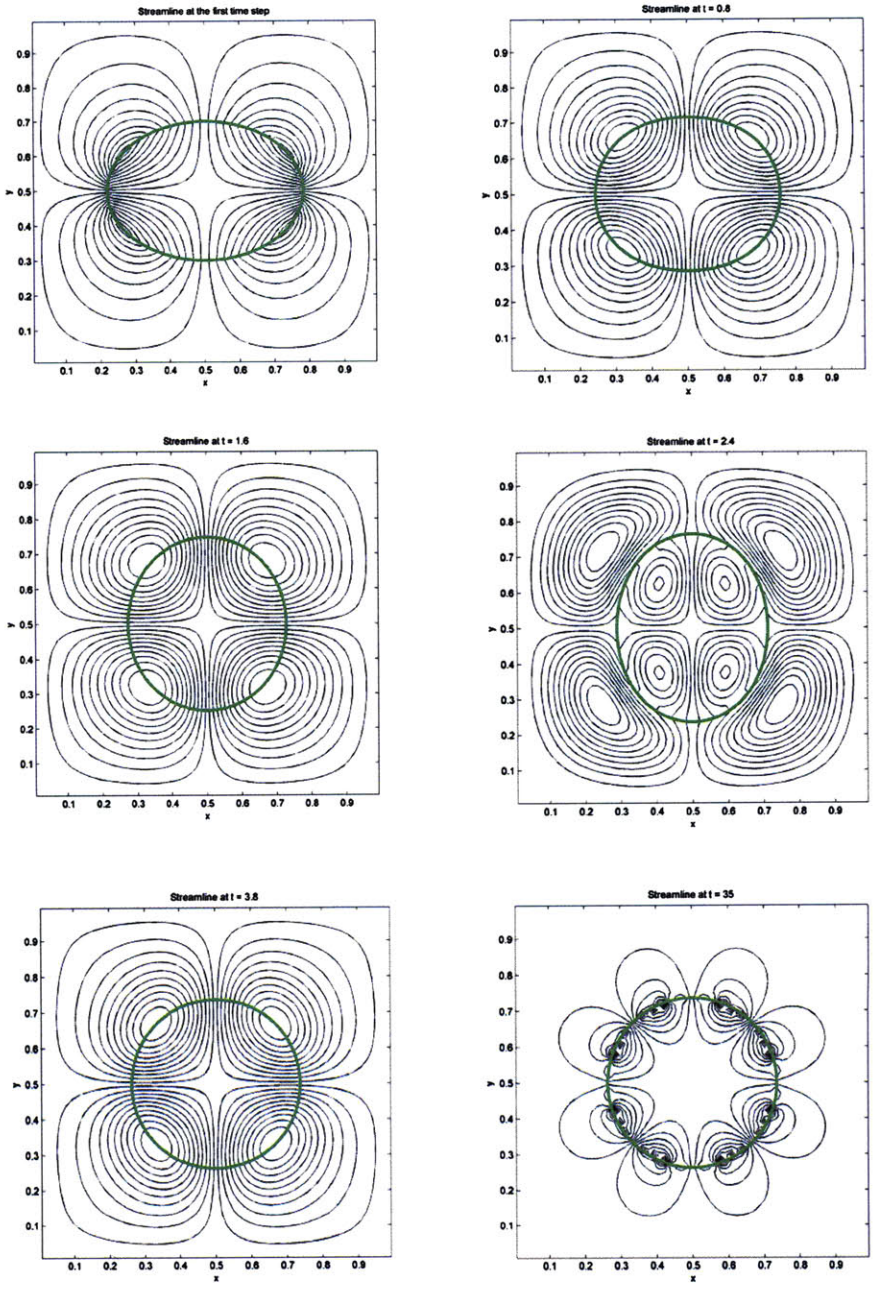


Figure 5-3: Streamlines at different times

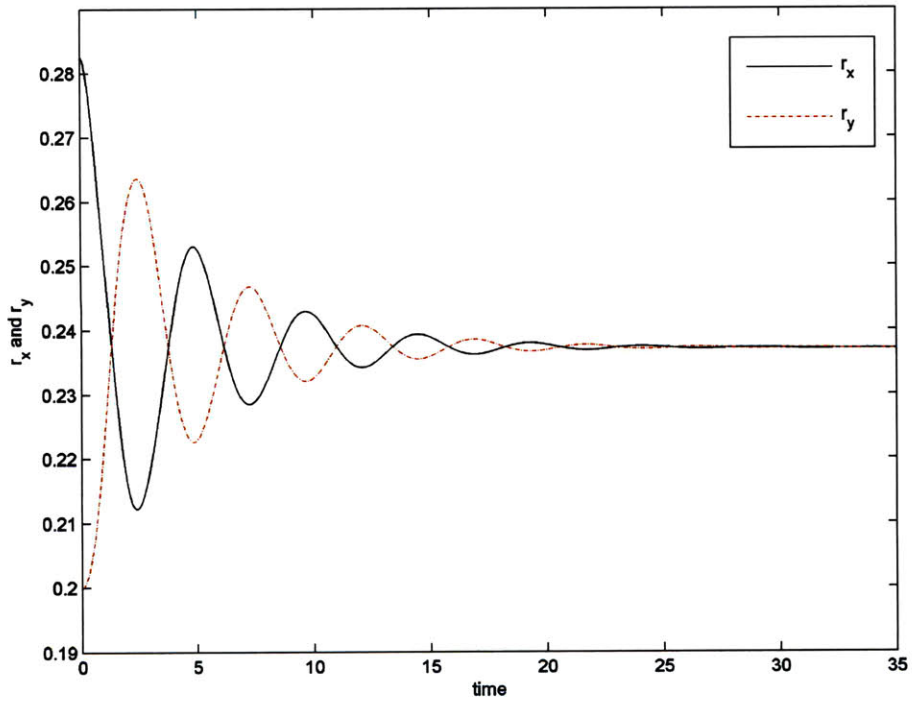


Figure 5-4: The evolution of r_x and r_y

5.2 Rotating Cylinder

In this section, we investigate the capacity of viscous stress jump through a simulation induced a rotating cylinder. A highly viscous fluid is used for the stability at initial stage. We apply a tangential density force $f_\tau = 0.2$ and viscosity $\mu = 0.02$ without advection of level set field. The following figures are result when the flow reached to a steady state. For this, $50x50$ grid points are used.

