2006

An Adaptively refined Cartesian grid method for moving boundary problems applied to biomedical systems

Sreedevi Krishnan
University of Iowa

AN ADAPTIVELY REFINED CARTESIAN GRID METHOD FOR MOVING BOUNDARY PROBLEMS APPLIED TO BIOMEDICAL SYSTEMS

by

Sreedevi Krishnan

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

July 2006

Thesis Supervisor: Associate Professor H. S. Udaykumar

ABSTRACT

A major drawback in the operation of mechanical heart valve prostheses is thrombus formation in the near valve region potentially due to the high shear stresses present in the leakage jet flows through small gaps between leaflets and the valve housing. Detailed flow analysis in this region during the valve closure phase is of interest in understanding the relationship between shear stress and platelet activation.

An efficient Cartesian grid method is developed for the simulation of incompressible flows around stationary and moving three-dimensional immersed solid bodies as well as fluid-fluid interfaces. The embedded boundaries are represented using Levelsets and treated in a sharp manner without the use of source terms to represent boundary effects. The resulting algorithm is implemented in a straightforward manner in three dimensions and retains global second-order accuracy. When dealing with problems of disparate length scales encountered in many applications, it is necessary to resolve the physically important length scales adequately to ensure accuracy of the solution. Fixed grid methods often have the disadvantage of heavy mesh requirement for well resolved calculations. A quadtree based adaptive local mesh refinement scheme is developed to complement the sharp interface Cartesian grid method scheme for efficient and optimized calculations. Detailed timing and accuracy data is presented for a variety of benchmark problems involving moving boundaries.

The above method is then applied to modeling heart valve closure and predicting thrombus formation. Leaflet motion is calculated dynamically based on the fluid forces acting on it employing a fluid-structure interaction algorithm. Platelets are modeled and tracked as point particles by a Lagrangian particle tracking method which incorporates

the hemodynamic forces on the particles. Leaflet closure dynamics including rebound is analyzed and validated against previous studies. Vortex shedding and formation of recirculation regions are observed downstream of the valve, particularly in the gap between the valve and the housing. Particle exposure to high shear and entrapment in recirculation regions with high residence time in the vicinity of the valve are observed corresponding to regions prone to thrombus formation.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

AN ADAPTIVELY REFINED CARTESIAN GRID METHOD FOR MOVING BOUNDARY PROBLEMS APPLIED TO BIOMEDICAL SYSTEMS

by

Sreedevi Krishnan

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

July 2006

Thesis Supervisor: Associate Professor H. S. Udaykumar

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Sreedevi Krishnan

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Mechanical Engineering at the July 2006 graduation.

Thesis Committee: _____
H. S. Udaykumar, Thesis Supervisor

_____
V. C. Patel

_____
K. B. Chandran

_____
K. S. Schulze

_____
J. S. Marshall

To my parents, my sister and my husband for all their love and support.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Prof. H.S. Udaykumar for his constant guidance and encouragement in all aspects of my professional life. He has been an outstanding mentor and role model throughout the period of my doctoral study. The past five years spent working with him have been an invaluable learning experience, the lessons of which I will carry with me for the rest of my career. I am also grateful to him and to the Department of Mechanical and Industrial Engineering for the uninterrupted financial support during the entire period of my graduate study.

I have personally interacted and worked with all the other members of my committee, Prof. V. C. Patel, Prof. K. B. Chandran, Dr. K. S. Schulze and Prof. J. S. Marshall on many occasions during the course of the past few years. I am grateful to them for taking time out of their busy schedules for our meetings and consultations and for their valuable insights and suggestions in giving direction and focus to my doctoral research work.

My list will not be complete till I acknowledge the past and present students of our research group who have helped make my long stay in the lab so happy and enjoyable. I consider myself fortunate to have been able to work with such a group of intelligent, motivated and friendly people.

Finally, I would like to thank my family, especially my parents Dr. K. N. Krishnan and Mrs. Savithri Krishnan and my sister, Shalini for their unconditional love and encouragement at all times. Ever since we first met, my husband Raj Venuturumilli has been involved in every facet of my personal and professional life including attending classes, doing homework, discussing research and so on. I would like to thank him for

being there for me as a constant source of love and strength through all the ups and downs of life in the past six or seven years.

ABSTRACT

A major drawback in the operation of mechanical heart valve prostheses is thrombus formation in the near valve region potentially due to the high shear stresses present in the leakage jet flows through small gaps between leaflets and the valve housing. Detailed flow analysis in this region during the valve closure phase is of interest in understanding the relationship between shear stress and platelet activation.

An efficient Cartesian grid method is developed for the simulation of incompressible flows around stationary and moving three-dimensional immersed solid bodies as well as fluid-fluid interfaces. The embedded boundaries are represented using Levelsets and treated in a sharp manner without the use of source terms to represent boundary effects. The resulting algorithm is implemented in a straightforward manner in three dimensions and retains global second-order accuracy. When dealing with problems of disparate length scales encountered in many applications, it is necessary to resolve the physically important length scales adequately to ensure accuracy of the solution. Fixed grid methods often have the disadvantage of heavy mesh requirement for well resolved calculations. A quadtree based adaptive local mesh refinement scheme is developed to complement the sharp interface Cartesian grid method scheme for efficient and optimized calculations. Detailed timing and accuracy data is presented for a variety of benchmark problems involving moving boundaries.

The above method is then applied to modeling heart valve closure and predicting thrombus formation. Leaflet motion is calculated dynamically based on the fluid forces acting on it employing a fluid-structure interaction algorithm. Platelets are modeled and tracked as point particles by a Lagrangian particle tracking method which incorporates

the hemodynamic forces on the particles. Leaflet closure dynamics including rebound is analyzed and validated against previous studies. Vortex shedding and formation of recirculation regions are observed downstream of the valve, particularly in the gap between the valve and the housing. Particle exposure to high shear and entrapment in recirculation regions with high residence time in the vicinity of the valve are observed corresponding to regions prone to thrombus formation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

FLUID MECHANICS OF MECHANICAL HEART VALVES

1.1 Introduction

This work is directed towards developing efficient numerical techniques to analyze the fluid mechanics of moving boundary problems. The specific characteristic targeted for study is the effect of moving boundaries on transport of molecular and particulate matter in biological systems. In as much as that motion is intrinsic to life, every living organism, from single-celled organisms to man, employs the action of moving boundaries to conduct basic life processes. In the human system moving boundaries abound, such as the motion of the heart, operation of heart valves, deformation of red blood cells as they squeeze through narrow capillaries, the breathing motion of lungs, contractions of the stomach, ureter, uterus, intestines and other organs to transport material from gases to solids, motion of the tongue and the production of sound, the motion of eyelids to replenish tears to maintain vision and so on. Of interest herein is the interaction between a moving boundary and particle-laden flows. The method developed in this thesis is validated against benchmark solutions involving moving boundaries and applied to the thorough analysis of fluid mechanics of mechanical heart valve closure. Specifically, the motion of blood cells (RBCs and platelets) through heart valves during the closure of the valves is analyzed in great detail. During this passage of the blood cells, large shears can be generated, particularly in the closure phase of valvular function, which can lead to lysis of the cells or activation of cells resulting in deleterious thrombus formation.

To investigate the physics of moving boundary – particle interaction in the mechanical heart valve during its closure, an advanced numerical technique is required that can simulate flows in the presence of moving boundaries, while also transporting particles. In this dissertation, a Cartesian grid approach is advanced for this purpose, with particular attention paid to efficiency of the solution procedure. The methodology is then applied to study the physics of interaction of particle-laden flows with moving solid walls. In the following section, the physiological and functional aspects of heart valves are reviewed to set the stage for the following chapters, where the method development and physics are covered.

1.2 Heart Valves

1.2.1 Structure and Function

The function of the heart valves is to prevent back flow of blood from the ventricles into the atria or from the aortic and pulmonary arteries into the ventricles [26] (see Figure 1.1). The valves open and close passively but rapidly under the action of fluid stresses. The thin leaflets of the valves withstand very high repetitive loads for billions of cycles during the human lifetime. There are four heart valves, which ensure that the blood flows in one direction and which play a vital role in maintaining the normal cardiac output and pressures throughout the body. The tricuspid valve is located between the right atrium and ventricle while the pulmonic valve is located between the right ventricle and main pulmonary artery. The mitral valve is located between the left atrium and ventricle while the tricuspid aortic valve is located between the left ventricle and aorta. The structure of the valves is shown in Figure 1.1(b). The left side of the heart is the high

pressure side and valvular heart diseases predominantly occur in this side of the heart [26].

The aortic or the semi-lunar valve consists of three thin crescent shaped leaflets of about 0.1 mm thickness. In the closed position, aortic valve leaflets coapt and seal the aortic orifice. The leaflets are light and open and close passively due to the pressure gradient across the valves [26].

The bileaflet mitral valve has two leaflets, which are thinner than those of the aortic valve and form an elliptical orifice. The leaflets are attached to the papillary muscles by chordea tendinea. In the fully open position, the upper portion of the mitral valve resembles a funnel. When closed, the free edges of the leaflets are pressed against each other and the papillary muscles prevent the cusps from falling back into the atrium during ventricular contraction [26].

Valvular diseases are caused by many factors. Rheumatic valvular disease, aging, stenosis are some of the common problems with valves. Stenotic (stiffening of the leaflets) valves require higher pressure gradients for the valves to open. This can put higher stress on the heart. Even in fully open position, the valve orifice area may be small resulting in high pressure loss through the valves. The valves can also become incompetent, resulting in increased back flow (aortic or mitral insufficiency) [26].

1.2.2 Optimal Replacement Valve

Valve replacement is a common form of treatment for valvular diseases. Approximately 250,000 prosthetic heart valves are implanted each year throughout the world [163, 164]. Despite years of research, problems associated with heart valve prostheses have not been eliminated and new designs continue to be developed.

An optimal heart valve needs to allow the patient to lead a relatively normal life. The ideal prosthetic valve should be able to mimic the functional characteristics of the natural valves. The valves should open with minimum pressure gradient across the valve and minimum back flow. The material should be durable, compatible with biological tissues and anatomic functions. The material used in making the prostheses should be durable, non-toxic and non-thrombogenic so that long term anticoagulant therapy is not required after implantation. The prosthetic valve should also be easy to implant surgically. The operation of the valve should be relatively silent and the stresses developed because of the presence of the valve should not damage formed elements like red blood cells and platelets. Blood is a corrosive and chemically unstable fluid, which tends to clot (thrombus formation) in the presence of foreign bodies because of platelet activation. The contact between the valve and the walls of the vessel should be avoided except along the fixation rim. The geometry of the valve should be such that stagnation zones and regions of relative stasis are avoided or minimized. The prosthetic valve should also be easily manufactured, readily available and inexpensive.

1.2.3 Types of Prosthetic Valves

The prosthetic valves currently available today can be broadly classified into two categories: (i) mechanical valves; and (ii) bioprostheses (tissue valve prostheses). Some of the more commonly used valves are shown in Figure 1.2. Different geometries have been tried for mechanical valves. Freshly explanted natural aortic valves from human cadavers, known as homografts or allografts have also been successfully used as replacement valves. These are the only valves entirely consisting of biological material and sewn into place. However, human tissue valves are difficult to obtain and hence not

very viable. Porcine and bovine valves after treatment have also been successfully implanted. Figure 1.3(a) shows the detailed view of a bi-leaflet valve which is the focal point of the current work.

1.2.4 Problems with Implanted Valves

Many patients with implanted valves lead a relatively normal life. However, there are still many problems associated with implanted valves. Some of the problems associated with implantation are: (i) thrombus formation due to platelet deposition (ii) mechanical failure due to fatigue or changes in the valve material; (iii) leakages (iv) infection (v) mechanical damage to blood including hemolysis, destruction of platelets, and protein denaturation; (vi) mechanical damage to the walls of the vessels and (vii) tissue overgrowth. The most serious problems with current mechanical valves are thrombosis and thromboembolism as shown in Figure 1.3(b), anticoagulant-related hemorrhage, tissue overgrowth, infection, paravalvular leak and valve failure owing to material fatigue or chemical change [19, 25, 55]. Patients with implanted mechanical valves are treated with chronic anticoagulant or blood thinning therapy throughout their life. Some patients cannot tolerate chronic anticoagulant therapy due to the risk of bleeding complications. In these patients, a bioprosthetic valve implant is used. Long term anticoagulant therapy is not required with bioprostheses but tissue calcification and degeneration of the leaflets are common. Replacement of the valve is required after an average period of about seven to ten years [26].

Thrombus formation is caused mainly due to deposition and aggregation of activated platelets. Platelets are the smallest corpuscular constituents of human blood with concentration in the range of 150,000 to 300,000 per mm^3 of blood. Platelet

activation may be the result of an imbalance in the homeostatic system due to chronic blood damage and the destruction of formed blood elements due to non-physiological pressure and shear forces in the flow dynamics through the mechanical heart valve. Unlike compliant native tissue valves, mechanical valves are essentially rigid structures which cause high stress and pressure build-up. Direct mechanical trauma by impact with the valve housing and other support structures such as hinges and local flow induced stresses may be two other mechanisms responsible for the deleterious effects on blood components [163].

1.2.5 Fluid Mechanics

Flow velocity and shear stress fields can be significantly different for various prosthetic heart valve designs. Elevated levels of shear stress lead to lethal damage to blood cells as well as platelet activation. Platelets have been shown to be activated when subjected to shear stresses of about $10\,Pa$ [67] and this will also be further affected by presence of foreign surfaces of the valve housing and leaflets. This critical stress level is also affected by the residence time of the cell in the damaging fluid environment, which further complicates the damage mechanism [16-18, 163, 164]. Furthermore, the regions of flow stagnation and flow separation that occur adjacent to the valves could promote the deposition of damaged blood elements, leading to thrombus formation on the prosthesis.

In bi-leaflet valves, thrombus formation is mostly observed in the hinge region and also on the valve housing [45-47, 66]. It is hypothesized that the local flow conditions in these regions contribute to the thrombus formation. All the studies on bi-leaflet valves, experimental as well as computational, report flow separation and vortex

shedding at both ends of the valve leaflets [17, 95]. The high velocity leakage flow generates regions of high shear stress which are likely sites for platelet activation. The activated platelets have high residence times in the valve vicinity, when caught in regions of stagnant flow or re-circulation regions which are the likely sites of thrombus formation [156, 157].

1.2.6 Computational Modeling

The development of computational models of the fluid mechanics in heart valves is motivated by medical and economic concerns as well as efficiency [164]. Accurate and descriptive computational models are a versatile, safe, non-invasive and cost-effective tool for scientific and medical research. Design changes can be easily tested and validated in a cost-effective way before prototype development and experimental assessment. However, hardware and software constraints force many simplifying assumptions on the computations.

One significant difficulty in modeling the heart valve mechanics is the wide disparity in length scales from the large-scale motions with length scales in the valve orifice ($\sim 25\,mm$) to the leakage flow through the gap ($\sim 0.4\,mm$) to the flow in the hinge regions of mechanical valves ($\sim 100\,\mu m$). To capture the flow dynamics around the heart valve accurately, the above length scales need to be resolved adequately. The leakage flow is a significant part of the valve dynamics that must be adequately resolved for meaningful flow analysis by employing state-of-the-art computational techniques.

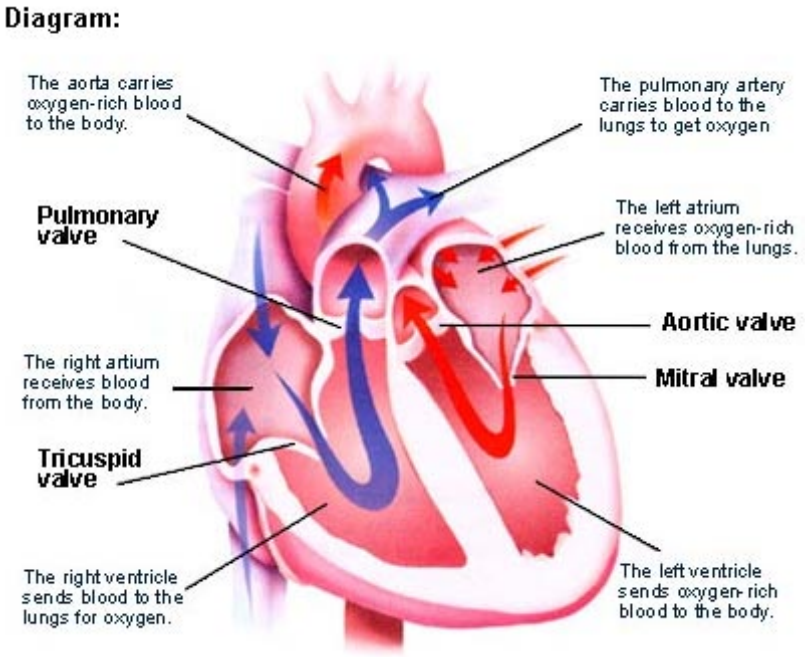
The other main feature that needs to be captured is the leaflet motion. The valve sweeps an angle of almost 64° in the process of valve closure. It is well established by

previous studies [33, 113] that flow patterns recorded from fixed valve models are entirely different from moving valve models. Use of body-fitted meshes may be able to adequately resolve the leakage flow by use of highly stretched meshes [15, 28, 33, 52, 53, 64, 75, 76, 80, 81, 85, 118, 119, 149, 161]. However, use of moving meshes presents additional complications of re-meshing, in particular the rather complex task of mesh generation to accommodate changes in geometry to prevent mesh skewness, entanglement, extreme disparity in cell sizes and aspect ratios and so on. All these factors impact negatively on the accuracy and robustness of body-fitted mesh schemes. The natural way of handling moving boundary problems by entirely obviating complexity of mesh generation is by using a Cartesian grid approach. In this method, calculations are performed on fixed rectangular meshes and the valve geometry is superimposed on it with modifications made in the discretization scheme to accommodate presence of the valve structures. Cartesian grid methods have been previously employed in modeling of heart valve dynamics [108, 109] but allowing sufficient resolution in the small gaps can make the computations extremely tedious because of the fine meshes that have to be used. A further consideration is that algorithms for solving the Navier–Stokes equations must employ at least second-order accurate numerics both in space and time to capture the complex flow dynamics near the heart valve. First-order methods are more stable but introduce excessive artificial viscosity into the numerical solutions and tend to smear out most scales of motion, except perhaps the largest ones, and could dramatically misrepresent hemodynamically relevant flow features—such as the intensity of regions of high shear, the existence of pockets of reversed flow and flow separation, etc [163].

1.3 Current Aims

The current work aims at developing an optimized computational model to predict regions of platelet activation by exposure to high shear stress and subsequent likely regions of deposition due to high residence time in these areas. The current focus is on the flow dynamics through the small gaps between the valve and the housing and its effects on platelet activation in the valve closure and rebound stages. While the significant shear stress buildup and platelet activation is expected to take place in the near-closure and rebound stages, the macro-scale flow dynamics of the closure stage will determine the flow features, such as separated boundary layers and shear layers, observed when the valve is near closure. The incorporated local mesh refinement enables calculation of macro-scale closure dynamics as well as more detailed flow features of the leakage flow regions. A second-order locally refined Eulerian Sharp Interface Levelset based Cartesian Grid flow solver discussed in Chapters 2 and 3 is employed with platelets being modeled as point particles in a Lagrangian particle tracking algorithm. The time history of platelet exposure to shear stress is tracked to demarcate areas of high likelihood of platelet activation and deposition in the housing region. Chapter 4 describes validation and testing of the current method. Chapters 5 and 6 present detailed analysis of the fluid mechanics of heart valve closure while Chapter 7 concludes the thesis with suggestions for future work.

(a)



(b)

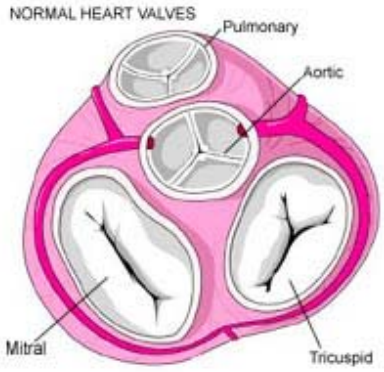


Figure 1.1. Heart valves, structure. (a) Heart with parts marked in detail. (b) Structure and location of the heart valves.

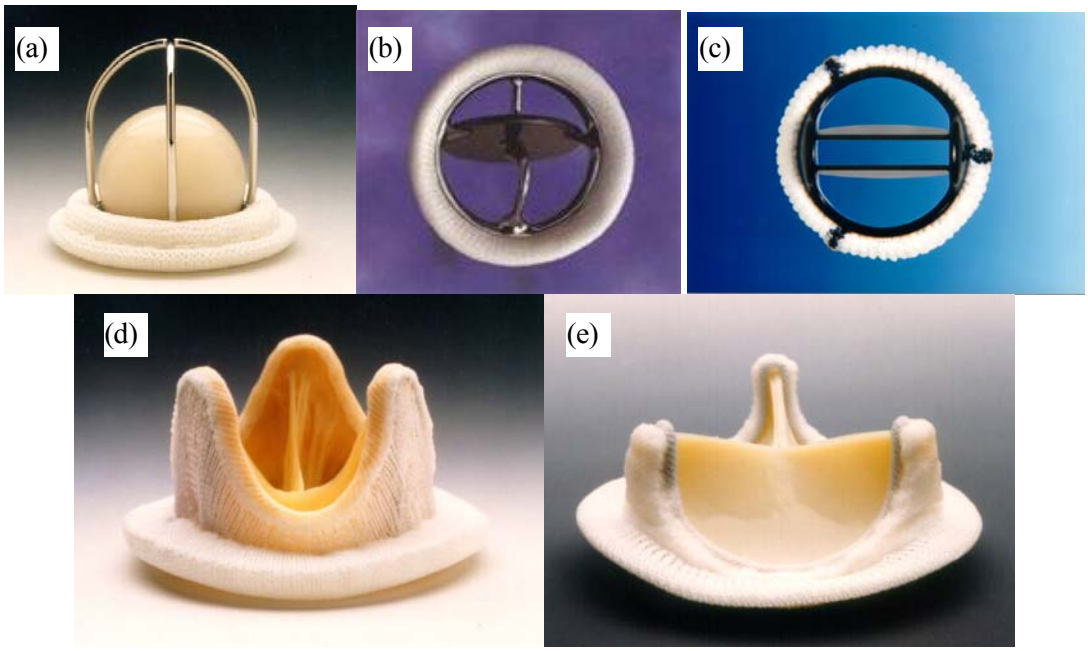


Figure 1.2. Various types of replacement heart valves. (a) The ball and cage valve, (b) the tilting disc valve, (c) the bi-leaflet valve, (d) and (e) bio-prosthetic porcine heart valves.

(a) (b)

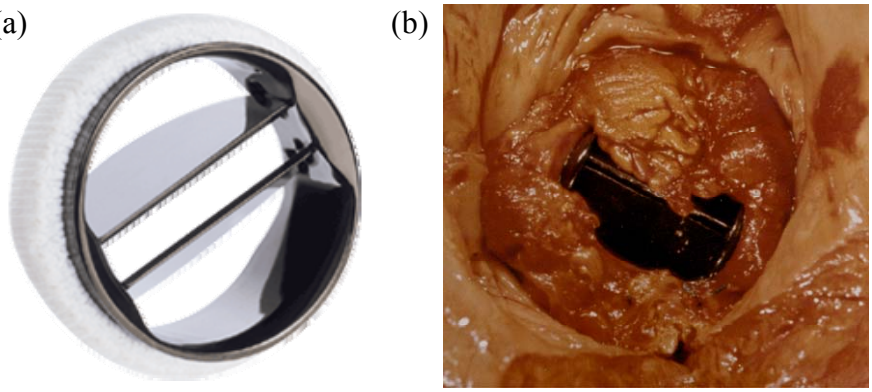


Figure 1.3. The bi-leaflet mechanical valve. (a) Structure, (b) thrombus formation and blockage of valve.

CHAPTER 2

NUMERICAL METHODOLOGY

This chapter and the next describe the evolution of the computational method used to model the flow through mechanical heart valves in the closure phase. As stated earlier, the method must be able to solve moving boundary problems easily allowing adequate resolution in small spaces to capture the flow dynamics over a range of length scales.

2.1 Introduction to Cartesian Grid Methods

The main advantage of Cartesian grid methods is the ease of mesh generation. The computational domain is typically rectangular in 2D and cuboid shaped in 3D. Rectangular meshes are generated conforming to the rectangular domain and any immersed objects or arbitrary boundaries are allowed to cut across the meshes as demonstrated in Figure 2.1. The figure shows the set-up of a problem involving calculation of flow dynamics in the stomach. Cartesian grid methods can be broadly classified in two categories:

In the first category are methods where the interface effects are transmitted through forcing functions. The most commonly used fixed grid approach for moving boundary computations involving solid-fluid as well as fluid-fluid interfaces is the immersed boundary method introduced by Peskin [107]. In the original immersed boundary method as well as later versions, the interaction of the boundary with the fluid is effected using smoothed delta-functions located at the boundaries. The effect of the boundaries, in particular boundary forces (such as elastic forces in structures, tethering

forces and surface tensions) are transmitted to the momentum equations as source (or forcing) terms as shown in the equation below:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}.\vec{\nabla}\vec{u} = -\frac{1}{\rho}\vec{\nabla}p + \upsilon\nabla^2\vec{u} + \vec{f} \tag{2.1}$$

In the above equation, $\vec{f}$ is the forcing function that represents the effect of the immersed boundary. Typically, a numerical δ-function with a support of a few cells is used to convert singular surface forces (such as surface tension) into volume forces $\vec{f}$. The strategy of transmitting interface effects to the flow field through source terms has also been used to solve problems in multiphase flows [97, 138, 145] and solidification [4]. However, one shortcoming of these methods is that discontinuities at the immersed boundary are smeared across a few cell widths. Methods where the interface properties are smeared are termed as diffuse interface methods.

It has been shown [89] that such smearing can adversely impact the accuracy of solutions when the boundary motion is closely coupled with the evolution of surrounding fluid flow. There are also issues involved with stability and stiffness of the computations [30, 124], particularly when the embedded objects deform along with the flow. Improvements to the delta-function based immersed boundary methods have appeared in the literature in recent years [84, 87, 112]. In the finite-element setting the fictitious domain method [1, 56, 106] and the immersed finite element method [151] have followed this idea of transmitting the boundary forces to the fluid using an interaction source term. However, a number of papers on immersed boundary methods in recent years have deviated from the use of delta-functions in transmitting boundary forces to the fluid. For example, in the finite difference/ finite volume methods presented in several recent

papers [9, 48, 78, 139], the idea of using a forcing term in the momentum equation has been retained. These (non-smoothed) forces are placed at points that adjoin the immersed boundary (either inside or outside the solid object) in order to impose the appropriate velocity boundary conditions on the solid surface. Thus, unlike the original immersed boundary method of Peskin and its derivatives, these new immersed boundary methods are in fact sharp interface methods.

In the second category are methods where the interface effects are included in the discrete spatial operators. These methods do not use forcing terms but incorporate the presence of the embedded boundaries into the discrete form of the Navier-Stokes equations. Thus, in contrast with the above approach, the momentum equation is retained in the form:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}.\vec{\nabla}\vec{u} = -\frac{1}{\rho}\vec{\nabla}p + \upsilon\nabla^2\vec{u} \tag{2.2}$$

During the discretization procedure however the spatial differential operators $(\vec{\nabla}, \nabla^2)$ in the equation are constructed at points that adjoin the interface in such a way that the interfacial jump conditions are incorporated. Examples of this class are the immersed interface method [83, 90, 91], the sharp interface method [142-144], the ghost fluid method [49, 93] and the XFEM method [31, 32, 43, 130].

The immersed interface method (abbr. IIM) [90] enables a sharp interface treatment by casting the governing equations in a coordinate system with axes oriented with the local normal and tangent to the interface. Problems involving embedded fluid-fluid interfaces with singular sources and jumps in material property across the interface have been solved using this approach [90, 91]. The method seeks to preserve second-

order accuracy at points adjacent to the interface as well as away from it. In Ghost Fluid Method (abbr. GFM) [49] the governing equations are discretized along the Cartesian coordinate directions. Only first-order accuracy is demanded at interface-adjacent points. In both IIM and GFM, jumps and singular sources at the interface are incorporated into the discrete operators in the transport equations.

In the sharp interface method [143, 160] a finite volume technique is used to discretize the equations within the domains separated by the embedded boundary in such a way that information is not smeared at the immersed boundary. This requires reshaping of the control volumes through which the interface passes and the integration of the weak form of the governing equations over non-rectangular control volumes. Second-order accuracy is maintained at bulk as well as interface-adjacent grid points. Problems involving fluid-structure interactions [143, 144] and solidification [140-142, 146] have been solved using this approach. In the finite element community, the XFEM method [131] follows a similar strategy, in that the elements through which the boundary passes are enriched (i.e. these elements are subdivided and conform locally to the immersed boundary; in other words, degrees of freedom are added) to facilitate integration of the weak form of the governing equations.

Traditionally, diffuse interface methods have been considered to be simpler to implement and therefore have found wide usage [5]. In such methods the onus is on designing the source terms so that the formulation approaches the sharp interface limit for vanishing interface thicknesses. Although straightforward to formulate, sharp interface methods have been considered to be more difficult to implement and therefore their use has been restricted to some specific situations and to a few practitioners. A finite-

difference based generalized sharp interface formulation on Cartesian grids is formulated in this chapter. This scheme based on interface dependent switch functions is easy to implement and applicable to a wide variety of problems involving solid-fluid and fluid-fluid interfaces.

2.2 The Current Method

The present thesis uses a Levelset [105] based sharp-interface Cartesian grid method to model moving boundary problems. Methods where the interface effects are included in the discrete spatial operators are called sharp interface methods. The critical issues that arise in developing sharp interface Cartesian grid methods for moving boundaries problems are:

1. Representation of embedded interfaces: Explicit surface tracking (involving surface meshing and re-meshing) can be challenging for complex moving boundaries, particularly in the presence of sharp edges, cusps, instabilities and topological changes in the boundaries. Geometric details such as intersections between the triangulated surface mesh and the underlying flow solver mesh need to be computed repeatedly. Normal and curvature computations need to be performed accurately on the surface. The Levelset technique presents a solution to many of these problems due to the implicit interface representation and built-in regularization due to the entropy-satisfying solutions obtained from Levelset advection. In this approach, at each mesh point in a narrow band surrounding the interface the signed normal distance to the interface is stored. The interface is implicitly contained in this information as the zero-level surface and can be deduced to desired accuracy. Representation of all interfaces (whether solid-fluid or

fluid-fluid) using Levelsets simplifies discretization at computational points adjacent to the interfaces.

2. Solving for flows around the immersed boundaries: Since interfaces cut through the mesh in arbitrary fashion finite volume discretization requires reshaping of the cells cut by the interfaces. This can be avoided by employing a finite-difference discretization of the strong form of the governing equations. In adopting the finite-difference approach, the main concern is the accuracy and conservation properties of the flow solver. Fortunately, for the Cartesian grid approach the deviation of the finite-difference approach from the finite-volume approach appears only at the grid points adjoining the interface. Here, a second-order accurate Cartesian grid based finite-difference scheme is used to discretize the incompressible Navier-Stokes equations. The discretization depends essentially on convolving the differential operators with the distance function field. Thus the present algorithm handles embedded solid-fluid interfaces (using the sharp-interface method detailed in this chapter) and fluid-fluid interfaces (using the Ghost Fluid method) [93] in a unified fashion.

2.2.1 Equations to be Solved

The governing equations for incompressible flow are:

$$\vec{\nabla}.\vec{u} = 0 \tag{2.3}$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\vec{\nabla}\vec{u} = -\vec{\nabla}p + \frac{1}{\mathrm{Re}}\nabla^2\vec{u} \tag{2.4}$$

In the above equation, $\mathrm{Re} = \rho_0 U_0 D/\mu$ is the Reynolds number. ρ_0, U_0, D and μ are the density, characteristic velocity, characteristic length and viscosity respectively.

The scalar (temperature, species concentration) transport equations take the general form:

$$\beta_t \frac{\partial \xi}{\partial t} + \vec{\nabla} \cdot \vec{u}\xi = \beta_d \nabla^2 \xi \tag{2.5}$$

where β_t and β_d are material constants associated with the time-dependent and diffusive terms respectively.

2.2.2 The Flow Solver

A cell-centered collocated arrangement of the flow variables is used to discretize the governing equations. A two-step fractional step method [160, 165] is used to advance the solution in time. The first step evaluates an intermediate velocity by solving an unsteady advection-diffusion equation.

$$\frac{\vec{u}^* - \vec{u}^n}{\Delta t} = -\vec{u} \cdot \vec{\nabla}\vec{u} + \frac{1}{\mathrm{Re}} \nabla^2 \vec{u} \tag{2.6}$$

where the intermediate velocity $\vec{u}^*$ is evaluated with central-difference discretization schemes for convection and diffusion terms. The convective terms are treated explicitly and discretized using a second-order accurate Adams-Bashforth method:

$$\vec{u} \cdot \vec{\nabla}\vec{u} = \frac{1}{2}\left(3\vec{u}^n \cdot \vec{\nabla}\vec{u}^n - \vec{u}^{n-1} \cdot \vec{\nabla}\vec{u}^{n-1}\right) \tag{2.7}$$

The diffusion terms are treated semi-implicitly using Crank-Nicholson scheme:

$$\frac{1}{\mathrm{Re}} \nabla^2 \vec{u} = \frac{1}{2\,\mathrm{Re}}\left(\nabla^2 \vec{u}^* + \nabla^2 \vec{u}^n\right) \tag{2.8}$$

The second fractional-step involves the correction of the intermediate velocity field $\vec{u}^*$ to

enforce mass conservation:

$$\frac{\vec{u}^{n+1}-\vec{u}^{*}}{\Delta t}=-\vec{\nabla}p \tag{2.9}$$

where the pressure field p is evaluated to impose a divergence-free velocity field at time step $n+1$. This is done by taking the divergence of Equation (2.4) to obtain a Poisson equation for pressure:

$$\nabla^2 p=\frac{\vec{\nabla}.\vec{u}^{*}}{\Delta t} \tag{2.10}$$

The final semi-discrete form of the equations including each of the above discretization schemes is as follows:

$$\frac{\vec{u}^{*}}{\Delta t}-\frac{1}{2\,\mathrm{Re}}\nabla^2\vec{u}^{*}=\frac{\vec{u}^{n}}{\Delta t}+\frac{\rho\vec{g}}{Fr^2}+\frac{1}{2\,\mathrm{Re}}\nabla^2\vec{u}^{n}-\frac{1}{2}\left(3\vec{u}^{n}.\vec{\nabla}\vec{u}^{n}-\vec{u}^{n-1}.\vec{\nabla}\vec{u}^{n-1}\right) \tag{2.11}$$

$$\nabla^2 p^{n+1}=\frac{\vec{\nabla}.\vec{u}^{*}}{\Delta t} \tag{2.12}$$

The intermediate velocity is then corrected to obtain the final divergence-free velocity field:

$$\vec{u}^{n+1}=\vec{u}^{*}-\Delta t\vec{\nabla}p \tag{2.13}$$

The advection-diffusion equations for scalar (heat and species) transport are discretized in a similar manner, i.e.:

$$\beta_t\frac{\xi^{n+1}}{\Delta t}-\frac{\beta_d}{2}\nabla^2\xi^{n+1}=\beta_t\frac{\xi^{n}}{\Delta t}+\frac{\beta_d}{2}\nabla^2\xi^{n}-\frac{1}{2}\left(3\vec{u}^{n}.\vec{\nabla}\xi^{n}-\vec{u}^{n-1}.\vec{\nabla}\xi^{n-1}\right) \tag{2.14}$$

2.2.3 Implicit Interface Representation Using Levelsets

Embedded surfaces are represented implicitly on the mesh using a standard Levelset approach [105, 116, 117]. In addition to the flow variables, the Levelset method advects a scalar field ϕ_l, where subscript l denotes l^{th} embedded interface. The value of ϕ_l at any point is the signed normal distance from the l^{th} interface with $\phi_l < 0$ inside the immersed boundaries and $\phi_l > 0$ outside. The interface location is implicitly embedded in the ϕ_l-field since the $\phi_l = 0$ contour represents the l^{th} immersed boundary.