The convergence study can be obtained by taking an exact solution which is done in 256×256 grid.

Table 5.1: Convergence Study

N	$\lvert u_N - u_{256}\rvert_\infty$	$\lvert v_N - v_{256}\rvert_\infty$	$\lvert p_N - p_{256}\rvert_\infty$
32	0.0639	0.0639	0.1602
64	0.0250	0.0250	0.0864
128	0.0077	0.0077	0.0321

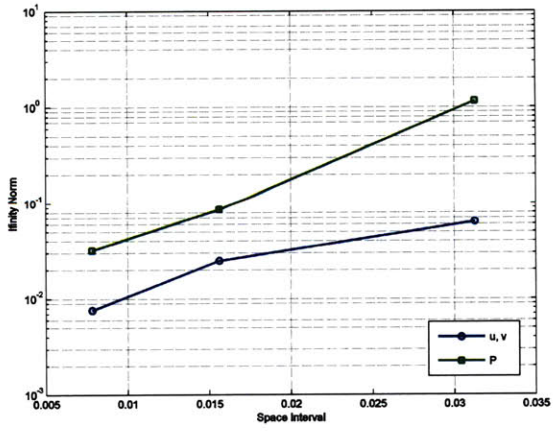
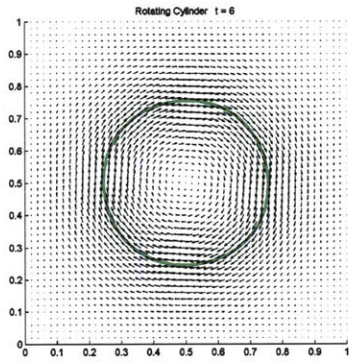
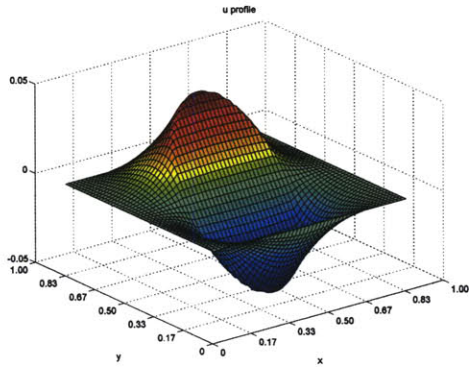


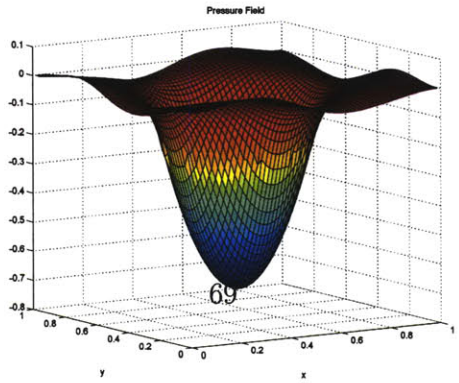
Figure 5-5: Convergence study of solutions



((a)) The vector plot of velocity field at time $t = 6$



((b)) The plot of the x component of the velocity field at time $t = 6$



((c)) The pressure field at time t=6

Figure 5-6: A steady state flow around a rotating cylinder

5.3 Simulation of Closed Curves with Bending Rigidity

In this section, we investigate the motions of interface which has bending stress. As in the 2D balloon cases, the initial shape is prescribed as an ellipse for which we can express by level set function, i.e.,

$$\phi(x,y) = \sqrt{(x-x_c)^2 + 0.5 \times (y-y_c)^2} - r_c \tag{5.2}$$

where x_c and y_c denote the center of principal Cartesian axes, and r_c is the radius of interface. We prescribe a circle and make it an ellipse by the above equation. As shown in following figure, the initially given interface vibrates and finally reaches to an equilibrium state. Firstly, we investigate the motions of a closed interface with different Young's modulus. In these simulations, we employ 64×64 grids in $\{x,y\} \in [0,1] \times [0,1]$ space, $(x_c, y_c) = (0.5, 0.5)$, $r_c = 0.25$, and we take three cases in which $E = 1e6$, $1e7$, and $1e8$ for the Young's modulus. The fluid density and viscosity are set as $\rho_f = 1.0$, the thickness of interface is $h = 1e-3$ and $\mu = 0.01$ which are consistent in all cases.

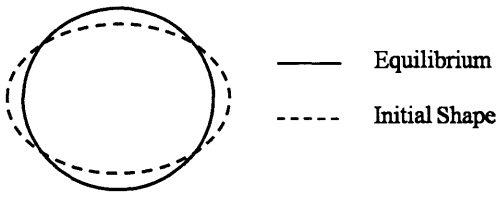
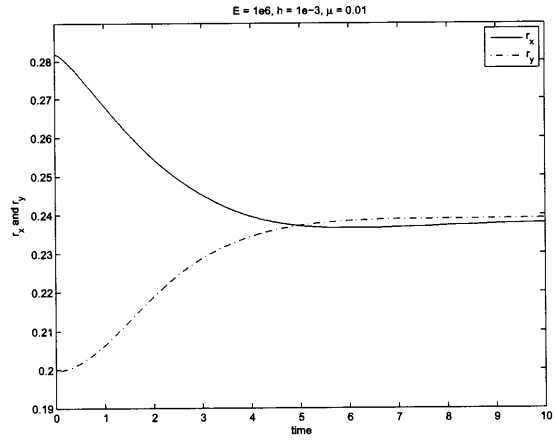
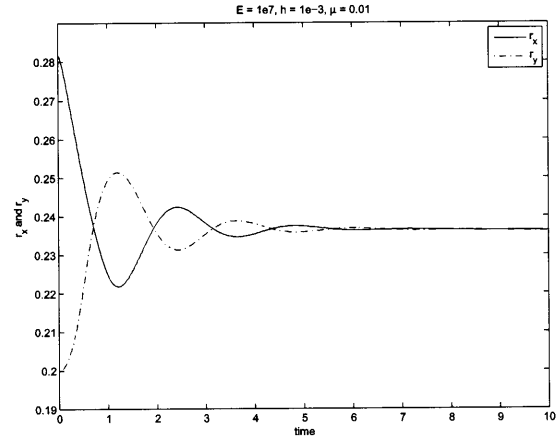