In case of moving interfaces, the motion of the boundary is tracked by advecting the Levelset using:

$$(\phi_l)_t + \vec{V}_l . \vec{\nabla} \phi_l = 0 \tag{2.15}$$

where $\vec{V}_l$ is the l^{th} Levelset velocity field which is derived directly from the physics of the problem. A fourth-order ENO scheme in space and fourth-order Runge-Kutta integration in time are used for the evolution of the Levelset field. Since $\vec{V}_l$ is prescribed by the physics only on the interface (i.e. on the zero-Levelset), the value of velocity at the grid points that lie in the narrow band around the zero-Levelset needs to be obtained. This is done by extension of the interfacial velocity [116] away from the front using:

$$\psi_\tau + \vec{V}_{ext} . \vec{\nabla} \psi = 0 \tag{2.16}$$

where ψ is any quantity (such as interface velocity components $(\vec{V}_l)_x$ or $(\vec{V}_l)_y$) that needs to be extended away from the interface. A choice for the extension velocity is

$\vec{V}_{ext} = sign(\phi_l)\frac{\nabla\phi_l}{|\nabla\phi_l|}$. This populates the narrow band around each interface in the time $\tau = O(\Delta x)$ with a Levelset velocity that has been extended outward from the interface in a direction normal to it. A reinitialization procedure [132, 133] is carried out after Levelset advection to return the ϕ-field to a signed distance function, i.e. to satisfy $\left|\vec{\nabla}\phi_l\right| = 1$. Suppose $(\phi_l)_0$ is the Levelset field prior to re-initialization. The following equation is solved to steady state to re-initialize the Levelset field.

$$(\phi_l)_\tau + \vec{w}\cdot\nabla\phi_l = sign(\phi_l) \tag{2.17}$$

$$\vec{w} = sign((\phi_l)_0)\frac{\nabla(\phi_l)_0}{|\nabla(\phi_l)_0|} \tag{2.18}$$

where $sign((\phi_l)_0) = \frac{(\phi_l)_0}{\sqrt{(\phi_l)_0{}^2 + (\Delta x)^2}}$ with the initial condition $\phi_l(\vec{x},0) = (\phi_l)_0(\vec{x})$. The calculation of normal and curvature of the interface from the Levelset field is simple. The normal ($\vec{n}$) and curvature (κ) are given by:

$$\vec{n} = \vec{\nabla}\phi_l \Big/ \left|\vec{\nabla}\phi_l\right| \tag{2.19}$$

$$\kappa = -\vec{\nabla}.\vec{n} \tag{2.20}$$

2.3 Discretization of Operators

2.3.1 Classification of Grid Points

The grid points on the Cartesian mesh are classified as bulk points and interfacial points. Bulk points are those points that lie away from the interface and interfacial points

are those that lie immediately adjacent to the immersed interface. Figure 2.2(a) illustrates an immersed interface and the points that are classified as interfacial points, i.e. points that satisfy the condition $(\phi_l)_{i,j}(\phi_l)_{nb} \le 0$, where nb denotes an immediate neighbor along the coordinate directions. The filled circles represent the interface points in the figure. The discrete operators at the interfacial points are different from those that apply at the bulk points. The strategy adopted to deal with this situation for the differential operators in the governing equation are discussed below.

2.3.2 Discretization at Bulk Points

A standard 5-point central-difference stencil applies for a typical bulk point shown in Figure 2.2(b) (i.e. a grid point that does not adjoin the embedded interface) in a two-dimensional (2D) Cartesian mesh. While only two-dimensional situations are shown in the figures for ease of visualization, the discussion below carries over to three-dimensions (3D). The three-dimensional counterpart would involve a 7-point stencil and the discretization for the momentum equations is identical in all the three dimensions as described below.

The second derivative with reference to x in the diffusion term is discretized as follows:

$$\frac{\partial^2 \psi}{\partial x^2} = \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x^2} - \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta x^2} \tag{2.21}$$

while the convection term in the x-direction is obtained as:

$$\frac{\partial u\psi}{\partial x} = \frac{u_{i+1/2,j}\psi_{i+1/2,j} - u_{i-1/2,j}\psi_{i-1/2,j}}{\Delta x} \tag{2.22}$$

where u is the x-component of the velocity and

$$\psi_{i\pm1/2,j} = \frac{\psi_{i\pm1,j} + \psi_{i,j}}{2} \tag{2.23}$$

$u_{i\pm1/2,j}$ are the velocities on the cell faces. Similar considerations apply along the y and z directions.

2.3.3 Discretization for Solid-Fluid Interfaces

As pointed out before, the main challenge in sharp interface fixed-grid methods is to accurately impose interfacial conditions in the discrete system of equations. Moreover, a sharp-interface method demands one-sided discretization for all the partial derivatives to avoid smearing of the interfaces. For the case of immersed solid-fluid boundaries, as in fluid-structure interaction problems, the no-slip and no-penetration velocity boundary conditions are applied on the solid surfaces. For small Strouhal numbers the Neumann condition for pressure applies on such boundaries [141]. These boundary conditions are then supplied to the governing equations through the discretization at the interfacial points as described below.

2.3.3.1 $\partial^2\psi/\partial x^2$ with a Dirichlet boundary condition

Second derivatives need to be computed in the diffusion terms in the momentum as well as scalar transport equations and in these cases typically a Dirichlet condition applies at the boundary. With particular reference to the point (i, j) in Figure 2.2 (c), this point lies in the fluid and the velocities, scalars (temperature and species concentration) and pressure are computed there. Note that for points that lie in the solid only the

temperature and species fields are computed. In discretizing $\partial^2\psi/\partial x^2$ at point (i, j) in Figure 2.2(c), the neighbor point $(i+1, j)$ lies across the interface in the solid and hence cannot be used in discretization. In order to include the interfacial values (i.e. apply interfacial conditions), it is necessary to find the location where the zero-Levelset intersects the line joining the cell-centers (indicated by the square symbols in Figure 2.2(c)). In the following expressions for the coefficients the quantity $\chi = \frac{\Delta x_I}{\Delta x}$ (see Figure 2.2(c)) will be used frequently. Here, Δx_I is the distance between the cell center and the intersection of cell centerlines with the interface (filled square) and Δx is the nominal cell width. By noting that the intersection point has a zero Levelset value, χ can be easily evaluated using the Levelset information at (i, j) and $(i+1, j)$:

$$\chi = \frac{\Delta x_I}{\Delta x} \cong \left| \frac{(\phi_l)_{I_x} - (\phi_l)_{i,j}}{(\phi_l)_{i+1,j} - (\phi_l)_{i,j}} \right| = \left| \frac{0 - (\phi_l)_{i,j}}{(\phi_l)_{i+1,j} - (\phi_l)_{i,j}} \right| \tag{2.24}$$

Note that in evaluating the quantity χ a linear profile is assumed for the distance function between adjacent grid points. Higher-order approximations of the geometry can be implemented as well [34, 129]. The second-derivative can be estimated to second-order accuracy using the form:

$$\frac{\partial^2\psi}{\partial x^2} = \alpha_I \psi_I + \alpha_{i,j}\psi_{i,j} + \alpha_{i-1,j}\psi_{i-1,j} + \alpha_{i-2,j}\psi_{i-2,j} \tag{2.25}$$

Here, ψ_I is the value on the interface at the location I_{+x} with reference to Figure 2.2 (c). Using Taylor series expansions for each of $\psi_I, \psi_{i-1,j}, \psi_{i-2,j}$ about point (i, j) and

further demanding that $\partial^2\psi/\partial x^2$ be estimated to $O(\Delta x^2)$ yields the following expressions for the coefficients.

$$\alpha_I = 6/(\chi(\chi+1)(\chi+2)\Delta x^2) \tag{2.26}$$

$$\alpha_{i-1,j} = (4-2\chi)/((\chi+1)\Delta x^2) \tag{2.27}$$

$$\alpha_{i-2,j} = (\chi-1)/((\chi+2)\Delta x^2) \tag{2.28}$$

$$\alpha_{i,j} = -\alpha_I - \alpha_{i-1,j} - \alpha_{i-2,j} \tag{2.29}$$

In practice the above implementation may present difficulties due to the singular behavior of α_I as $\chi \to 0$. Therefore for small values of χ (< 0.01), i.e. when the interface is very close to the mesh point, the value χ is replaced by $\max(\chi, 0.01)$. This involves a slight perturbation of the boundary within a grid cell and decreases the order of accuracy locally from second-order. However this situation arises at only a few mesh points and the global accuracy is not impacted. Note that a first-order approximation is given by:

$$\alpha_I = 2/(\chi(1+\chi)\Delta x^2) \tag{2.30}$$

$$\alpha_{i-1,j} = 2/((\chi+1)\Delta x^2) \tag{2.31}$$

$$\alpha_{i,j} = -\alpha_I - \alpha_{i-1,j} \tag{2.32}$$

$$\frac{\partial^2\psi}{\partial x^2} = \frac{2}{\chi(1+\chi)}\frac{(\psi_I - \psi_{i,j})}{\Delta x^2} - \frac{2}{(1+\chi)}\frac{(\psi_{i,j} - \psi_{i-1,j})}{\Delta x^2} \tag{2.33}$$

The singularity with respect to χ remains in this case as well. However, positivity of the off-diagonal coefficients is maintained in the first-order case while the second-

order form will lead to a negative coefficient $\alpha_{i-2,j}$. In practice this negatively impacts the convergence of the iterative solver used for solving the discrete system of equations; to maintain robustness a first-order treatment for the diffusion term is employed at the interfacial points in the present calculations. While this practice lowers the order of approximation in the lower-dimensional set of interfacial cells, global second-order accuracy is still maintained as shown in the results.

2.3.3.2 ($\partial(u\psi)/\partial x$ and $\partial\psi/\partial x$) with Dirichlet conditions

Similar to the second-derivative terms above, the discretization scheme for $\partial(\boldsymbol{u}\psi)/\partial\boldsymbol{x}$ consists of contributions from points in the same phase. The differential operator for the convection term is obtained in the following form:

$$\frac{\partial u\psi}{\partial x} = \lambda_I (u\psi)_I + \lambda_{i-1/2,j}(u\psi)_{i-1/2,j} + \lambda_{i-3/2,j}(u\psi)_{i-3/2,j} \tag{2.34}$$

Note that to avoid pressure-velocity decoupling in the current collocated variable arrangement cell face velocities are also stored along with cell center velocities [165]. These cell-face velocities are used in evaluating the convective fluxes in Equation (2.34). By employing Taylor expansions for each of the $\psi_{i',j'}$ in Equation (2.34) about point (i, j) and demanding an O(Δx^2) scheme the following expressions are obtained for the constants:

$$\lambda_{i-1/2,j} = (2\chi - 3)/\big((2\chi + 1)\Delta x\big) \tag{2.35}$$

$$\lambda_{i-3/2,j} = (1 - 2\chi)/((3 + 2\chi\Delta x) \tag{2.36}$$

$$\lambda_I = 8/((4\chi^2 + 8\chi + 3)\Delta x) \tag{2.37}$$

The general form and the coefficients for a first-order scheme in this case are:

$$\frac{\partial u\psi}{\partial x} = \lambda_I (u\psi)_I + \lambda_{i-1/2,j} (u\psi)_{i-1/2,j} \tag{2.38}$$

$$\lambda_{i-1/2,j} = -2/((1+2\chi)\Delta x) \tag{2.39}$$

$$\lambda_I = 2/((1+2\chi)\Delta x) \tag{2.40}$$

The above general form can be rewritten as follows:

$$\frac{\partial u\psi}{\partial x} = \frac{2}{(1+2\chi)} \frac{\left((u\psi)_I - (u\psi)_{i,j}\right)}{\Delta x} \tag{2.41}$$

Since the convection terms are explicitly computed the second-order approximation can be employed except at those points where two opposing interfaces approach to within a mesh point. In this exigency the first-order approximation needs to be adopted.

2.3.3.3 $\partial^2\psi/\partial x^2$ operator with a Neumann boundary condition

This situation arises in the pressure Poisson equation for points that adjoin the embedded boundary. The stability and accuracy of the flow solver depends critically on the construction of this term. In fact, devising a discrete form for the Laplace operator with a Neumann condition on the immersed interface proved to be the key to the robustness of the overall flow solver.

From Figure 2.2(c) it is evident that when discretizing the above operator the

boundary condition that will apply at point I_{+x} is $\partial\psi/\partial n = 0$. If the immersed boundary is accelerating, then the interface boundary condition also accounts for the added mass effect. It is not immediately clear how such a Neumann condition can be incorporated into the discrete form of the above operator. Apart from being crucial to the robustness of the overall method, the treatment of the pressure boundary condition is a key distinguishing feature of the finite-difference approach adopted here as opposed to the finite-volume approach detailed in previous work [143]. In the latter, a weak form of the pressure Poisson equation was employed, i.e.

$$\oint \frac{\partial p}{\partial n} dS = \oint \frac{\vec{u}^* \cdot \vec{n}}{\Delta t} dS \tag{2.42}$$

The interfacial cells were reshaped into irregular shaped cells where one of the cell edges coincided with the interface. Due to the weak form above the Neumann boundary condition for pressure is easily incorporated by setting the interfacial contribution to zero (i.e. $\partial p/\partial n = 0$) implicitly in the discrete pressure Poisson equation. However, since the strong form is employed in the present finite-difference scheme on a Cartesian grid, one has to impose a Neumann boundary condition on the pressure in some way in the discrete Laplace operator. Several approaches were tried for discretizing terms such as $\partial^2\psi/\partial x^2$ with a Neumann condition on the immersed boundary and the one chosen for robustness is described below.

The method chosen is a robust, albeit first-order, approximation. The Laplace operator in the pressure Poisson equation is assembled as for the Dirichlet boundary condition case (i.e. Equation (2.33)). The interfacial pressure is found using the Neumann condition as follows. Looking at Figure 2.2 (d) the interface pressure can be estimated by

extending a normal from point I and placing two points distant Δx apart along the normal. The locations of the points $I_x, I1_x, I2_x$ on the probe are therefore,

$$\begin{aligned}
\vec{x}_{I_x} &= \vec{x}_{i,j} + \Delta x_I \vec{i} \\
\vec{x}_{I1_x} &= \vec{x}_{I_x} + \vec{N}_{I_x} \Delta x \\
\vec{x}_{I2_x} &= \vec{x}_{I_x} + 2\vec{N}_{I_x} \Delta x \\
\vec{x}_{Iy} &= \vec{x}_{i,j} + \Delta y_I \vec{j} \\
\vec{x}_{I1_y} &= \vec{x}_{I_y} + \vec{N}_{I_y} \Delta x \\
\vec{x}_{I2_y} &= \vec{x}_{I_y} + 2\vec{N}_{I_y} \Delta x
\end{aligned} \tag{2.43}$$

Fitting a quadratic to the pressure field along the normal erected and demanding that $\frac{\partial \psi}{\partial n} = 0$ at point I, one obtains:

$$\psi_{I_{x/y}} = \frac{4}{3}\psi_{I1_{x/y}} - \frac{1}{3}\psi_{I2_{x/y}} \tag{2.44}$$

In the above expressions the normal vector appears in several places. The normal at any point is easily obtained by bilinear interpolation from the values at the grid points. Note that the values of ψ at the points $I1_{x/y}$ and $I2_{x/y}$ are required in the above equation. These are calculated by bilinear interpolation from the surrounding mesh points. One issue that arises while evaluating the $\psi_{I1_{x/y}}$ and $\psi_{I2_{x/y}}$ is that for certain interface orientations the bilinear interpolation may involve a point that lies in the solid. Thus, a single layer of ghost values of pressure are computed and stored at the points in the interfacial points in the solid. The ghost values of pressure are also obtained with the

condition that the Neumann condition applies at the interface. Thus, a normal to the interface is erected from the ghost point as shown in Figure 2.2 (d) where the locations of the points on the normal are $\vec{x}_{G1} = \vec{x}_G + \vec{N}_G |\phi_G|$ and $\vec{x}_{G2} = \vec{x}_{G1} + \vec{N}_G \Delta x$ respectively.

The normal at point G is obtained from the Levelset field. Fitting a quadratic to the pressure field along the normal and demanding that $\frac{\partial \psi}{\partial n} = 0$ be satisfied at the point $G1$ (i.e. on the solid boundary) leads to:

$$\psi_G = \frac{\psi_{G1} d_2^2 - 2\psi_{G1} d_1 d_2 + \psi_{G2} d_1^2}{d_2^2 - 2d_1 d_2 + d_1^2} \tag{2.45}$$

where $d_1 = |(\phi_l)_G|$ and $d_2 = |(\phi_l)_G| + \Delta x$.

The interfacial value of the pressure ψ_{G1} is obtained from Equation (2.45) above and the value at $G2$ is obtained by bilinear interpolation. Note that the value of the ghost pressure and the interfacial pressure become inter-dependent through Equations (2.44) and (2.45). The determination of the interfacial pressure and ghost pressure are embedded within an iterative solver for the pressure and therefore the ghost and interfacial value of pressure converge along with the pressure field at the fluid computational points. This scheme works reliably for the entire range of Reynolds numbers and flow problems tested using the present solver. The above scheme applies in three-dimensions as well.