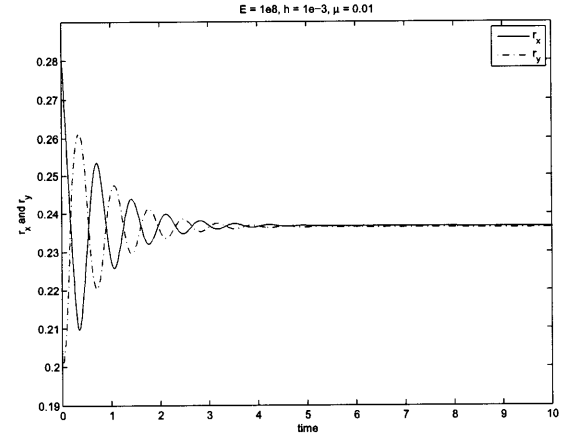
Figure 5-7: Problem statement of the simulation



((a)) $E = 10^6$



((b)) $E = 10^7$



((c)) $E = 10^8$

Figure 5-8: Radii evolution with time for different E

The Figure 5-8 shows the evolution of major radii at each choice of Young's modulus. As seen in the results, the motions of interface are very different in each case. For low E, the interface behaves slowly and the viscous damping is dominant in motion. However, the higher E is the faster motion can be seen, in which the motion of interface is oscillatory.

5.4 Comparison with a Lagrangian Method

In order to verify whether the numerical result is correct, this section provides a comparison to a Lagrangian method. As a choice of Lagrangian particle-tracking methods, we implement the Grid Based Particle Method (GBPM) which is one of the state-of-the-art Lagrangian formulations for evolving interface or surfaces [7] [1]. The GBPM evolves the interface by using the given velocity field on grid points and can store certain Lagrangian information of the interface at the control points, including normal, curvature and parametrization, which will be useful in various applications. The figure (5-9) shows the way of transferring information in the GBPM. The control points, which are called as foot-points, spread out the information onto the active grid points, and simultaneously the information (e.g. velocity) moves to the foot-points by sorts of interpolations.

Using the Lagrangian control points, we evaluate the bending rigidity and calculate singular forces acting on the fluid. For a comparison of the two methods, we can try the vibrating closed interface simulations again. Starting from an ellipse form, the interface reaches to an equilibrium state, which is a circle again. The following Figure (5-10) shows that the evolution of principal radii are excellently matching in two different methods. For this simulation, 64×64-grid are used and the following parameters were chosen, in which all parameters has SI Units.

$$\rho_f = 1.0,\ \mu = 0.01,\ \rho_s = 1000,\ h = 0.001,\ E = 1e7$$

In addition to the similar physical behaviors, there is a dramatic difference in computational cost. The following figure (5-11) shows the comparison of computational time of the two method. The time scale was measured per iteration and normalized by the smallest time that is observed in the Level Set approach.

The results shows that the computation costs more for the GBPM than the Level

[1] Dr. Shingyu Leung and the author had done a collaborative work on the coupling of fluid and GBPM numerical codes. Dr. S. Leung who is a PostDoc at University of California at Irvine visited to the Dept of Applied Math of MIT as a visiting researcher in Spring 2009. The results shown in this paper is based on the work with Dr. Shingyu Leung.

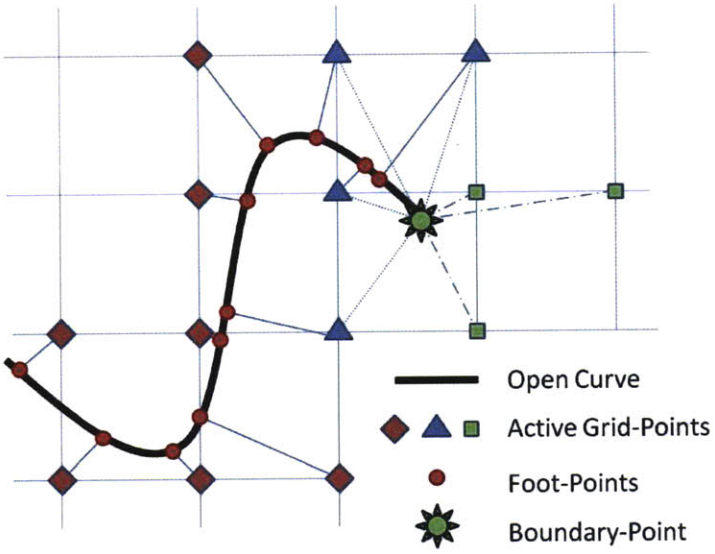


Figure 5-9: Definition of Active Grid Points and Foot-points and their Connections