2.3.4 Discretization for Fluid-Fluid Interfaces

When a fluid-fluid interface is present, property and flow variable jump conditions make their appearance in the discrete form of the Laplace operator ($\nabla \cdot \beta \nabla \psi$) in the viscous terms in the momentum equation and in the pressure Poisson equation. The

jump conditions at the fluid-fluid interfaces manifest themselves primarily in the form $(\beta\psi_x)_x$. These arise in the viscous terms in the momentum equation (where ($\beta = \nu$) and the pressure Poisson equation (where $\beta = \frac{1}{\rho}$). Consider the picture in Figure 2.2(c) assuming a fluid-fluid interface separating the points (i, j) and $(i+1, j)$. The discretization of the Poisson-type term in the x-direction proceeds as follows:

$$(\beta\psi_x)_x = \frac{\left(\beta\frac{\partial\psi}{\partial x}\right)_{i+1/2\,j} - \left(\beta\frac{\partial\psi}{\partial x}\right)_{i-1/2\,j}}{\Delta x} \tag{2.46}$$

The jump conditions at the interface are:

$$[\psi] = a_{I_{+x}} = \psi^+_{I_{+x}} - \psi^-_{I_{+x}} \tag{2.47}$$

$$[\beta\psi_x] = b_{I_{+x}} = (\beta\psi_x)^+_{I_{+x}} - (\beta\psi_x)^-_{I_{+x}} \tag{2.48}$$

In discrete form, following the GFM approach [93, 104] the second jump condition can be written as:

$$\beta^+\left(\frac{\psi_{i+1,j} - \psi^+_{I_{+x}}}{(1-\chi)\Delta x}\right) - \beta^-\left(\frac{\psi^-_{I_{+x}} - \psi_{i,j}}{\chi\Delta x}\right) = b_{I_{+x}} \tag{2.49}$$

This involves a first-order estimate of the gradients on each side of the interface. In fact the GFM [93, 104] approach is identical to the IIM [91] if the Taylor expansions in IIM are carried to first-order only and the second-derivative jumps are ignored. Using the first jump condition in Equation (2.47),

$$\beta^{+}\left(\frac{\psi_{i+1,j}-\bar{\psi}_{I_{+x}}-a_{I_{+x}}}{(1-\chi)\Delta x}\right)-\beta^{-}\left(\frac{\bar{\psi}_{I_{+x}}-\psi_{i,j}}{\chi\Delta x}\right)=b_{I_{+x}} \tag{2.50}$$

This gives:

$$\begin{aligned}\bar{\psi}_{I_{+x}} &= \frac{\beta^{+}\chi}{(\beta^{+}\chi+\beta^{-}(1-\chi))}\psi_{i+1,j}+\frac{\beta^{-}(1-\chi)}{(\beta^{+}\chi+\beta^{-}(1-\chi))}\psi_{i,j} \\ &-\frac{\beta^{+}\chi}{(\beta^{+}\chi+\beta^{-}(1-\chi))}a_{I_{+x}}-\frac{(1-\chi)\chi\Delta x}{(\beta^{+}\chi+\beta^{-}(1-\chi))}b_{I_{+x}}\end{aligned} \tag{2.51}$$

Therefore, using this interfacial value in Equation (2.49) one obtains:

$$(\beta\psi_x)_x=\frac{\hat{\beta}}{\Delta x^2}\left(\psi_{i+1,j}-\psi_{i,j}\right)-\frac{\beta^{-}}{\Delta x^2}\left(\psi_{i,j}-\psi_{i-1,j}\right)-\frac{\hat{\beta}}{\Delta x^2}a_{I_{+x}}-\frac{(1-\chi)}{\Delta x}\frac{\hat{\beta}}{\beta^{+}}b_{I_{+x}} \tag{2.52}$$

$$\hat{\beta}=\frac{\beta^{+}\beta^{-}}{(\beta^{+}\chi+\beta^{-}(1-\chi))} \tag{2.53}$$

Similarly, the expression for the case where the interface lies between points (i,j) and $(i-1,j)$ can also be obtained.

2.3.5 The General Expression

The general algorithm for discretization of the Navier Stokes' equations by the sharp interface method on an Eulerian mesh includes both solid-fluid [96] and fluid-fluid interfaces [92] discussed in the previous sections. Figure 2.3 shows all the likely situations that can arise in the solution of a general moving boundary problem involving fluid-fluid and solid-fluid interfaces. The situation in Figure 2.4 can be considered to be the general case encompassing the cases shown in Figure 2.3(a)-(d) involving both solid-

fluid and fluid-fluid interfaces. Therefore, based on the expressions for the sharp interface method obtained above obtained for the above two cases and considering that Case 4 shown in Figure 2.3(d) corresponds to Case 1 shown in Figure 2.3(a) when the unresolved sliver of the Fluid 1 phase is ignored, a general discrete form for the discrete Laplace operator, $(\beta_x)_x$ can be obtained. The following expressions apply where multiple (say $L_{\max}$) embedded boundaries are present in the flow.

$$(\beta\psi_x)_x = \hat{\beta}_{+x}\alpha_{+x}\frac{(\psi_{+x}-\psi_{i,j})}{\gamma_x\Delta x^2} - \hat{\beta}_{-x}\alpha_{-x}\frac{(\psi_{i,j}-\psi_{-x})}{\gamma_x\Delta x^2} + \frac{\hat{\beta}_{+x}\alpha_{+x}}{\gamma_x\Delta x^2} + \frac{\hat{\beta}_{-x}\alpha_{-x}}{\gamma_x\Delta x^2} + \frac{\hat{\beta}_{+x}(1-\chi_{+x})b_{+x}}{\beta_{i+1}\gamma_x\Delta x} + \frac{\hat{\beta}_{-x}(1-\chi_{-x})b_{-x}}{\beta_{i-1}\gamma_x\Delta x} \tag{2.54a}$$

where the coefficients $\hat{\beta}_{\pm x}, \alpha_{\pm x}$ and γ_x are obtained as follows.

$$(s_l)_{\pm x} = \left\{\frac{(\phi_l)_{i,j}(\phi_l)_{i\pm1,j}}{\left|(\phi_l)_{i,j}(\phi_l)_{i\pm1,j}\right|}\right\}$$
$$s_{\pm x} = \min_{l=1,L_{\max}}\{(s_l)_{\pm x}\} \tag{2.54b}$$

$$\chi_{\pm x} = \min_{l=1,L_{\max}}\left\{\left|\max((s_l)_{\pm x},0)\right| + \frac{\left|(\phi_l)_{i.j}\right|}{\left|(\phi_l)_{i.j}\right| + \left|(\phi_l)_{i\pm1.j}\right|}\left|\min((s_l)_{\pm x},0)\right|\right\} \tag{2.54c}$$

$$\delta_{\pm x} = \begin{cases} 1 & \text{if solid - fluid interface between (i, j) and } (i \pm 1, j) \\ 0 & \textit{otherwise} \end{cases} \tag{2.54d}$$

$$\psi_{\pm x} = \delta_{\pm x}\psi_{I_{\pm x}} + (1-\delta_{\pm x})\psi_{i\pm1,j} \tag{2.54e}$$

$$\alpha_{\pm x} = \delta_{\pm x}\frac{1}{\chi_{\pm x}} + (1-\delta_{\pm x}) \tag{2.54f}$$

$$\hat{\beta}_{\pm x} = \frac{\beta_{i,j}\beta_{i\pm1,j}}{\beta_{i,j}\chi_{\pm x} + \beta_{i\pm1,j}(1-\chi_{\pm x})} \tag{2.54g}$$

$$\begin{aligned}\gamma_x = {} & \delta_{+x}\frac{\chi_{+x}}{2} + \delta_{-x}\frac{\chi_{-x}}{2} + (1-\delta_{+x})\left\{\frac{1}{2} + \frac{1}{2}\chi_{-x}\delta_{-x}\left|\min(s_{+x},0)\right|\right\} \\ & + (1-\delta_{-x})\left\{\frac{1}{2} + \frac{1}{2}\chi_{+x}\delta_{+x}\left|\min(s_{-x},0)\right|\right\}\end{aligned} \tag{2.54h}$$

$$a_{\pm x} = \frac{(\phi_{l_1})_{i,j}}{\left|(\phi_{l_1})_{i,j}\right|} a_{I_{\pm x}} \left|\min(s_{\pm x},0)\right|(1-\delta_{\pm x}) \tag{2.54i}$$

$$b_{\pm x} = \pm\frac{(\phi_{l_2})_{i,j}}{\left|(\phi_{l_2})_{i,j}\right|} b_{I_{\pm x}} \left|\min(s_{\pm x},0)\right|(1-\delta_{\pm x}) \tag{2.54j}$$

The advantage of casting the equations in the above form is that implementation in a computer code is straightforward. Note that Equations (2.54) reduce in the appropriate cases to the discrete form for a solid-fluid or for a fluid-fluid interface and to standard central differences in the absence of interfaces. The above coefficient assembly also applies to any point in the domain, including points that lie away from the interface and interface adjacent points that conform to any of the cases shown in Figure 2.3.

2.3.6 Velocity Correction

Once the intermediate velocity and pressure fields have been obtained as described above, the velocity correction step is performed to update to a divergence-free velocity field. For grid points that lie away from the interface this is straightforward. For points that lie next to the immersed boundary the corrections are to be performed based on the different situations that may arise at such points, as illustrated in Figure 2.3. The pressure gradients required to correct the cell center and cell face velocities have to be

evaluated in a manner consistent with the evaluation performed for obtaining the gradients while discretizing the Laplace operator in the pressure-Poisson equation. For the particular cases illustrated in Figure 2.3 the corrections are effected as follows in a generalized fashion:

$$(s_l)_{\pm x} = \left\{ \frac{(\phi_l)_{i,j}(\phi_l)_{i+1,j}}{\left|(\phi_l)_{i,j}(\phi_l)_{i+1,j}\right|} \right\}$$
$$s_{\pm x} = \min_{l=1,L_{\max}} \{(s_l)_{\pm x}\} \tag{2.55a}$$

$$\chi_{\pm x} = \frac{(1+\left|\min(s_{\pm x},0)\right|)}{2} \min_{l-1,L_{\max}} \left\{ \begin{matrix} \left|\max((s_l)_{\pm x},0)\right| \\ + \dfrac{\left|(\phi_l)_{i,j}\right|}{\left|(\phi_l)_{i,j}\right| + \left|(\phi_l)_{i\pm 1,j}\right|} \left|\min((s_l)_{\pm x},0)\right| \end{matrix} \right\} \tag{2.55b}$$

$$u_{i,j}^{n+1} = u_{i,j}^{*} - \frac{\Delta t}{\rho_{i,j}} \frac{\alpha_x (p_{+x} - p_{0-x})}{\Delta x} \tag{2.55c}$$

$$p_{+x} = p_{i+1/2,j} \left|\max(s_{+x},0)\right| + p_{I_{+x}} \left|\min(s_{+x},0)\right| \tag{2.55d}$$

$$\alpha_x = \frac{1}{(\chi_{+x} + \chi_{-x})} \tag{2.55e}$$

Note that cell face velocities are corrected independently:

$$u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^{*} - \frac{\Delta t}{\rho_{i,j}} \left(\frac{\partial}{\partial x} p^{n+1}\right)_{i+1/2,j} \tag{2.55f}$$

The pressure gradient at the cell face is obtained based on straightforward central differences. The discrete correction expressions are similar to those given above.

The advantage of casting the equations in the above form is that implementation in a computer code is straightforward. Note that Equations (2.55) reduce, in the

appropriate cases, to the discrete form for a solid-fluid interface or for a fluid-fluid interface and to standard central differences in the absence of interfaces. Thus, a sharp-interface calculation that handles solid-fluid immersed boundaries, fluid-fluid immersed boundaries and their interactions can be easily programmed by a few lines of code that modify a simple uniform Cartesian grid flow solver. The discretization of the components of the Laplace operator involving derivatives in the y and z directions is performed with procedures identical to that presented above for the x derivatives. The above form unifies the treatment of the sharp interface method with the Ghost-Fluid method [74] and is a first-order implementation of the Immersed Interface Method [91].

2.4 Moving Boundaries

In Eulerian sharp interface methods, when the solid boundary moves across a grid point, the state of the point can change from liquid to solid or vice versa. Different approaches have been employed to handle this situation. In Ghost-Fluid type methods and immersed boundary methods [48, 78, 93] or fictitious domain methods [57] flow fields are computed within as well as outside the immersed solid object. Thus, when the boundary crosses over a grid point, changing the state from solid to fluid, the newly emerged fluid point simply takes on the flow field variables that were available at that point in the previous time step. In the sharp interface method [143] as well as immersed interface method [89, 90] where the flow is computed separately in each sub-domain (fluid and solid) separated by the interface and no ghost flow field exists in the solid, a scheme must be devised to obtain the flow field variables at the newly emerged fluid point. Note that the converse case where the emergence of a grid point that was in the fluid phase into the solid phase presents no issues since the flow field is not computed in

the solid phase.

A newly emerged fluid grid point is defined by the condition $(\phi_l)_{i,j}^{n+1}(\phi_l)_{i,j}^{n} < 0$. Since the point was previously in the solid phase $((\phi_l)_{i,j}^{n} < 0)$ it had no history in the fluid phase ($(\phi_l)_{i,j}^{n+1} > 0$), i.e. $\vec{u}^n$ (as also ξ^n) does not exist in the fluid phase for such a point. Therefore, these points are to be evolved to time level $n+1$ in a special fashion. Note that since the pressure Poisson equation does not have a time-dependent term the pressure in such a cell can be evaluated as usual once a $\vec{u}*$ value is available after solving the momentum equation. The method to obtain $\vec{u}^n$ (and ξ^n) for such points follows along the lines detailed in [143, 144] and is analogous to the approach taken in moving grid formulations when a fresh grid point is inserted following mesh refinement. The value at such points is obtained by interpolation from the known values in the surrounding cells and on the moving boundary (where the boundary conditions are specified). For the particular time step when a grid point changes from solid to fluid phase, the value of $\vec{u}^n$ at that point is found using a linear interpolation operator spanning points in the fluid and on the interface.

The interpolation points that are picked depend on the orientation of the interface in the cell as illustrated in Figure 2.5. For the particular case in Figure 2.5, the value at the freshly cleared cell (i, j) is calculated as $\psi_{i,j}^{n} = (\chi_{-y}\psi_{i,j+1}^{n} + \psi_{I_{-y}}^{n})/(1+\chi_{-y})$, where χ_{-y} is the distance between the grid point (i, j) and the interfacial point I_y. The interpolation points are chosen depending on the direction of the normal vector at L_y ($\vec{n} = (n_x, n_y)$) and the ratio n_y / n_x. For instance, in the above expression for the case in

Figure 2.5, points L_y and $(i, j+1)$ are chosen since (n_y>0 and $n_y > n_x$).Consistent with the difference scheme in the interface-adjacent grid points the treatment at the newly emerged fluid points is first-order accurate. Note that this procedure is equivalent, in analogy with purely Lagrangian (moving grid) methods, to interpolating the value of variables to a newly inserted point after mesh refinement from values at the old mesh points (i.e. before refinement). In diffuse interface Eulerian methods (where interfaces may be captured using VOF, Levelset, phase field etc.), where the interfacial forces are spread over the mesh [6, 23] this issue of cross-over does not arise since there is no clear-cut interface location and all properties are taken to vary smoothly over a few mesh points.

2.5 Algebraic Multigrid with Moving Embedded Boundaries

The typical problems to be solved using the above developed method involve embedded solid-fluid and fluid-fluid moving interfaces. A well designed geometric multigrid method has been previously shown to be effective in accelerating the Poisson solver for solid-fluid interface problems [144, 160]. Geometric multigrid methods require information on the manner in which the embedded geometry cuts through the mesh to design an effective grid coarsening strategy and special modifications at each grid level have to be made to account for embedded boundaries. As the complexity of the problems increases upon the inclusion of fluid-fluid and solid-fluid-fluid interfaces with property jumps and surface tension, the complexity of designing a geometric multigrid method also increases. Algebraic Multigrid (AMG), on the other hand, is virtually a 'black-box' solver. In case of fluid-fluid problems with implementation of Ghost Fluid Method, the

formulation is likely to have a coefficient matrix that is asymmetric with large coefficient jumps depending on the density difference between the two fluids. Assembly of coefficients on coarser meshes for the geometric multigrid method then becomes a complex process.

Moreover, in the case of moving boundary problems, coefficient assembly may need to be done at every time step. In moving interface problems a local mesh refinement (LMR) algorithm with a tree structure is attractive in order to better resolve the interfaces [110, 162]. Such a local mesh refinement would result in the coefficient matrix losing its penta-diagonal structure and becoming sparse and unstructured. In such cases the AMG approach would be a natural choice to accelerate the Poisson solver, since it relies only on the coefficient information on the finest level of mesh. The geometry information carried by the fine grid coefficients is automatically transmitted to the coarser meshes. With some modification to the standard algebraic multigrid algorithm, an effective grid coarsening and solution strategy has been designed that naturally accounts for the presence of embedded objects, whether they are stationary or moving solid-fluid or fluid-fluid interfaces.

2.5.1 Local Coarsening for Moving Boundary Problems

In AMG, the coarser grids are recursively set up (starting from the finest mesh) based on the coefficient matrix from the previous level. An appropriate interpolation scheme is defined and coarse grid coefficients are obtained from the next fine level coefficients using the Galerkin identity [63]. The process of coarsening is repeated till a sufficiently coarse level system is obtained. Different methods and algorithms are available for the definition of the interpolation operator. The present implementation uses

the algorithm described by Wagner [148] with a point-by-point Gauss-Seidel as the smoother. A parallel implementation of the algorithm is also possible[24, 35, 65, 69]

The main components of the multigrid algorithm include the grids from level 1 (finest) to M (coarsest) denoted by Ω^m where m denotes the level of mesh. Coefficient matrices on each level of mesh m are represented by A^m. The interpolation operator (coarse to fine) is represented by I_{m+1}^m and the restriction operator (fine to coarse) by I_m^{m+1}. The restriction operator matrix is the transpose of the interpolation operator and the coefficient matrix on the next coarser level of mesh is computed by a Galerkin type operation as $A^{m+1} = I_m^{m+1} A^m I_{m+1}^m$. The set-up phase includes dividing the set Ω^m into the coarse and fine sets, defining the interpolation matrix I_{m+1}^m and computing the new operator matrix A^{m+1}.

A significant trade-off for the robustness of AMG is the long setup time required to assemble the coarse meshes and coefficients. In the case of stationary boundaries, the set-up needs to be done only once and can be ignored as a one-time investment of CPU time. But for moving boundary problems, the coefficients on the fine mesh, in the vicinity of the interface are re-assembled at every time step, which means that the coarse meshes also need to be re-assembled at every time step. This is not a very efficient process. To solve this problem, a 'local coarsening' strategy is developed which effectively deals with this obvious disadvantage of AMG as a fast solver. In the Cartesian grid framework, when the interface moves, only the coefficients at grid points close to the interface need to be re-assembled. Therefore it seems unnecessary to re-coarsen and re-assemble the grids and coefficients in the entire domain at every time step.