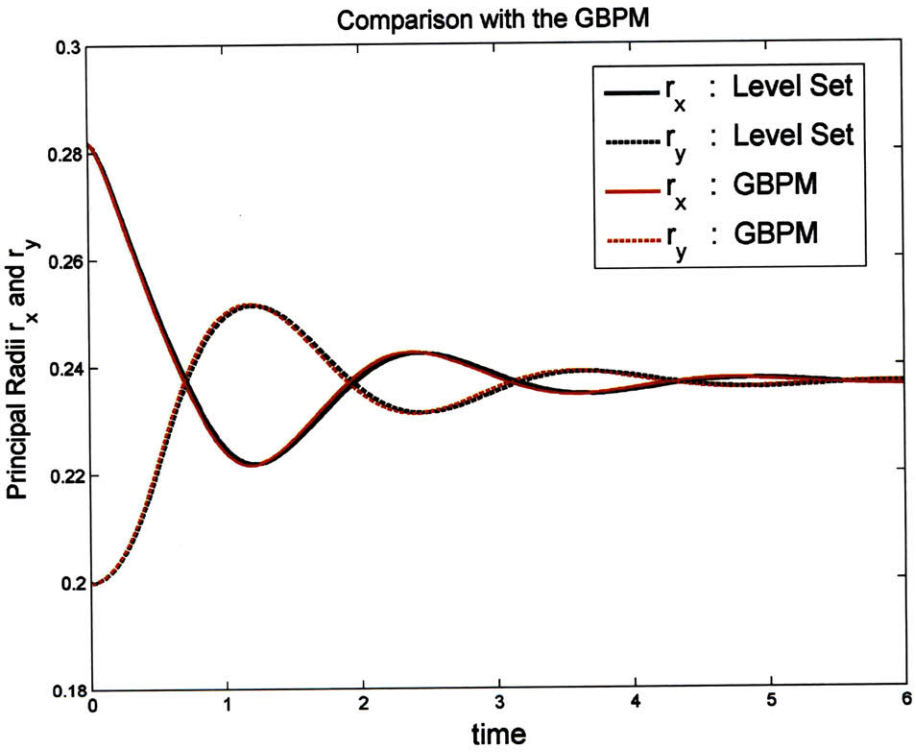


Figure 5-10: Comparison of Solutions between Level Set and GBPM

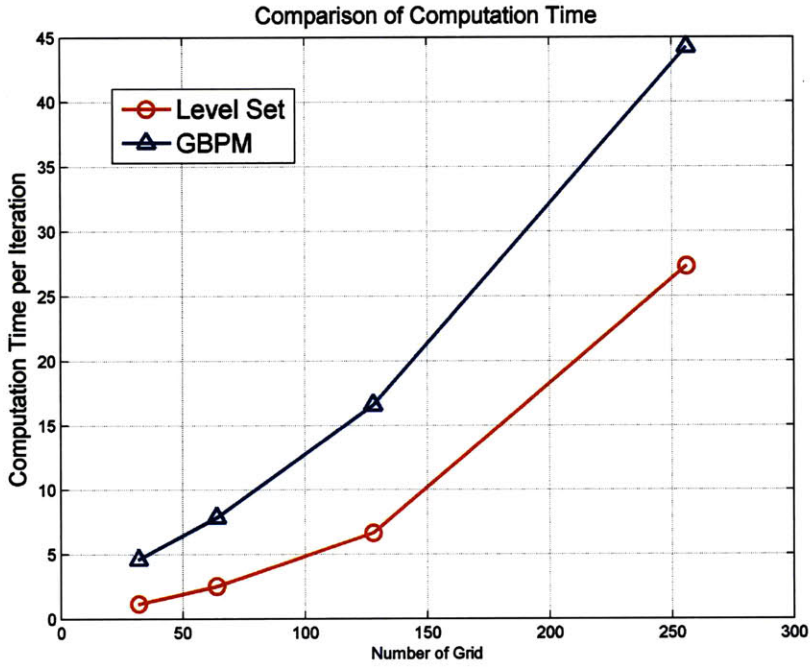


Figure 5-11: Comparison of Computation Time between Level Set and GBPM

Set formulation. Here, by estimating the portion of computation at each method, we can investigate what the most demanding part of the computation is. The following Figure (5-12) shows the averaged ratio of time-for-interpolation to the total computation time. Here the both time scale was normalized by the Level Set time scale so that the total computational cost has a value of 1 in the Level Set. As seen in the figure, the interpolation occupies most of portion in computation. The GBPM not only takes more computational cost, but the interpolation also has larger portion among the total computation. This is because that the interpolation requires a large amount of information, especially about the locations of control points. Also, the computation time for the interpolation depends on the order of accuracy. Both of the GBPM and the Level Set employ spiral spatial interpolation. If we reduce it to linear interpolation, the computation cost will be decreased, but we lose the accuracy. Therefore, reducing the number of interpolation is necessary to decrease the computational cost, for which the Level Set gives the minimum number of interpolation because most linear interpolations happen between two grid point except the spiral interpolation for the no-slip condition.

5.5 Simulation of Two Interfaces

Another important property of this algorithm is that we do not have any interface-penetration problem. Because this algorithm solves Navier-Stokes equations only and uses the fluid velocity for interface evolution, two or more number of interfaces do not penetrate through each other as long as the incompressibility condition of fluid is satisfied.

To verify the fact, we provide two-interface simulations. Firstly, in a boxed fluid domain, we apply constant wall velocities on the top and bottom walls. Here the wall speed is set as $U_{top} = -3\ [m/s]$ and $U_{bottom} = 3\ [m/s]$ in $\rho = 1.0\ [kg/m^3]$ and $\mu = 0.01\ [kg/m \cdot s]$. For the solid properties, we used $E = 10^8\ [Pa]$, $\rho_s = 1000\ [kg/m^3]$ and $h = 0.001[m]$. Because of the wall motion, the fluid rotates and the pressure in the middle region becomes lower, and therefore the interfaces would rotates without

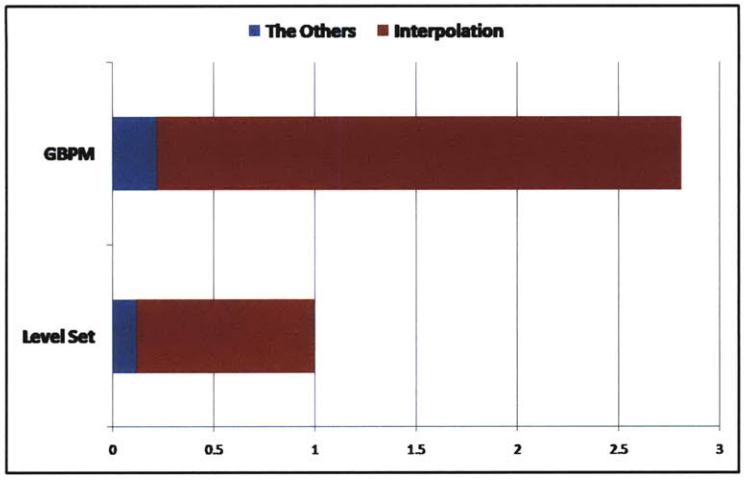
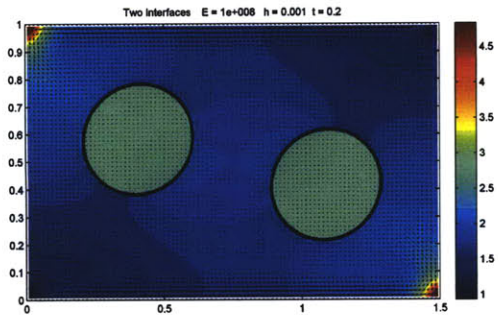


Figure 5-12: Comparison between Level Set and GBPM : normalized by the Level Set time scale

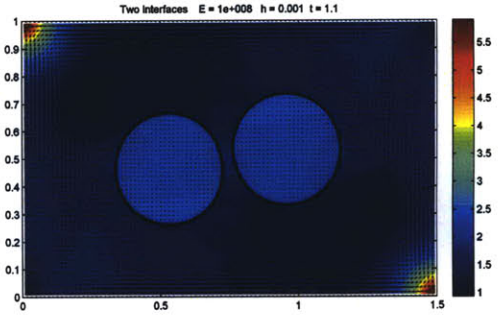
contacting to the walls. The Figure (5-13) and (5-14) show the time evolution of solid in the viscous rotation flow. The arrow means the velocity vectors, color denotes pressure and solid lines are interfaces. As you can see in the figures, the interfaces contact once, rotates a little keeping the contact and finally are separated by the flow field.

And the Figure (5-15)-(5-16) show the same simulation of a faster moving wall velocity, $U_{speed} = 5\ [m/s]$, with same parameters as we did in the above example. As you can see, since the wall velocities are faster than previous case, the motion of interfaces becomes more flexible. You can see that the interfaces do not penetrate through each other even in this case.

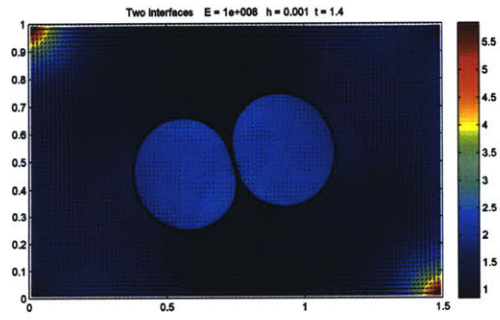
The penetration problem is an important topic in computer graphics. In animation field, people do hair-blowing simulations by solving independent hair-thread dynamics systems. In this simulation, the numerical code should detect if some hair threads are not crossing though each other, and whenever we find it we have to impose a special numerical technique to make them separate. However, by introducing an idea of using incompressibility of fluid into the motion of hair particles, we can automatically prevent the penetration between interfaces.



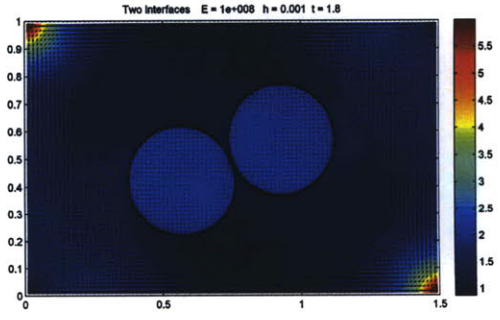
((a)) t = 0.1 sec



((b)) t = 1.1 sec

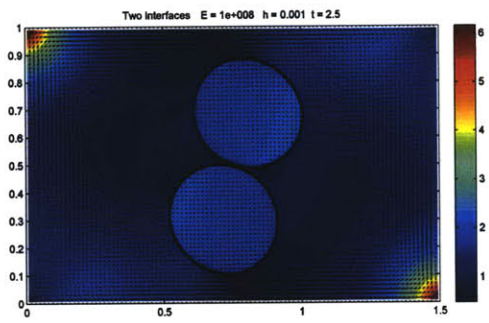


((c)) t = 1.4 sec

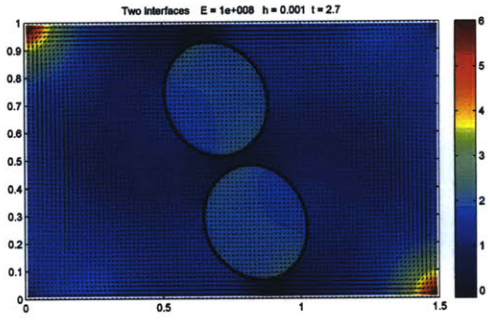


((d)) t = 1.8 sec

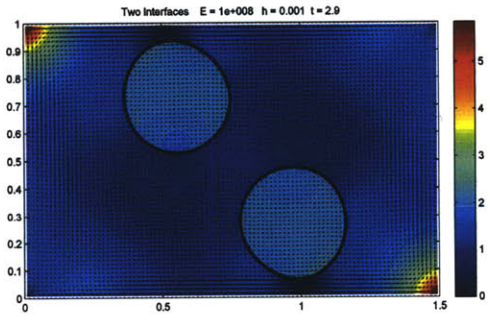
Figure 5-13: The evolution of two Interfaces with a slow top and bottom wall speed