A strategy has been developed to re-coarsen only the cells within the narrow band of the Levelset field corresponding to the moving interface. This is done at every level of multigrid. At coarser levels of mesh, the Levelset information is coarsened and only those mesh points that are likely to have dependencies on the interfacial cells are cleared for re-coarsening. Since the interface generally moves less than a grid spacing during a single time step (due to a CFL-type criterion), this process ensures that all the grid points that are likely to be dependent on the interface points are re-coarsened, while the rest of the domain uses the same coarse grid as in the previous time step. Figure 2.6 illustrates this 'local' coarsening process. The domain consists of two objects placed in the flow path shown in the figure by thick lines. The coarsened Levelset tube is marked by the thin lines. The first object is a solid cylinder of non-dimensional radius 0.1, placed at (0.3, 0.7), the second is a bubble of the same radius placed at (0.7, 0.3). The latter fluid-fluid interface has large coefficient jumps because the density of the liquid is 100 times that of the bubble. Figure 2.6(a) shows the coarsened mesh at the beginning of a time step. At this point a full coarsening of the domain has been done for the multigrid. Figure 2.6(b) shows the starting of coarsening process at the next time step. As shown in the figure, only the mesh points within the Levelset tube are cleared for coarsening while the remaining domain is maintained as in the previous time step. AMG set up strategy is then applied to the cleared cells. Figure 2.6(c) shows the domain that has been re-coarsened locally at the second time step. It can be noticed from Figure 2.6(c) that the local coarsening strategy does not necessarily choose the same points that were chosen when the whole domain was coarsened. This implies that with local coarsening, there is a likelihood that the moving interface may leave a trail of irregular coarsening following its

path of motion as the solution progresses. From rigorous testing of moving interface problems involving solid-fluid and fluid-fluid interfaces, it was found that although the points chosen by local coarsening are not necessarily exactly the same as the ones chosen by full coarsening, the convergence trends of AMG are not significantly affected. This is demonstrated later in sections 2.5.3 and 2.5.4. Therefore local coarsening is a strategy that can be used for set up of multigrid for fast solution of the pressure Poisson equation in moving interface problems. If desired, full re-coarsening can be performed periodically at a fixed interval so that all traces of the moving boundary are eliminated periodically in the course of calculation. It has been observed that depending on the size of the interface, the local coarsening strategy takes up only about 10% of the time taken up by full coarsening for most applications.

2.5.2 Evaluating the Performance of AMG

The performance of the algebraic multigrid solver is evaluated for typical problems with embedded sharp interfaces, which involve solid-fluid or fluid-fluid interfaces or both. The treatment of the solid-fluid interfaces has been detailed in this chapter and the fluid-fluid interface is handled using the Ghost Fluid Method [49]. An important consideration while selecting a solver is the scalability of the method. The scalability of a method is defined as the ratio of the solve time to the number of computational points on mesh. With increasing number of computational points, the performance of a method with better scalability gets increasingly better when compared to other solvers [154, 155]. The number of work units in the solution process is defined by:

$$\varpi = \sum_{m=1}^{M} n_m \frac{N_m}{N_1} \tag{2.56}$$

ϖ indicates the number of work units and n_m denotes the number of inner iterations on level m of mesh, N_m is the number of points on level m of the mesh and N_1 is the number of points on the finest level of mesh. This is a measure of the computational effort expended in the solution of a given problem. The CPU time required for the solution to converge to a residual level of 1e-6 in the L_∞ norm is measured for the advancement of the solution for 25 time steps starting from an arbitrary initial condition for the pressure and velocity fields. For the stationary boundary problems considered below, at the end of 25 time steps, the effects of the arbitrary initial conditions have generally been overcome and the subsequent convergence is fairly rapid. Because of the presence of embedded boundaries and the non-uniformity of coefficients of the cells adjacent to the interface, the highest values of residuals are generally concentrated close to the interfacial cells. The data presented has been recorded for a maximum of 3 levels of multigrid so as to ensure uniformity. More levels can be used as well depending on the number of mesh points on the finest mesh and the aspect ratio of the flow domain. The performance of the AMG solver is tested below for a hierarchy of problems involving embedded interfaces. The performance is compared with the standard point Gauss-Seidel and Line-SOR solvers.

2.5.3 Scalability Tests

The scalability characteristics of the three solvers tested are shown in Table 2.1. for different applications with and without embedded interfaces. In each case the

scalability characteristic of the AMG solver is found to be at least three times better than that of the other solvers depending on the specific application. The relative scalability of the solvers is more or less independent of the type of application with or without interfaces as seen from Table 2.1.. With increasing grid size the performance of AMG gets increasingly better when compared to the Line solver and Gauss-Seidel point solver. For instance, the data for the flow across cylinder in 2D in Table 2.1. shows a scalability of 0.052 for the Line solver, 0.054 for the point solver and 0.01 for the Multigrid. Hypothetically, if a grid with 100 computational points requires a solve time of 'n' secs, a grid with 1,000000 grid computational points would require 'n+51994' secs for the point solver, 'n+53994' secs. for the Line solver and only 'n+9999' secs., for the Algebraic multigrid.

2.5.4 Solve Time Comparison

Table 2.2 shows the comparison of solver parameters (the solve time, work units and number of fine mesh iterations) on a variety of moving boundary problems with immersed interfaces. The problem specifications are listed in the table. The recorded data is scaled by the corresponding values for a Gauss-Seidel Point solver. The AMG solver is found to consistently yield a speed-up of 5 times over the point solver in each case, regardless of the type of the immersed interface. The line solver on the other hand performs better in the case of solid-fluid interfaces than fluid-fluid interfaces.

2.5.5 Moving Boundaries with Local Coarsening

This case involves moving solid-fluid as well as fluid-fluid interfaces and tests the performance of the AMG solver with multiple embedded boundaries with and without

local coarsening. A bubble is placed in a tube with moving walls. The tube contracts and expands periodically. The bubble is deformed and transported due to the contraction of the tube. The evolution of the shape of the tube walls and the bubble in the contraction phase are shown in Figure 2.7(a). As seen in Figure 2.7 AMG achieves a speed up of about 10 times that of the point solver for this problem, even in the presence of multiple moving interfaces. Figure 2.7(b) illustrates the recorded solver data for this application. The figure shows the comparison of solve time, work units, fine mesh iterations tabulated in Table 2.2 as well as the relative set-up times for the multigrid with and without local coarsening. While there is a 10 times speed-up in the set-up process with local coarsening the solve times in both cases are comparable. This shows that in the case of moving boundary problems, where the interface coefficients change at every time step, local coarsening is a viable alternative to save time required for multigrid set-up. A comparison between the multigrid with and without local coarsening shows a marginal increase in the solve time when the local coarsening strategy is used. But this is offset by the increased set-up time required by the multigrid without local coarsening. Over increasing number of time steps the increasing savings in the set-up time leads to a large saving in total computation time when local coarsening strategy is used.

2.6 Particle Tracking

In solving the valve closure problem, the principal interest is in predicting sites prone to thrombus formation due to platelet activation. Platelets are modeled as point particles by a Lagrangian particles tracking algorithm [27, 40, 98]. The method adopts a one way coupling mechanism in which the particles are influenced by the flow field but the flow field is not influenced by the particles. A dilute flow is assumed and particle-

particle interactions are neglected. Particle length scales are assumed to be very small compared to the fluid length scales and particles are treated as points. No wake effects are included. All particles are assumed to be spheres of the same diameter with particle material density being larger than that of fluid. The carrier fluid is assumed to be incompressible. Drag and lift forces acting on the particles due to the fluid flow depend on the particle size and mass and influence the particle motion. A particle momentum equation is solved to calculate particle velocity as:

$$\frac{d\vec{v}}{dt} = \frac{\partial v}{\partial t} + \vec{v} \cdot \vec{\nabla} v = \vec{f} \tag{2.57}$$

In the above equation, $\vec{v}$ is the particle velocity vector, $\vec{f}$ represents the total contact force on the particle including the drag and lift forces. The lift and drag forces are modeled as functions of the particle Reynolds number and Stokes number. The modeled drag force is given by [36]:

$$\vec{f}_D = -\frac{1}{St}\left(1 + 0.15\,\mathrm{Re}_p^{\,0.687}\right)\left(\vec{v} - \vec{u}\right) \tag{2.58}$$

In the above equation, $St = \rho_p d_p^{\,2} U / 18\mu L$, is the Stokes number defined as the ratio of particle response time to the fluid time scale. In the present case, the Stokes number correlates with the particle size or mass. $\mathrm{Re}_p = \rho_f \mid \vec{v} - \vec{u} \mid d_p / \mu$, is the particle Reynolds number and d_p, ρ_p are the particle diameter and particle density respectively. L, U are the particle length and velocity scales. Lift forces are taken into account when there is particle rotation and the pressure differences between the top and bottom sides cause lift on the particle. The 'Saffman' lift force [40] and the overall lift force for the

specific range of Reynolds numbers are given by [114, 115]:

$$f_{saffman} = 1.61 d_p{}^2 (\mu \rho_f)^{\frac{1}{2}} \left|\vec{\omega}_f\right|^{-\frac{1}{2}} \left[(\vec{u} - \vec{v}) \times \vec{\omega}_f\right] \tag{2.59}$$

$$f_L = f_{saffman} \left(\left(1 - 0.3314 \beta^{\frac{1}{2}}\right) e^{\left(\frac{-\mathrm{Re}_p}{10}\right)} + 0.3314 \beta^{\frac{1}{2}} \right) \forall \mathrm{Re}_p \le 40$$
$$f_L = f_{saffman} \left(0.0524 \left(\beta \mathrm{Re}_p\right)^{\frac{1}{2}} \right) \forall \mathrm{Re}_p > 40 \tag{2.60}$$

In the above equations, $\beta = \frac{d_p}{2|\vec{u} - \vec{v}|} |\omega_f|$ and $\omega_f = \nabla \times \vec{u}$ is the vorticity. After solving the particle momentum equation to obtain the particle velocities, the particle locations are updated as:

$$\vec{x}^{n+1} = \vec{x}^n + \vec{v} \delta t \tag{2.61}$$

$\vec{x}^{n+1}, x^n$ are the new and old particle locations and δt is the time step size. In the post processing phase, the concentration of particles is computed. The particle concentration equation is solved:

$$\frac{\partial c}{\partial t} + \vec{\nabla}.(c\vec{V}) = 0 \tag{2.62}$$

$\vec{V} = \langle a\vec{v} \rangle / \langle a \rangle$ is the averaged particle velocity. Equation (2.62) is solved either in explicit or implicit fashion. In certain cases when the particles are scattered in a

arbitrarily there may be situations when the velocity and concentration gradients computed are not stable because of lack of sufficient number of interpolatory points. In most cases, such particles may be removed by applying specific criteria. However, in problem where particles are scattered in response to fluid forces, particle removal can cause the results to present an unrealistic picture.

Therefore, a different method is adopted to compute the particle concentration. It is an SPH-like method [39, 101, 102]. The particles are treated as a cloud of particles and a Gaussian distribution is assumed. An average particle concentration is computed at any location as:

$$\langle c \rangle = \sum_{j=1}^{N_p} A_j e^{\left(\frac{-(r-r_j)^2}{h_j^2} \right)} \tag{2.63}$$

In the above equation N_p is the total number of particle in the domain, A_j is the amplitude of the cloud around particle j. $r - r_j$ represents the distance between particle j and the point at which concentration is being computed. h_j is the region of influence of particle j or the particle cloud radius. Since the distribution of particle is assumed to be Gaussian, the summation is carried out only on particles in a small region around the required location. This method is always stable since there is no gradient computation. However, this method does not give a very smooth particle concentration field. For the current application, this method of particle concentration computation is adopted.

2.7 Coupling Particle Transport with Flow

In this chapter, three parts of the overall computational algorithm including the

flow solver, the moving boundaries and particle tracking were developed. These components are essential in simulating the heart valve closure and predicting sites with high likelihood of platelet activation. The overall algorithm for the particle-interface-flow coupling is as follows:

1. Time is advanced $t^{n+1} = t^n + \delta t$.
2. The new boundary location is defined by advecting Levelsets to obtain ϕ^{n+1}.
3. Intermediate velocity field $\vec{u}^*$ is computed.
4. The scalars c^{n+1} (temperature, species concentration etc.) are computed.
5. The pressure field is calculated by solving the pressure Poisson equation.
6. The velocity field is corrected.
7. Computed velocities are input to the particles
8. Particle momentum equation solved with drag, lift and buoyancy forces
9. Particles locations are updated.
10. Post processing: Particle concentrations are calculated

Solver/ Problem	**Scalability (sec./computational point)**		
	Gauss-Seidel Point Solver	Line Solver	Algebraic Multigrid
Channel Flow in 2D (Re=250): No interfaces	0.045	0.05	0.009
Flow across cylinder in 2D (Re=50): Solid-Fluid Interface	0.052	0.054	0.01
Flow across sphere in 3D (Re=100) : Solid-Fluid Interface	0.07	0.06	0.02
Bubble Rising in Gravity (Re=526, We=1.5, ρ_w / ρ_a =1000): Fluid-Fluid Interface	0.2	0.4	0.07.

Table 2.1. Comparison of the scalability of different solvers with increasing grid-size. As seen in the above table the scalability of the AMG solver is better than that of the other solvers for problems with and without embedded interfaces.

Note: The solve time is plotted against the number of computational points in the domain. This is a straight line, the slope of which indicates the scalability of the method [154, 155]. A method with better scalability has a lower slope. This indicates that with increasing number of computational points, the performance of a method with better scalability gets better when compared to other solvers.

Solver / Problem	**Line Solver**			**Algebraic Multigrid**		
	Solve Time	Work Units	Fine Mesh Iteration	Solve Time	Work Units	Fine Mesh Iteration
Flow across cylinder in 2D (Re=20)	0.2720	0.187	0.1871	0.2008	0.0744	0.0434
Flow across cylinder in 2D (Re=300)	0.3265	0.186	0.1868	0.1916	0.0717	0.0415
Bubble Rising in Gravity (Re=526, We=1.5, ρ_w / ρ_a=1000)	0.8709	0.656	0.6566	0.1195	0.0476	0.0290
Bubble evolution under the action of a peristaltic wave (Re=10, We=0.01, ρ_w / ρ_a=100)	0.8224	0.704	0.7048	0.3092	0.2067	0.1299
Flow Across Sphere in 3D (Re=100)	0.5170	0.176	0.1762	0.3269	0.1015	0.0667

Table 2.2. Quantitative comparison of solver data for different applications.

Note: The recorded data is scaled by corresponding values for a Gauss-Seidel Point Solver. For example, the reported value for solve time is the ratio of solve time for Line solver or the Algebraic multigrid solver to the solve time for Gauss-Seidel Point solver. The other values are reported in a similar way.

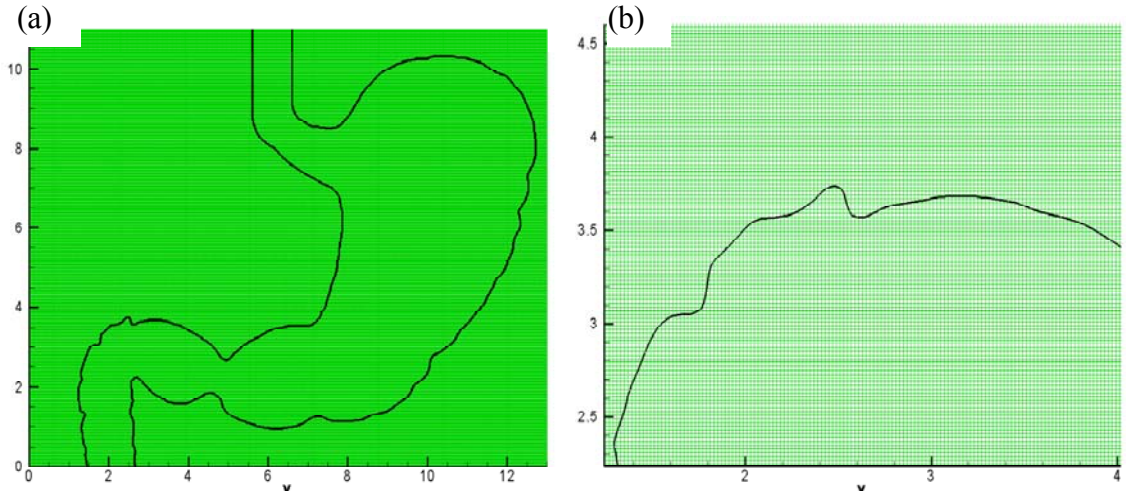


Figure 2.1. Demonstration of the Cartesian grid method. The computational domain is rectangular and interfaces are allowed to cut across the mesh. (a) Set-up of calculation flow calculation in the human stomach. (b) Zoomed-in view to demonstrate placement of interfaces.

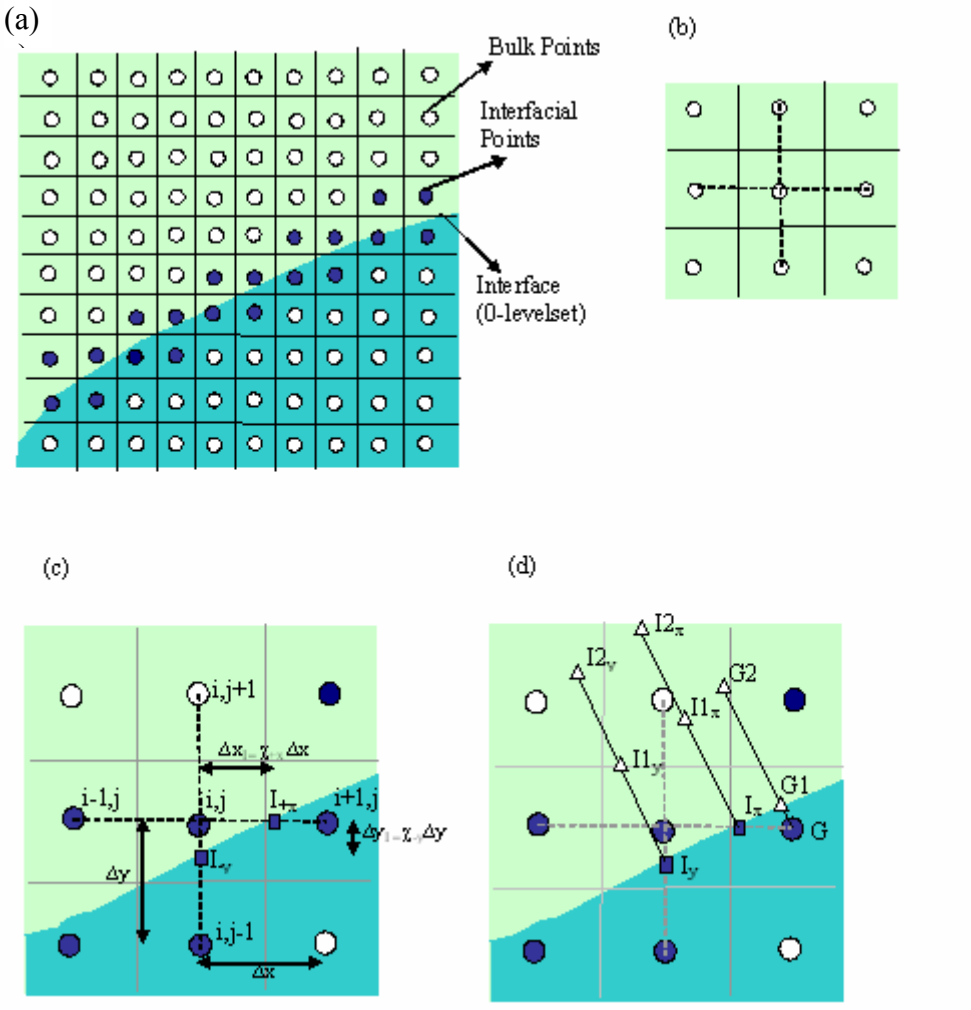


Figure 2.2. Description of the sharp-interface Cartesian grid method. (a) Definition of the bulk (clear circles) and interfacial (filled circles) points. The interface is given by the zero-Levelset. (b) Standard 5-point bulk point stencil in two-dimensions. (c) The configuration of a typical interfacial point. (d) System for evaluating the Neumann boundary condition on the interface and evaluation of ghost pressures.