((a)) t = 2.5 sec

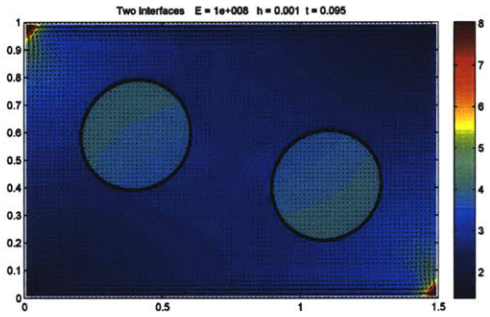


((b)) t = 2.7 sec

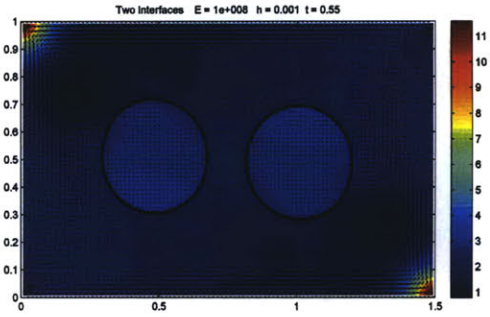


((c)) t = 2.9 sec

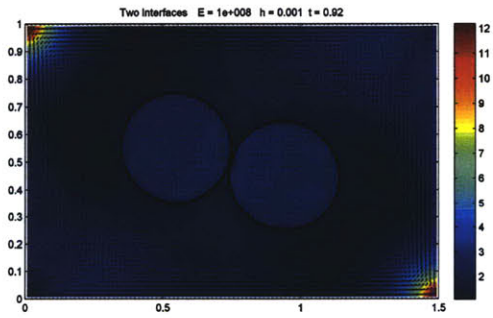
Figure 5-14: The evolution of two Interfaces with a slow top and bottom wall speed (Continued)



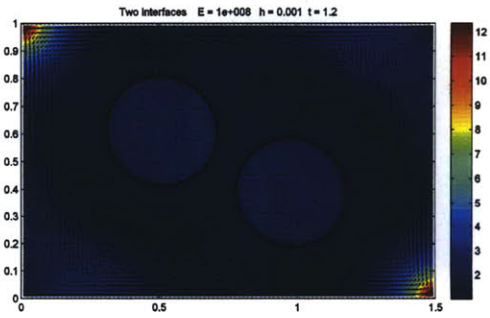
((a)) t = 0.1 sec



((b)) t = 0.5 sec

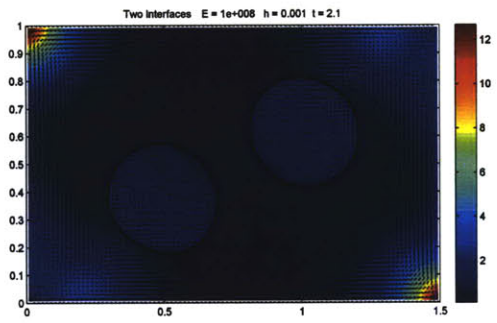


((c)) t = 0.9 sec

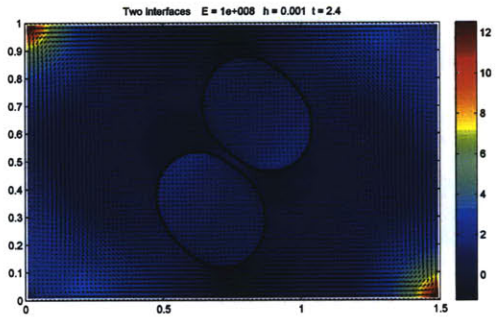


((d)) t = 1.2 sec

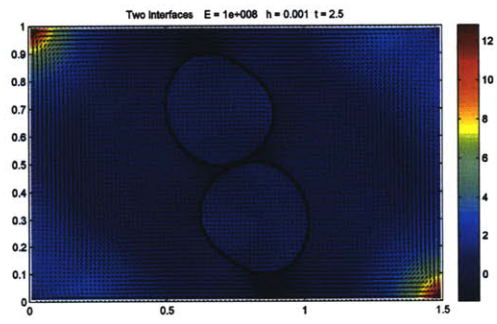
Figure 5-15: The evolution of two Interfaces with a fast top and bottom wall speed



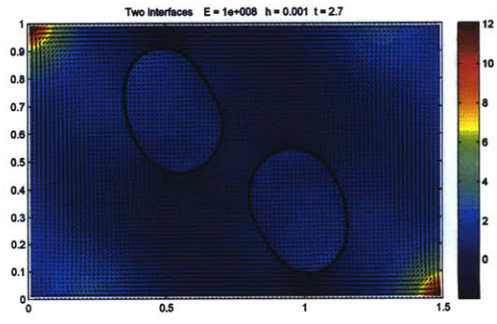
((a)) t = 2.1 sec



((b)) t = 2.4 sec



((c)) t = 2.5 sec



((d)) t = 2.7 sec

Figure 5-16: The evolution of two Interfaces with a fast top and bottom wall speed (Continued)

Chapter 6

Summary and Future Works

6.1 Summary

In Chapter 5, we surveyed the accuracy of the implicit algorithm and numerical efficiency in terms of computational cost. The following facts can be concluded by this work.

- The Level Set function can represents a real thin solid interface for fluid-structure interaction. In Chapter 3, the linear bending rigidity has been formulated as a second-order in terms of space interval.

- The implicit representation of the interface can reduce complexity of numerical code and computational cost. In Chapter 5, we realized that the interpolation requires demanding computational cost. By reducing the number of interpolation and degree of freedom of the data structure, the computation time for fluid-structure interaction was decreased in the case of vibrating closed-interface. We did not have theoretical method to evaluate the quantity of reduction. But the results implies that the computation time will be highly reduced when the length of interface is small compared to the outer fluid region. Even if the interface occupies a small portion of fluid domain, the computation time will be much smaller than the case that we use moving body-fitted grid.