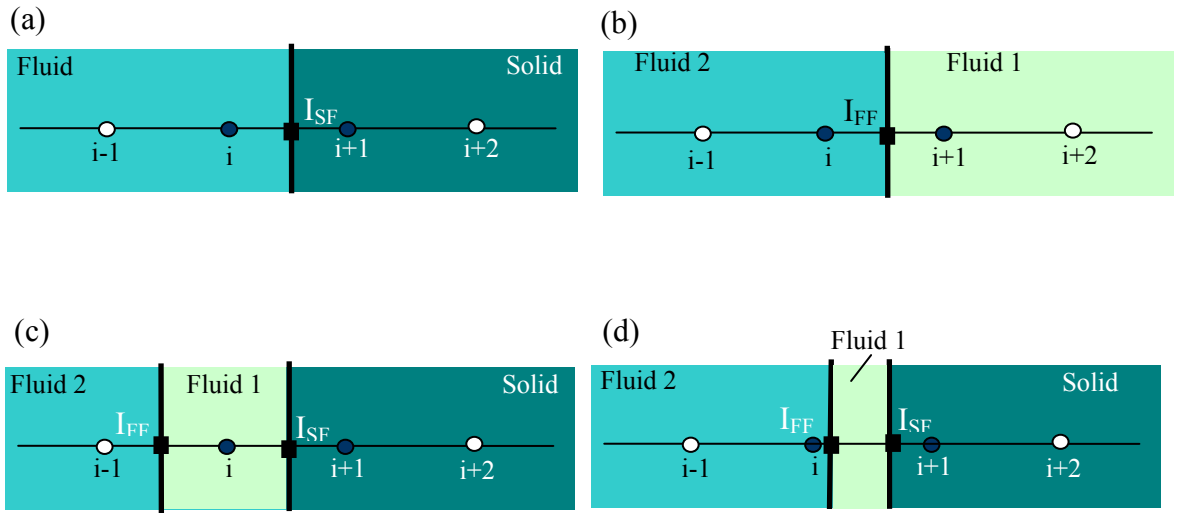


Figure 2.3. Some of the possible interfacial point situations in the two-dimensional case.

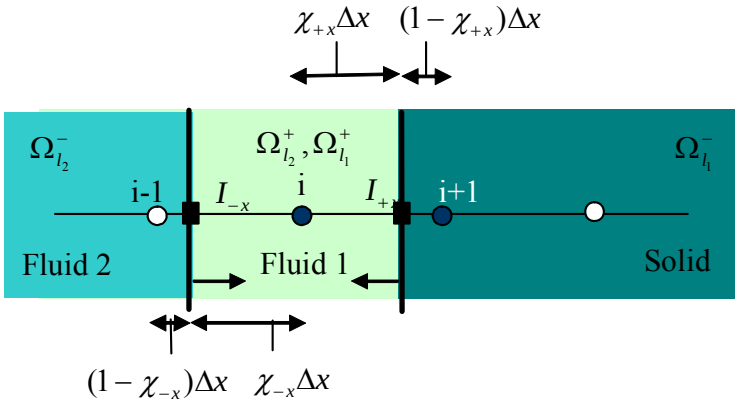


Figure 2.4. General case for formulating discretization at interfaces. This case involves solid-fluid and fluid-fluid interfaces. This is same as the case illustrated in Figure 2.2(c).

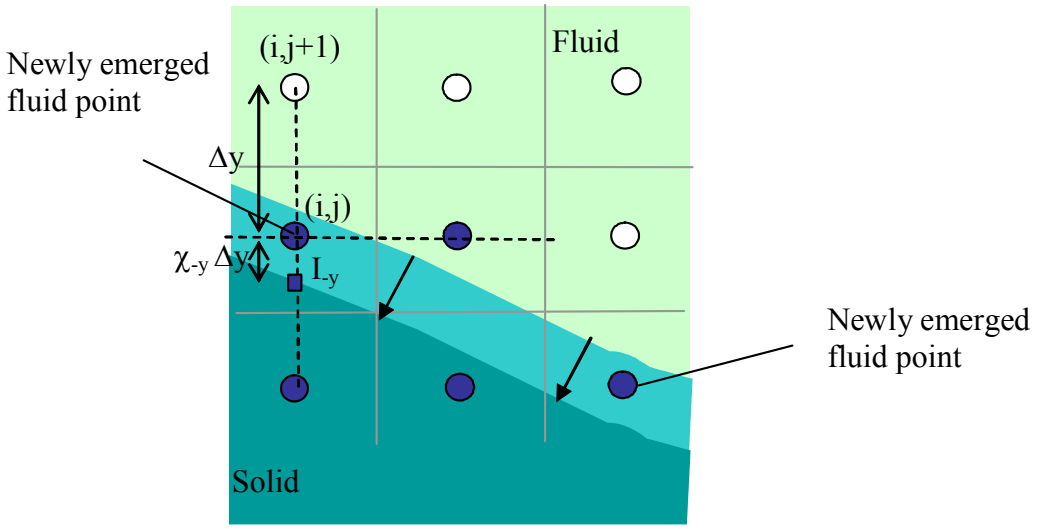


Figure 2.5. Illustration of the emergence of points from the solid to fluid phase when the sharp interface moves through the mesh.

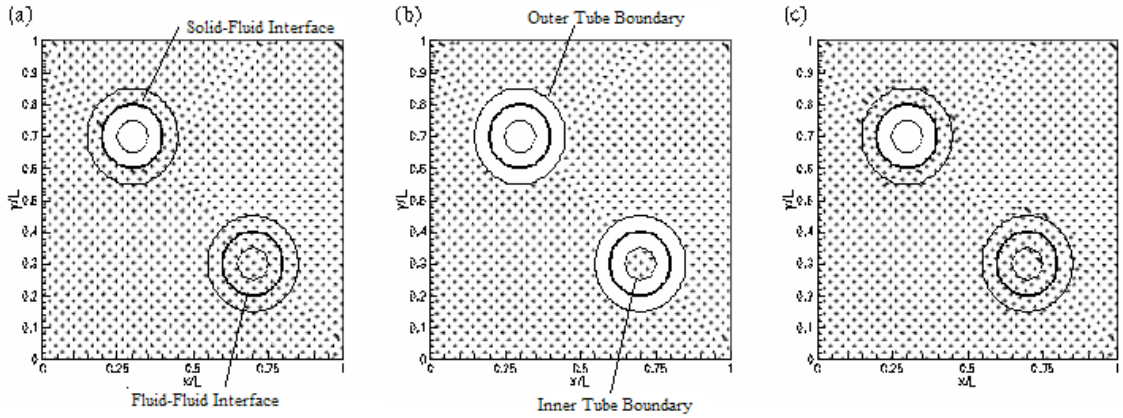
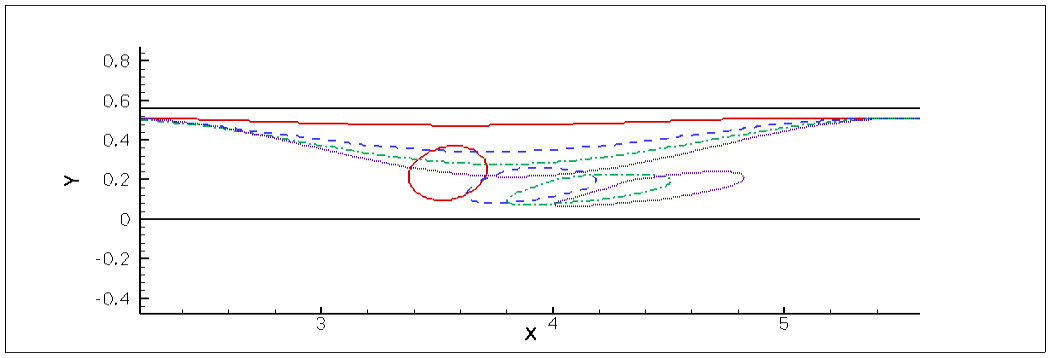


Figure 2.6. Local Coarsening process (a) Full coarsening at first time step (b) Outer tube cleared of coarse points when the interface moves (c) Outer tube re-coarsened for multigrid without changing coarse points in non-interface cells.

(a)



(b)

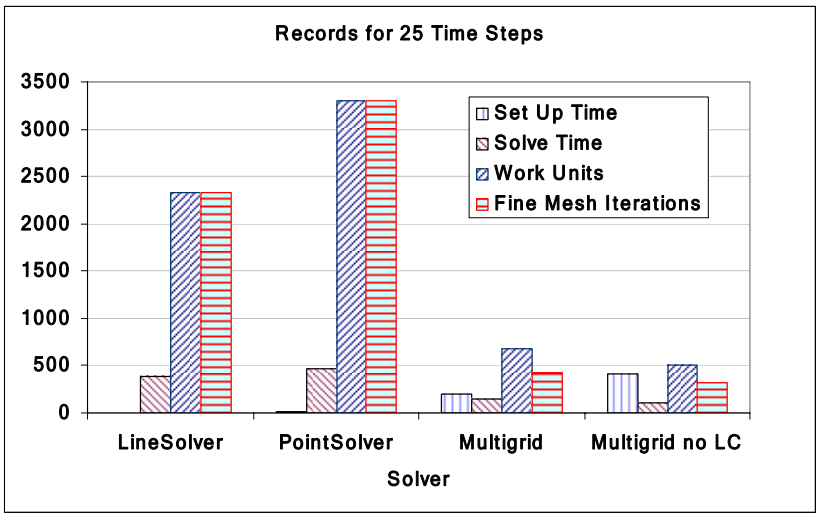


Figure 2.7. Solid-fluid-fluid moving interface problem. (a) Bubble evolution when subjected to peristaltic wave in a channel. Re=1, We=0.01, ρ_w / ρ_a=100. This problem involves tracking of fluid-fluid interfaces with the GFM approach for the immersed bubble and solid-fluid interface tracking for the moving wall. (b) Comparison of solver data for 25 time steps. (LC: Local Coarsening)

CHAPTER 3

LOCAL MESH REFINEMENT

3.1 Introduction

A unified formulation of the Sharp Interface Cartesian Grid method for moving boundary problems has been developed and validated in the last chapter [92, 96, 159]. The technique was shown to provide accurate solutions to problems involving solid-fluid and fluid-fluid interfaces and interactions between the two types of interfaces. However, while Cartesian grid methods allow for calculation of moving boundary problems with large deformations without the problems associated with mesh management, the one disadvantage of fixed grid methods is the heavy mesh requirement for well resolved calculations. Particularly, in calculations such as the mechanical heart valve simulations which involve a range of length scales, it is important to be able to adequately resolve all the critical length scales. To be able to simulate the problem at physiological Reynolds numbers with the realistic dimensions in a reasonable time frame, an adaptive meshing scheme is critical. Furthermore, since the mesh is not changed throughout the calculation it may be necessary to make intelligent estimates of regions where the interface traverses or where flow features such as boundary layers or high vorticity concentrations appear and to construct a mesh with required resolution at the beginning of calculation. Frequently, in the course of the simulations, it is seen that the region of the domain that requires fine resolution changes due to the movement of the interface and developing flow features. Resolving the whole domain to the finest resolution results in long computation times and overloading of computational resources. Under-resolving the

problem will result in loss of small length scale features. The solution is to adaptively refine the grid as the problem demands.

When dealing with problems of disparate length scales encountered in many applications, it is necessary to resolve the physically important length scales adequately to ensure accuracy of the solution. Various techniques have been proposed to resolve the issue of multiple length scales. The commonly used techniques are domain decomposition [137], adaptive mesh refinement (AMR) [12, 14] and local mesh refinement [58, 59] techniques. Domain decomposition methods decompose the domain into different sections with different meshes and mesh densities. These methods require an approximate idea of where the problem demands finer resolution and are hence not very versatile in the case of many moving boundary problems that may demand re-meshing at regular intervals. AMR and LMR schemes are ideally suited for moving boundary problems since mesh refinement is solely based on addition/ deletion of computational points from the existing grid. The mesh can thus be adapted as, when and where the solution demands.

3.2 LMR vs. AMR

Adaptive mesh refinement (AMR) [10-14] schemes were first developed for compressible flow problems. The main feature of this type of refinement scheme is that the flow equations are solved on every level of mesh for every time step with interpolations back and forth between the fine and coarse meshes. The problem is first solved on a base coarse mesh. Those cells that meet a pre-set refinement criterion are tagged for refinement. Optimal rectangular fine meshes are constructed encompassing the tagged cells. The flow equations are then solved on this level of fine meshes with the

interpolated coarse mesh solution as the initial and boundary conditions. Assuring solution continuity at the grid interfaces does not guarantee continuity of solution gradient and vice versa. Solution continuity and gradient continuity depend on an efficient synchronization algorithm [147] since both cannot be assured at the same time by applying boundary conditions. The fine mesh may be further refined if the refinement criterion is met and the process repeated. Implementation wise, the AMR scheme is less complicated than the other methods. No major changes are required in the data structure, discretization schemes or solvers since rectangular meshes are formed at every level. AMR schemes have been in use for many years because of the ease of implementation. Recent efforts have been made to parallelize AMR [86] as well as to adopt it for problems with deformable meshes [7].

The other mesh refinement scheme that has been widely applied to many problems is the local mesh refinement (LMR) scheme. The main difference between this scheme and the AMR scheme is that in the LMR scheme, instead of building optimized rectangular refined meshes, the existing cells are divided into four or eight cells. The computations are performed only on the undivided cells. In the LMR scheme, calculations are performed on the coarse mesh. Depending on the refinement criteria, the required cells are tagged for refinement. All the tagged cells (parent cells) are then split equally into four (2D) or eight (3D) child cells. The discretization scheme is modified so that the cells at the top level of refinement are uniformly integrated into the coefficient matrix. New cells derive their initial conditions from the solution on the previous level of mesh. Implementation wise, this scheme requires changes in the data structure to add and delete child cells as required. Connectivity information on the top layer of cells needs to

be established as well. The mesh becomes more or less unstructured with rectangular cells of different shapes. The coefficient matrix assembled in this manner is sparse and may require different solvers. However, because fewer number of computational cells are included in the solution matrix and the solution for every time step is achieved in a single solution step the LMR scheme is an attractive alternative to AMR schemes. Also, since the scheme does not require optimized rectangular meshes, required regions may be refined without adding more cells than required.

Local mesh refinement schemes have been applied to many different flow problems involving stationary and moving boundaries. A few examples are, flow around airfoils [41, 42], low to moderate Reynolds number flows with embedded solid boundaries [37, 38], motion of deformable cells [3], simulation of separated flows [60], Levelset advection problems [125-129] and simulation of water and smoke [94]. All the above works report comparable error estimates and speed up when compared to uniform fine mesh calculations. For standard test problems presented in the next chapter, the solution with grid adaptation has been found to be at least five times faster than that with uniform meshes with same resolution in the critical areas and with ten times less number of computational cells.

3.3 Data Structure and Implementation

In this chapter the sharp interface method discussed in Chapter 2 is extended by applying a local mesh refinement algorithm [58, 59, 162] that is designed to allow efficient and fast flow computations. Figure 3.1(c) summarizes the sharp interface Cartesian grid method with local mesh refinement. By means of several benchmark problems we demonstrate the accuracy of the LMR-supported sharp interface method for

a variety of problems. In particular, we demonstrate the capability of the technique to resolve boundary layers and other regions of steep gradients in addition to tracking moving interfaces. Detailed analysis is presented of the behavior of solution accuracy as well as efficiency with the changes in refinement criteria.

An important aspect of the local refinement scheme is the data structure that maps to the structure of the locally refined computational grid. Implementation of the LMR scheme requires compatible data structures for cell addition, removal, neighbor finding and easy mesh traversal. The quadtree structure in 2D and the octree structure in 3D are the most common methods of designing the data structure for local mesh refinement [58, 61, 110] as shown in Figure 3.2. The hierarchical data structure provides an easy way of coarsening and refining cells and finding neighbor connectivities [41, 42]. Greaves et al. [58-62] in a series of papers detail the implementation of the quadtree and octree structures for local mesh refinement. Various schemes have been tested for the best storage and accessibility of data. New indexing and storage schemes and neighbor finding algorithms have been shown.

A variation from the quadtree scheme is [44] to construct a ghost grid at the finest level of refinement with the variables being stored and calculations being performed only in the active cells. With this type of data structure it is possible to maintain the i, j, k format of cell numbering and is hence very compatible with existing FD and FV codes. The penalty of this scheme is the increased storage required at least for some variables. Linked lists are the other alternative to tree structures [20-22]. Operations such as cell removal and addition, neighbor finding and implementation of multigrid strategies are relatively difficult when using the linked list data structure.

In the present work, the quadtree in two-dimensions and the octree in three-dimensions has been adopted as the data structure most compatible with local refinement schemes. The description of data structure and discretization in this chapter is based on the quadtree structure because it is easier to explain. The explanations are easily extended to three dimensions. To begin, the domain is divided into a base coarse mesh. Unlike schemes which start out with a single cell with the refinement proceeding from there, this allows the domain to be divided into an unequal number of cells with no restrictions on the cell numbers in each direction. The base mesh is fixed so that the presence of embedded objects is identified. Also, since the refinement is solution adaptive, the base mesh needs to be fine enough to capture the major flow features.

Once the base mesh is fixed, the Levelset is generated on the coarse mesh. Based on the coarse Levelset field, the mesh is refined uniformly to as many levels as required to a distance of defined by the narrow-band Levelset field (6 cell widths in the present case) from the interface location. This is the region within which all the Levelset operations are performed and is maintained at a uniform refinement level. The Levelset is then re-generated on this fine mesh, such that the shape of the body is resolved adequately. To optimize the refinement process and to ensure that there is no spurious refinement in regions as the solution evolves from the initial conditions, the solution is allowed to develop adequately before the solution based refinement process is started.

Each cell in the tree structure shown in Figure 3.2 carries with it indices that point to its base cell number and its hierarchy in the tree structure. The first index indicates the base cell number and the second index indicates the cell position number. A second index carries information about the cell level and its position in the grid as:

$$I_{cell} = P_{cell}(10)^{l_{parent}} + I_{parent} \tag{3.1}$$

where $I_{cell}, I_{parent}, P_{cell}, l_{parent}$ are the cell index, parent index, position number of the cell (north-east (NE-4), south-west (SW-1), south-east (SE-2), north-west (NW-3) in 2D and bottom-north-east (BNE-4), bottom-south-west (BSW-1), bottom-south-east (BSE-2), bottom-north-west (BNW-3), top-north-east (TNE-8), top-south-west (TSW-5), top-south-east (TSE-6), top-north-west (TNW-7) in 3D) and parent cell level respectively. Using these indices, the south-west (SW) child of the north-east (NE) child of base cell number 56 is identified by the indices (56, 041). Simple mathematical operations on the indices identify the exact location of the cell, its level and position in the tree structure. Each cell also carries pointers to its parent cell and child cells. Neighbor information is obtained from the tree structure as described in detail in the next section. When a cell is refined, memory is allocated to its child cells and when coarsened, memory to the child cells is de-allocated. Neighbor connectivities can be found by logical operations while traversing the data tree. Pointers to the neighbor cells are stored in the data structure to avoid time consuming logical operations each time connectivity information is required. The flow equations are solved on the top-layer of cells which are typically called 'leaf cells' in LMR terminology. Flow variables are stored on each leaf cell. Except for the leaf cells, all the other cells essentially support the data structure and are allocated minimum memory.

The Fortran code for the cell type is shown in Figure 3.3. The variable names are self-explanatory. Each cell in the data structure can be accessed in the code by recursive functions traversing the data tree. To avoid complexity of implementation of discretization schemes, the refined grid is smoothed to assure that no two neighboring

cells have more than a level difference between them.

3.3.1 Finding Neighbors

The locally refined mesh is highly unstructured. Pointers to neighbor cells are stored within the cell structure shown in Figure 3.3 to avoid the time intensive tree searches. The connectivity information stored in each cell is used in aiding neighbor searches. As a demonstration, consider the cell configuration shown in Figure 3.4(a). The numbers indicate the indices of the cells in the quadtree. The first index indicates the base mesh cell number and the second index reveals the location and level of the cell using the indexing system described in the previous section. For instance, the north neighbor of the cell indicated by the index (3,40) is found by accessing its parent cell (3,0). The north neighbor of the parent cell (3,0) in this case is (1,0). This cell is examined to decide if its is divided or not. In this particular case, it is an undivided cell. If this cell is undivided, then the pointer to the north neighbor of cell (3,40) is set to the cell (1,0).

Similarly, in searching for the east neighbor of the cell (3,40), the parent cell (3,0) is accessed. The east neighbor of the parent cell in this case as shown in Figure 3.4 is the cell (4,0). This cell is divided. Therefore its child cells are accessed to find the neighbor of the cell (3,40). The index '40' indicates that it is the north –east child cell of its parent. The east neighbor of the north-east child cell is necessarily the north-west child cell of the parent's neighbor. Hence the pointer to the east neighbor of the cell (3,40) is set to point to the cell (4,30). In case of the west neighbor of the cell (3,40), the search is relatively straight-forward. The west neighbor cell of a north-east child cell is necessarily the north-west child of its parent cell. Hence the cell (3,30) is the west neighbor of the cell (3,40). The logic works similarly in case of an octree in three dimensions. The rules

that govern neighbor searches in case of a north east (NE) child cell are shown in Figure 3.4(b). The logic works similarly in case of the other child cells as well. It may be noted that a cell is allowed to have only one neighbor in any direction which implies that a cell may have a neighbor which is at a level equal or lower than itself. A leaf cell can thus have divided neighbors.

3.4 Discretization at Mesh Interfaces

Various approaches are found in literature for the discretization across mesh interfaces [94]; [58, 59]; [20, 21]; [3, 44, 150, 152, 153]; [110] with and without embedded objects. Most of the previous works use finite volume formulations with the exception of Durbin et al. [44] who use a finite difference formulation. The main challenge in formulating a viable discretization scheme is the issue of mass conservation across mesh level interfaces. With finite difference schemes this is relatively more difficult to achieve than with finite volume schemes. Additional complexity is introduced by the embedded boundaries because the discretization as well as the cell refinement has to take into account the way in which the cells are being cut by the interfaces [37, 38]. In 2D this is relatively straightforward but in 3D this may be a formidable task because of the orientations of various cut cells. In the present work, simplicity of discretization at interfaces is preserved by a finite difference formulation at the interfaces. For additional simplicity, the mesh is refined uniformly to the same level up to a few grid cells from the interface. Though this may result in a few cells being refined unnecessarily the formulation remains relatively simple. Removal of this restriction on the refinement, while maintaining the simplicity of the sharp interface method is the subject of ongoing work.

In this work a mixed scheme is used for the discretization of the governing equations. For cells without refinement boundaries or interfaces cutting across them, the finite difference and finite volume formulations are identical. For the cells with an embedded boundary cutting across them the finite difference scheme as discussed in Chapter 2 is retained as it is far simpler than a finite volume scheme, especially in 3D. The method has been rigorously tested and validated for different applications [92, 96, 141-144, 159] With the refinement scheme proposed by Sochnikov et al. [121] where all the cells up to a certain distance from the interface are always refined to the finest level, the formulation for interface cells does not need to be modified. For mesh interfaces, a finite volume scheme is adopted to ensure mass conservation. Therefore, for the mesh interfaces a 'ghost cell' approach is implemented as explained below.

While refining cells, it is ensured that no two neighboring cells have more than a level difference between them. This ensures that only two specific configurations need to be addressed when discretizing the governing equations across mesh interfaces. The two configurations are shown in Figure 3.5 and the discretization schemes for each case are discussed below. For simplicity, the discretization of the equations is explained in 2D. Since the discretization is chiefly dependent on interpolations and the fluxed are assembled independently in each coordinate direction, extension to 3D is straightforward.

3.4.1 Case (a) Neighbor Cell is Divided

Figure 3.5 (a) shows the configuration in question. As seen in the figure, cell P is a leaf cell and its west neighbor W is divided. The discretization across the mesh interface is demonstrated for the Laplace operator, $\nabla^2\psi$. Equation (3.2) shows the

discretization of the diffusion operator for the variable ψ.

$$\nabla^2\psi = \frac{(f_e - f_w)}{\Delta x} + \frac{(f_n - f_s)}{\Delta y} \qquad (3.2)$$
$$where \;\; f_w = f_{w1} + f_{w2}$$

f represents the flux across the cell face. The flux from each of the west faces (f_{w1}, f_{w2}) into the current cell. For instance, the fluxes into each cell face are calculated as follows in Equation (3.3). Note that P_1^* and P_2^* are ghost cells. The variables $\psi_{P_1^*}$ and $\psi_{P_2^*}$ are interpolated at these locations from the surrounding cells by bi-linear weighted averaging as shown in the configuration shown in Figure 3.5(a) case by Equation (3.4).

$$f_e = \frac{\psi_E - \psi_P}{\Delta x}(\Delta y)$$
$$f_{w1} = \frac{\psi_{P_1^*} - \psi_{w1}}{0.5\Delta x}(0.5\Delta y),\; f_{w2} = \frac{\psi_{P_2^*} - \psi_{w2}}{0.5\Delta x}(0.5\Delta y) \qquad (3.3)$$
$$f_n = \frac{\psi_N - \psi_P}{\Delta y}(\Delta x)$$
$$f_s = \frac{\psi_P - \psi_S}{\Delta x}(\Delta y)$$

$$\psi_{P_1^*} = 0.5625\psi_P + 0.1875\psi_W + 0.1875\psi_N + 0.0625\psi_{NW} \qquad (3.4)$$

Since cell W is divided, the variable value ψ_W is interpolated bi-linearly from the surrounding leaf cells as:

$$\psi_W = 0.25\left(\psi_{w1} + \psi_{w2} + \psi_{w3} + \psi_{w4}\right) \qquad (3.5)$$

Putting everything together,

$$\psi_{P_1^*} = 0.5625\psi_P + 0.046875\left(\psi_{w_1} + \psi_{w_2} + \psi_{w_3} + \psi_{w_4}\right) + 0.1875\psi_N + 0.0625\psi_{NW} \quad (3.6)$$

And

$$\psi_{P_2^*} = 0.5625\psi_P + 0.046875\left(\psi_{w_1} + \psi_{w_2} + \psi_{w_3} + \psi_{w_4}\right) + 0.1875\psi_S + 0.015625\left(\psi_{SW_1} + \psi_{SW_2} + \psi_{SW_3} + \psi_{SW_4}\right) \quad (3.7)$$

The above 'ghost' quantities are substituted into Equation (3.3) to obtain the fluxes which are in turn substituted in Equation (3.2) to obtain the discretization of the Laplace operator. Finally, the stencil for $\nabla^2\psi_P$ in Figure 3.5(a) includes all the surrounding neighboring cells as shown in Equation (3.7).

$$\nabla^2\psi_P = f(\psi_P, \psi_E, \psi_{W_1}, \psi_{W_2}, \psi_{W_3}, \psi_{W_4}, \psi_N, \psi_S, \psi_{NW}, \psi_{SW_1}, \psi_{SW_2}, \psi_{SW_3}, \psi_{SW_4}) \quad (3.8)$$

The advection operators are discretized in a similar fashion but treated explicitly in time.

3.4.2 Case (b) Neighbor Cell Level Lower than Current Cell

Figure 3.5(b) shows the situation where the current cell P is at a higher level in the quadtree than its east (E) and north (N) neighbor. In an approach similar to that presented in Section 3.4.1, the discretization of the diffusion operator is shown in Equation (3.2). The calculation of flux from the east neighbor is presented here. A similar approach is implemented for the north neighbor. The fluxes from the cells are calculated as:

$$f_e = \frac{\psi_{E^*} - \psi_P}{\Delta x}(\Delta y)$$
$$f_n = \frac{\psi_P - \psi_{N^*}}{\Delta y}(\Delta x) \tag{3.9}$$

Again, E^* and N^* are ghost points where variable values ψ_{E^*} and ψ_{N^*} are interpolated from surrounding points for the configuration shown in Figure 3.5 as:

$$\psi_{E^*} = 0.5625\psi_E + 0.1875\psi_{P'} + 0.1875\psi_N + 0.0625\psi_{NE} \tag{3.10}$$

$$\psi_{N^*} = 0.5625\psi_N + 0.1875\psi_{P'} + 0.1875\psi_E + 0.0625\psi_{NE} \tag{3.11}$$

where $\psi_{P'}$ is interpolated bi-linearly from its child cells as

$$\psi_{P'} = 0.25(\psi_P + \psi_S + \psi_W + \psi_{SW}) \tag{3.12}$$

To summarize, the ghost point values are calculated as:

$$\begin{aligned}\psi_{E^*} &= 0.5625\psi_E + 0.046875(\psi_P + \psi_S + \psi_W + \psi_{SW}) + 0.1875\psi_N + 0.0625\psi_{NE} \\ \psi_{N^*} &= 0.5625\psi_N + 0.046875(\psi_P + \psi_S + \psi_W + \psi_{SW}) + 0.1875\psi_E + 0.0625\psi_{NE}\end{aligned} \tag{3.13}$$

Finally, the stencil for the Laplace operator at cell P is a function of all its surrounding cells and the discrete operator can be written as:

$$\nabla^2\psi_P = f(\psi_P, \psi_E, \psi_W, \psi_N, \psi_S, \psi_{NE}, \psi_{SW}) \tag{3.14}$$

3.5 Velocity Correction

Calculation of face velocities across mesh interfaces proceeds in a manner similar to that employed in discretizing the Laplace operator as described in the previous

sections. Cell face values are linearly interpolated between the cell center values and the 'ghost cell' values described in the previous sections. It may be noted that the cell face velocities are calculated in a similar fashion. The flow variables at the 'ghost cell' location are linearly interpolated from the surrounding cells.

In the first case shown in Figure 3.5(a) where west the neighbor cell is divided, let the west face value of a variable ψ be indicated by ψ_w.

$$\begin{aligned} \psi_w &= 0.5(\psi_{w1} + \psi_{w2}) \\ &\textit{where} \\ \psi_{w1} &= 0.5(\psi_{P_1^*} + \psi_{W1}) \\ \psi_{w2} &= 0.5(\psi_{P_2^*} + \psi_{W2}) \end{aligned} \tag{3.15}$$

Variables values at P_1^* and P_2^* are ghost cells. The variables $\psi_{P_1^*}$ and $\psi_{P_2^*}$ are interpolated at these locations from the surrounding cells by bi-linear weighted averaging as shown in the configuration shown in Figure 3.5(a) case by Equation (3.4) to Equation (3.7). Putting everything together, the face value ψ_w is interpolated from all the surrounding cells similar to that shown in Equation (3.8).

$$\psi_w = f(\psi_P, \psi_{W_1}, \psi_{W_2}, \psi_{W_3}, \psi_{W_4}, \psi_N, \psi_S, \psi_{NW}, \psi_{SW_1}, \psi_{SW_2}, \psi_{SW_3}, \psi_{SW_4}) \tag{3.16}$$

Similarly, in the case of a cell whose neighbor is at a lower level as shown in Figure 3.5(b), calculation of a variable on the east face, ψ_e proceeds as follows:

$$\psi_e = 0.5(\psi_{E^*} + \psi_P) \tag{3.17}$$

Again, E^* is a ghost point and variable value ψ_{E^*} is interpolated from surrounding points for the configuration shown in Figure 3.5(b) as shown in Equation (3.10). Finally, the stencil for calculation of the east face variable value for the cell P in Figure 3.5(b) is:

$$\nabla^2\psi_P = f(\psi_P, \psi_E, \psi_W, \psi_N, \psi_S, \psi_{NE}, \psi_{SW}) \tag{3.18}$$

Velocity correction proceeds at mesh interfaces by calculating the cell face values of pressure. Then gradient of pressure is then calculated using these cell face values and the velocities are corrected as detained in Equation (2.13).

3.6 Applying Boundary Conditions

In normalizing the refined mesh, boundary cells are refined to the same level as their adjacent interior cells as shown in Figure 3.6. The refinement proceeds in the same way as for interior cells with each boundary cell being divided into four child cells (instead of two) in 2D. This is done to avoid the complexities that arise in code implementation for the data structure as well as discretization schemes. However, as seen in the figure, of the four child cells, only two are effectively included in the discretization schemes. For instance, Figure 3.6 shows refinement at the east domain boundary. As seen in the figure, the NE and NE neighbors, indicated by E_3, E_4 and the SW and SE neighbors indicated by E_1, E_2 coincide. These cells have the same cell center locations and variable values. The same is also true of the west boundary. For the north and south boundaries, the NE and SE child cells and NW and SW child cells also coincide. In 3D, the boundary treatment proceeds in a similar way. The discretization at the boundary cells

is similar to that described in sections 3.4.1 and 3.4.2. However, in case of Neumann boundary condition the stencil is modified to implicitly apply the boundary condition.

For the situation shown in Figure 3.6, for the cell P, assembly of the east, west and north side fluxes is straightforward with the discretization following the standard central difference scheme because there are no cell level differences. However, the south neighbor of the cell is one level lower than cell P. The south side flux required for discretization as shown in Equation (3.2) can be written as:

$$f_s = \left(\frac{\psi_{S*} - \psi_P}{\Delta y} \right) \Delta x \tag{3.19}$$

S^* is a ghost point where variable values ψ_{S*} is interpolated from surrounding points for the configuration shown in Figure 3.6 as:

$$\psi_{S*} = 0.125(\psi_{E'} + \psi_{P'}) + 0.375(\psi_S + \psi_{SE}) \tag{3.20}$$

$$\psi_{P'} = 0.25(\psi_P + \psi_N + \psi_{NW} + \psi_W)$$
$$\psi_{E'} = 0.25(\psi_{E_1} + \psi_{E_2} + \psi_{E_3} + \psi_{E_4}) = 0.5(\psi_{E_1} + \psi_{E_3}) \tag{3.21}$$
since
$$\psi_{E_1} = \psi_{E_2}, \psi_{E_3} = \psi_{E_4}$$

On substituting Equation (3.21) into Equation (3.20), the ghost point value is obtained as:

$$\psi_{S*} = 0.0625(\psi_{E_1} + \psi_{E_3}) + 0.03125(\psi_P + \psi_N + \psi_{NW} + \psi_W) + 0.375(\psi_S + \psi_{SE}) \tag{3.22}$$

If Dirichlet boundary conditions are to be applied, then the ghost point value

obtain in Equation (3.22) is substituted into Equation (3.19) and the flux on the south face is calculated. If Neumann boundary conditions are to be applied then the stencil is modified to implicitly apply a zero-gradient condition at the interface. If the gradient at the domain boundary is zero, then for the case illustrated in Figure 3.6, the following is true:

$$\begin{aligned} \psi_{E_3} &= \psi_{E_4} = \psi_N \\ \psi_{E_1} &= \psi_{E_2} = \psi_P \\ \psi_{SE} &= \psi_S \end{aligned} \tag{3.23}$$

Applying the above condition in Equation (3.22), the ghost point value is modified as:

$$\psi_{S*} = 0.09375(\psi_P + \psi_N) + 0.03125(\psi_{NW} + \psi_W) + 0.75\psi_S \tag{3.24}$$

With this alteration, the assembly of fluxes and the discretization proceeds as described previously.