- By observing the comparison of computational cost between the Level Set approach and the Lagrangian approach, we realize that the efficiency of numerical code will increase further in three-dimensional simulations. In three-dimensional simulations, the Lagrangian control-points will include more complicated data information. Since the Level Set is a scalar variable, the Level Set representation of interface would have smaller degree of freedom to represent interfaces compared to the IIM.

- By solving the fluid equation only and introducing the incompressibility of fluid, we can prevent the penetration problem between the interfaces. To use local fluid velocity for the evolution of interfaces guarantees that the penetration does not happen during the simulation. This implies that we do not any special contact algorithm for structure. Since this works for two interfaces, the method would work for more than two interfaces.

6.2 Future Works

This section provides the main required future works for improvements of the FSI algorithm.

- Firstly, we need to update the reinitialization of the level set because reinitialization process takes the loss of volume. However, [1] provides a level set evolution technique without reinitialization procedure by introducing overall energy function. This work is beyond the reintialization process, but by introducing the non-reinitialization process we can improve the loss of volume problem.

- The second required improvement is that we have to track the tangential motion of solid exactly. Here by using the level set function ϕ which evolves in normal direction we introduced the FSI algorithm. However, this is not exact actually. One possible idea is to use two level sets: one tracks normal motion and the other tracks the tangential motion of solid. For the flapping motion to be happened in open curve simulation, we need the stretching term which was neglected in

Section 3.2.2. The stretching term can be expressed by the perpendicular level set that is introduced in the Equation (3.1). By adding the stretching motion, the fluid-structure interaction problem with the level set expression becomes complete.

Appendix A

Level Set Expression of Bending Rigidity

In this part, the exact forms of PDE which is identical to the bending elastic stress of the interface in two dimension. Note that $\phi_i \equiv \partial\phi/\partial\, i = \phi_i(\mathbf{X},\ t),\ \ i = x,\ y$ for simplicity. By the chain rule, the first component of $\mathbf{D}_{ssss}$ is

$$\frac{9(c_{11}\phi_{xy}+c_{12}\phi_{yy})(q_1)^2}{4(\phi_x^2+\phi_y^2)^{5/2}}-\frac{15\phi_y(q_1)^3}{8(\phi_x^2+\phi_y^2)^{5/2}}-\frac{3(q_1)(q_2)}{2(\phi_x^2+\phi_y^2)^{3/2}}-\frac{3(c_{11}\phi_{xy}+c_{12}\phi_{yy})(q_4)}{2(\phi_x^2+\phi_y^2)^{3/2}}+\frac{9\phi_y(q_1)(q_4)}{4(\phi_x^2+\phi_y^2)^{5/2}}$$

$$+\frac{q_5}{(\phi_x^2+\phi_y^2)^{1/2}}-\frac{\phi_y(6(c_{11}\phi_{xx}+c_{12}\phi_{xy})(q_3)+6(c_{11}\phi_{xy}+c_{12}\phi_{yy})(q_2)+2\phi_x(q_6)+2\phi_y(q_5))}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.1}$$

and the second component is

$$-\frac{9(c_{11}\phi_{xx}+c_{12}\phi_{xy})(q_1)^2}{4(\phi_x^2+\phi_y^2)^{5/2}}+\frac{15\phi_x(q_1)^3}{8(\phi_x^2+\phi_y^2)^{5/2}}+\frac{3(q_1)(q_3)}{2(\phi_x^2+\phi_y^2)^{3/2}}+\frac{3(c_{11}\phi_{xx}+c_{12}\phi_{xy})(q_4)}{2(\phi_x^2+\phi_y^2)^{3/2}}-\frac{9\phi_x(q_1)(q_4)}{4(\phi_x^2+\phi_y^2)^{5/2}}$$

$$-\frac{q_6}{(\phi_x^2+\phi_y^2)^{1/2}}+\frac{\phi_x(6(c_{11}\phi_{xx}+c_{12}\phi_{xy})(q_3)+6(c_{11}\phi_{xy}+c_{12}\phi_{yy})(q_2)+2\phi_x(q_6)+2\phi_y(q_5))}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.2}$$

, where the coefficients c_{11} through c_{32} and q_1 through q_6 are values defined as followings.

$$c_{11} = \frac{\phi_y}{(\phi_x^2+\phi_y^2)^{1/2}} \tag{A.3}$$

$$c_{12} = \frac{-\phi_x}{(\phi_x^2+\phi_y^2)^{1/2}} \tag{A.4}$$

$$q_1 = 2\phi_x(c_{11}\phi_{xx}+c_{12}\phi_{xy})+2\phi_y(c_{11}\phi_{xy}+c_{12}\phi_{yy}) \tag{A.5}$$

$$c_{21} = \frac{c_{11}\phi_{xy}+c_{12}\phi_{yy}}{(\phi_x^2+\phi_y^2)^{1/2}} - \frac{\phi_y(q_1)}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.6}$$

$$c_{22} = -\frac{c_{11}\phi_{xx}+c_{12}\phi_{xy}}{(\phi_x^2+\phi_y^2)^{1/2}} + \frac{\phi_x(q_1)}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.7}$$

$$q_2 = c_{21}\phi_{xy}+c_{22}\phi_{yy}+c_{11}(c_{11}\phi_{xxy}+c_{12}\phi_{xyy})+c_{12}(c_{11}\phi_{xyy}+c_{12}\phi_{yyy}) \tag{A.8}$$

$$q_3 = c_{21}\phi_{xx}+c_{22}\phi_{xy}+c_{11}(c_{11}\phi_{xxx}+c_{12}\phi_{xxy})+c_{12}(c_{11}\phi_{xxy}+c_{12}\phi_{xyy}) \tag{A.9}$$

$$q_4 = 2(c_{11}\phi_{xx}+c_{12}\phi_{xy})^2+2(c_{11}\phi_{xy}+c_{12}\phi_{yy})^2+2\phi_x(q_3)+2\phi_y(q_2) \tag{A.10}$$

$$c_{31} = -\frac{(c_{11}\phi_{xy}+c_{12}\phi_{yy})(q_1)}{(\phi_x^2+\phi_y^2)^{3/2}} + \frac{3\phi_y(q_1)^2}{4(\phi_x^2+\phi_y^2)^{5/2}} + \frac{q_2}{(\phi_x^2+\phi_y^2)^{1/2}} - \frac{\phi_y(q_4)}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.11}$$

$$c_{32} = \frac{(c_{11}\phi_{xx}+c_{12}\phi_{xy})(q_1)}{(\phi_x^2+\phi_y^2)^{3/2}} - \frac{3\phi_x(q_1)^2}{4(\phi_x^2+\phi_y^2)^{5/2}} - \frac{q_3}{(\phi_x^2+\phi_y^2)^{1/2}} + \frac{\phi_x(q_4)}{2(\phi_x^2+\phi_y^2)^{3/2}} \tag{A.12}$$

(A.13)

and

$$q_5 = c_{31}\phi_{xy}+c_{32}\phi_{yy}+2c_{21}(c_{11}\phi_{xxy}+c_{12}\phi_{xyy})+2c_{22}(c_{11}\phi_{xyy}+c_{12}\phi_{yyy})$$

$$+c_{11}(c_{21}\phi_{xxy}+c_{22}\phi_{xyy}+c_{11}(c_{11}\phi_{xxxy}+c_{12}\phi_{xxyy})+c_{12}(c_{11}\phi_{xxyy}+c_{12}\phi_{xyyy}))$$

$$+c_{12}(c_{21}\phi_{xyy}+c_{22}\phi_{yyy}+c_{11}(c_{11}\phi_{xxyy}+c_{12}\phi_{xyyy})+c_{12}(c_{11}\phi_{xyyy}+c_{12}\phi_{yyyy}))) \tag{A.14}$$

$$q_6 = c_{31}\phi_{xx} + c_{32}\phi_{xy} + 2c_{21}(c_{11}\phi_{xxx} + c_{12}\phi_{xxy}) + 2c_{22}(c_{11}\phi_{xxy} + c_{12}\phi_{xyy})$$

$$+c_{11}(c_{21}\phi_{xxx} + c_{22}\phi_{xxy} + c_{11}(c_{11}\phi_{xxxx} + c_{12}\phi_{xxxy}) + c_{12}(c_{11}\phi_{xxxy} + c_{12}\phi_{xxyy}))$$

$$+c_{12}(c_{21}\phi_{xxy} + c_{22}\phi_{xyy} + c_{11}(c_{11}\phi_{xxxy} + c_{12}\phi_{xxyy}) + c_{12}(c_{11}\phi_{xxyy} + c_{12}\phi_{xyyy}))) \quad \text{(A.15)}$$

Bibliography

[1] Changfeng Gui Chunming Li, Chenyang Xu and Martin D. Fox. Level set evolution without re-initialization: A new variational formulation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

[2] Benjamin S. H. Connell and Dick K. P. Yue. Flapping dynamics of a flag in a uniform stream. *J. Fluid Mech*, 581:33–67, 2007.

[3] Benjamin S.H. Connell. *Numerical investigation of the flow-body interaction of thin flexible foils and ambient flow.* PhD thesis, Massachusetts Institute of Technology, 2006.

[4] Eduard Harabetian and Stanley Osher. Regularization of ill-posed problems via the level set approach. *SIAM J. Appl. Math*, 58(6):1689–1706, 1998.

[5] Guang-Shan Jiang and Danping Peng. Weighted eno schemes for hamilton-jacobi equations. *SIAM J. Sci. Comput*, 21(6):2126–2143, 2000.

[6] Duc Vinh Le. *An immersed interface method for solving viscous incompressible flows involving rigid and flexible boundaries.* PhD thesis, HCMC University of Technology, 2001.

[7] Shingyu Leung and Hongkai Zhao. A grid based particle method for evolution of open curves and surfaces. *J. Comput. Phys.*, 228(20):7706–7728, 2009.

[8] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sorces. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.

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

[10] Zhilin Li and Ming-Chih Lai. The immersed interface method for the navier-stokes equations with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.

[11] T. Dornseifer M.Griebel and T. Neunhoeffer. Numerical simulation in fluid dynamics: A practical introduction. *Society for Industrial and Applied Mathematics*, 1998.

[12] J.C. Nave. *Direct Numerical Simulation of Liquid Films.* PhD thesis, UCSB, 2004.

[13] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1997.

[14] Carolyn L. Phillips. The level-set method. *MIT Undergraduate Journal of Mathematics*, 2001.

[15] Benjamin Seibold. A compact and fast matlab code solving the incompressible navier-stokes equations on rectangular domains. *Massachusetts Institute of Technology*, 2008.

[16] J.A. Sethian. *Level Set Methods.* Cambridge University Press, 1996.

[17] J.A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 1999.

[18] Sheng Xu. The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flow. *J. Comput. Phys.*, 227:5045–5071, 2008.

[19] Sheng Xu. Singular forces in the immersed interface method for rigid objects in 3d. *Applied Mathematics Letters*, 22:827–833, 2009.

[20] Sheng Xu and Z. Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *J. Comput. Phys.*, 216:454–493, 2006.

[21] Ronald P. Fedkiw Xu-Dong Liu and Myungjoo Kang. A boundary condition capturing method for poisson's equation on irregular domains. *J. Comput. Phys.*, 160:151–178, 2000.

[22] Stanley Osher Xu-Dong Liu and Tony Chan. Weighted essentially non-oscillatory schemes. *J. of Comput. Phys.*, 115:200–212, 1994.

[23] Chi-Wang Shu Yong-Tao Zhang, Jing Shi and Ye Zhou. Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high reynolds numbers. *PHYSICAL REVIEW*, 68, 2003.