3.7 Representing Interfaces

Various approaches of representing the interface have been coupled with LMR for different applications. Agresar et al. [3] demonstrated the use of front tracking methods for interface representation with adaptive grids. Volume of Fluid (VoF) methods also have been applied for many cases involving flow across embedded boundaries as well as free surface flows [58-62, 110, 152, 162]. Interface reconstruction using different schemes have been discussed with the VoF approach. Strain [125-129] in a series of papers has demonstrated use of Levelsets to represent and move interfaces on quadtree

meshes. The issue of re-distancing and velocity extension on quadtree meshes has also been addressed. To minimize interpolation errors of all variables close to the interface and also to simplify the implementation of discretization schemes for the Levelset functions as well as the state variables at the interface cells, the present method adopts the scheme adopted by Sochnikov et al. [121], where cells close to the interface up to a certain distance are refined uniformly. The boundary layer build-up close to the immersed boundaries cause mesh adaptation in these regions even without forcibly doing so. Furthermore, with this method, the discretization scheme at the interfaces does not need to be altered and the same finite difference scheme can be maintained as detailed in Chapter 2 and previous work [92, 96, 142, 144, 159].

3.8 Solver Issues and Multigrid

On a locally refined grid, the cell numbering is no longer ordered and hence the coefficient matrix becomes highly sparse and asymmetric. Besides, the discretization scheme uses many interpolations from surrounding cells depending on the cell distribution structure and hence the penta- or septa- diagonal structure no longer exists even if the numbering is ordered. There is a need for efficient sparse matrix solvers for fast convergence of solution [152]. Structured solvers such as the LineSOR can no longer be used. Most researchers use sparse solvers such as the GMRES or the BiCGStab [94] or preconditioned conjugate gradient schemes. Agresar et al. [3] use SPARSKIT modules for incomplete factorization and conjugate gradients to solve the sparse matrix. Lasasso et al. [94] present a new discretization scheme for the pressure Poisson equation which ensures symmetric matrices that can be solved by ordinary solvers. Multilevel Multigrid methods have been explored [2, 153, 166] successfully to speed up the convergence of

the pressure solver. The Multigrid methods are found to be naturally amenable to locally refined meshes since the coarse grid information already exists within the grid hierarchy. Coarse grids can be formed by coarsening the tree structure. Connectivity and interpolation are easily done within the grid hierarchy.

Though the algebraic multigrid scheme with local coarsening is very efficient when applied to moving boundary simulations by the Sharp Interface Cartesian Grid method, it is found that when the mesh is refined and coarsened by the LMR scheme, building the coarse and fine meshes for the AMG solver needs to be performed each time the mesh and numbering is changed. Within the present framework of implementation of the AMG solver, this is found to be highly inefficient and time consuming. Improvement of the AMG scheme within the framework of the LMR scheme falls out of the scope of the present thesis and hence presently, a preconditioned BiCGStab is adopted as the solver. The convergence of the solver is not found to be significantly affected due to the highly sparse and unstructured nature of the coefficient matrix.

3.9 Interpolation Schemes

Often, variable values need to be interpolated onto locations other than at cell centers of the 'leaf cells' where the variables are stored. For instance, when new cells are formed by refinement or coarsening, the initial values on these new cells needs to be obtained by interpolation from the cells existing at the previous time step. Thus, another critical element of the local refinement method is the interpolation scheme. The current method uses a simple bi-linear weighted averaging over the leaf cells to interpolate variables. Figure 3.7 (a) shows a particular situation. The variables values at the point (x, y) in any configuration of surrounding cells are linearly interpolated from the cells

surrounding it as shown in the Figure 3.7(a).

3.10 Higher Order Discretization for Explicit Terms

In the solution of the scalar transport equation, the central difference scheme is not stable when the equation is advection dominated. In this case, higher order schemes have to be used for discretization of the convection terms. A 4^{th} order ENO type scheme is adapted for the discretization of the convection terms. At the mesh interfaces, as shown in Figure 3.7(b), four equidistant points in each direction are included into the discretization. Each of these points required in discretization are interpolated from the surrounding cells as detailed in Section 3.9.

3.11 Refinement Criteria

The third critical element that governs the local refinement scheme is the refinement criteria. Refinement criteria are selected based on the specific application under consideration. Cells may be refined either on geometry based or solution based criteria. The best choice of refinement criteria is one that gives a smooth distribution of the error throughout the domain.

The most commonly used criteria include solution gradient, curvature, presence of the boundary, local Peclet numbers and vorticity strength. For instance, Coirier et al. [37, 38] in their Cartesian grid based finite volume formulation for flows with embedded solid objects, use the local velocity divergence to detect compressibility and velocity curl to detect shear. Lasasso et al. [94] use multiple refinement criteria including object boundaries, optical depth and vorticity concentration. Trompert et al. [136] use a curvature monitor or second gradient based criterion to determine the cells that need to be

refined. Strain [125-129] uses a Levelset function based refinement criterion such that any cell whose edge length exceeds its minimum distance to the zero Levelset is split. Sochnikov [121] recommends refinement to the finest level up to a certain distance from the interface so as to avoid interpolation errors at critical locations close to the interface. Agresar et al. [3] also use the intersection by the interface and proximity to walls as refinement criteria besides additional criteria based on gradients of state variables. A detailed analysis of refinement criteria and effects is found in [41, 42]. In the present work, it is proposed to adopt Sochnikov's [121] method of uniform refinement up to a certain distance from the interface. In the present work first and second gradients of the flow variables are adopted as refinement criteria. Second gradients help identify boundary layers while first gradient identifies features such as shocks. Presently, cells are marked for refinement when

$$\max\left(\alpha_g \frac{\nabla\psi}{\psi_0} h, \alpha_c \frac{\nabla^2\psi}{\psi_0} h^2\right) > \varepsilon \tag{3.25}$$

where ε is the error tolerance, ψ_0 is a characteristic value of the variable ψ, $h \sim \Delta x$ is a characteristic length scale of the cell and α_g, α_c are the relative weights of the gradient and curvature based criteria respectively. α_g, α_c depend on which flow characteristics need to be resolved better for the specific problem. Resolving boundary layers requires the curvature based criterion to be heavily weighted while resolution of shocks requires gradient based criterion to carry heavier weight.

3.12 Overall Algorithm of the Local Mesh Refinement Scheme

The overall algorithm for the local refinement scheme is shown below:

1. Tag cells for refinement based on refinement criterion discussed in Section 3.11.
2. Create new cells or remove un-necessary ones based on step 1. If a cell is created, memory is allocated to the child cell pointers shown in the pseudo-code in Figure 3.3.
3. Re-build the neighbor connectivity information.
4. Normalize the grid to ensure that no two neighboring cells have a cell level difference of more than one. Boundary cells are necessarily of the same level as the adjacent interior cells. These operations may involve creation of new cells.
5. Re-build neighbor connectivity such that the newly created cells are included.
6. The new 'leaf' cells are re-numbered and a pointer array pointing to all the leaf cells is initialized on the newly formed mesh. Henceforth, unless absolutely necessary, all the operations are performed by this pointer array. Tree searches can be time intensive and hence operation through this pointer array is highly efficient.
7. Initialize flow variables on the newly created cells. Coarsened cells, which have their child cells removed are initialized with interpolated values of the removed cells. Newly created cells are initialized with interpolated values from the previously existing cells.
8. All the flow variables on cells other than the leaf cells which are on the top layer of the tree are de-allocated. Only the connectivity information is maintained on all the cells in the tree. All the rest of the variables seen in the pseudo code in Figure 3.3 are de-allocated.

9. Boundary conditions are re-set so that newly formed boundary cells have the right values.
10. All the interface parameters such as interface normal required by the interface advection scheme are re-generated.

3.13 Integrating the LMR Scheme with the Sharp Interface Solver

Integration of the local refinement algorithm into the Sharp Interface Cartesian Grid Solver discussed in Chapter 2 is shown in Figure 3.8. The initial rectangular mesh set-up is simplified by the integration of the local mesh refinement scheme. The only restriction is that the base mesh must be fine enough to resolve the rough shape of the immersed object such that three or four mesh cells are formed across the narrowest portion of the geometry. After setting up the base mesh, the Levelset is generated on this coarse mesh. The mesh is then refined uniformly to a pre-set number of levels in the region surrounding the interface. The Levelset field is then re-generated on the newly refined mesh. This ensures that the geometry is well resolved and the calculation proceeds in the normal fashion from this point on.

Flow calculation is allowed to proceed to a pre-set number of time steps to ensure that all the artifacts of the initial conditions are eliminated before adaptive mesh refinement is started. This ensures that no spurious refinement of mesh occurs because of numerical artifacts. As shown in the Figure 3.8, the local mesh refinement scheme is inserted into the overall algorithm before the Levelset advection and the flow solver. Local refinement operations shown in the previous section are essentially time consuming. Since the flow patterns and interface locations are not expected to change drastically at every time step, the refinement process is invoked periodically. If the

refinement scheme is operated every 10 time steps on an average it requires less than 5% of overall solve time depending on the problem at hand. Each time the mesh is refined or coarsened, variables values on the new cells are initialized by interpolation from the existing cells at the previous time step before proceeding to advect the Levelset and solve for the flow.

The particle tracking algorithm functions as previously except for the interpolation of flow quantities from the refined mesh as discussed in Section 3.9.

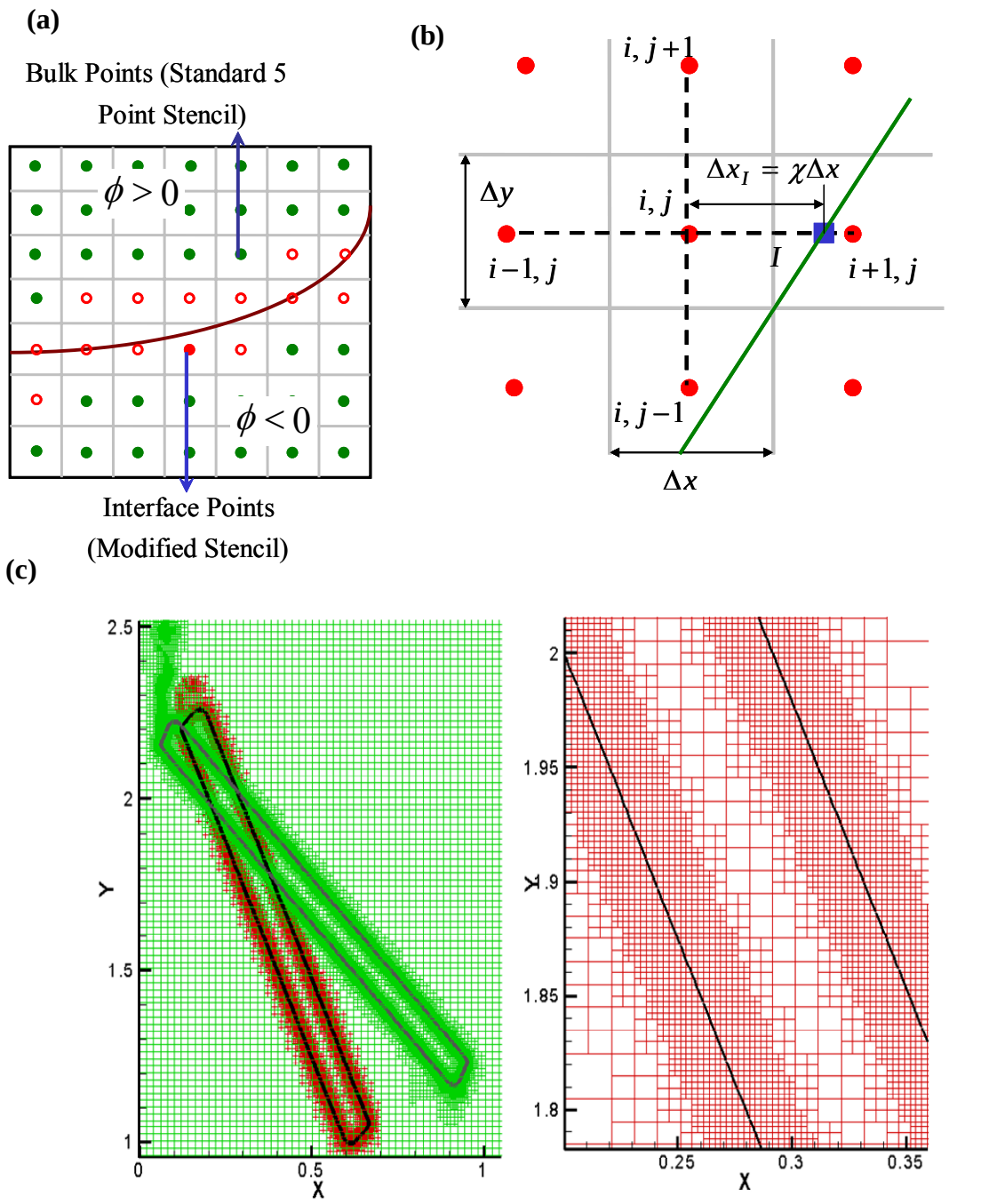


Figure 3.1. Sharp interface Cartesian grid method. (a) Demarcation of bulk cells and interface cells. (b) Details of discretization of interfacial cells. (c) Summary of Cartesian grid method with local mesh refinement.

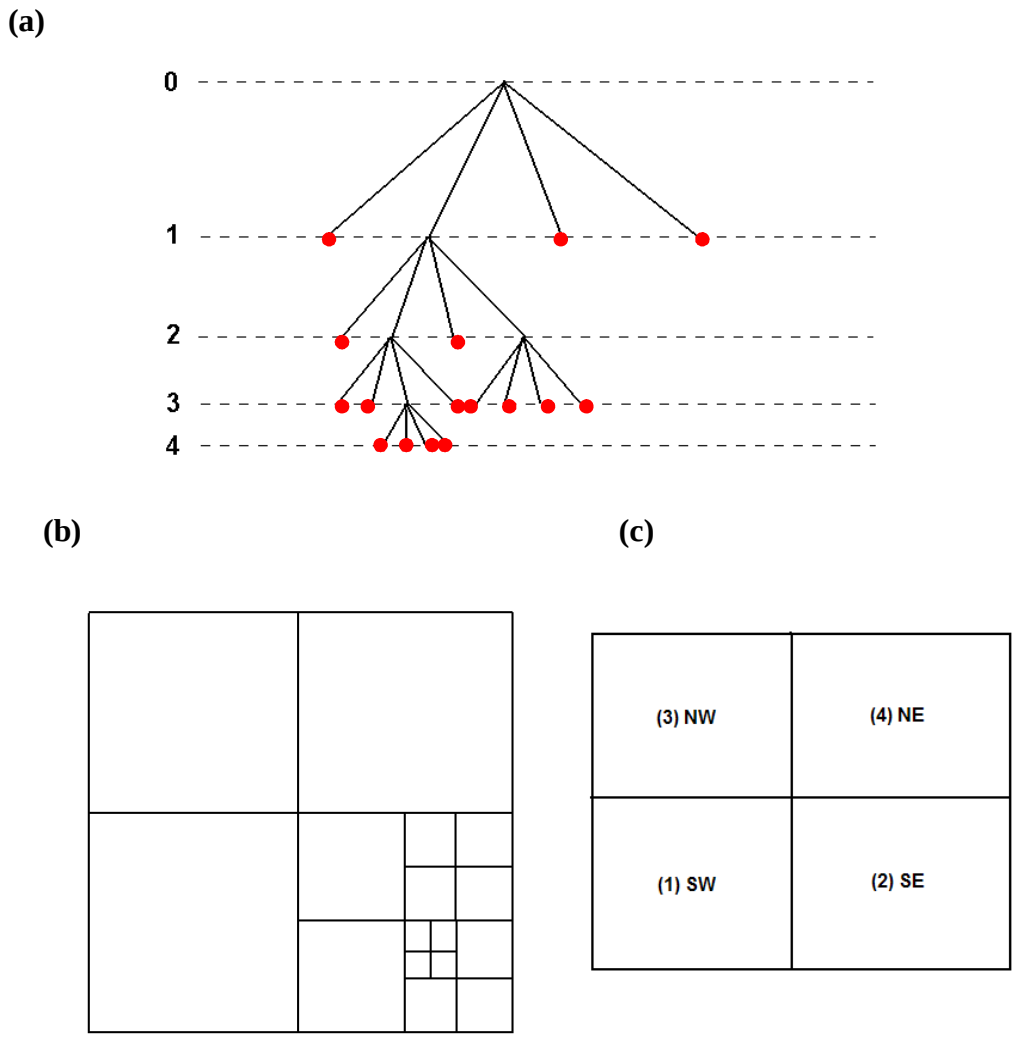


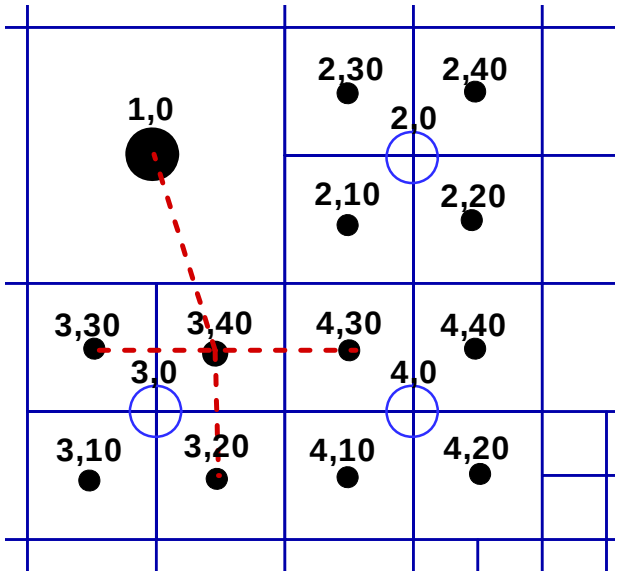
Figure 3.2. Data structure for the local refinement scheme. (a) Quadtree data structure. Leaf cells are marked with filled circles. (b) Locally refined mesh corresponding to the quadtree structure in (a). (c) Numbering scheme for child cells.

```
TYPE :: cell_type
     INTEGER, DIMENSION(2) :: index
     INTEGER :: lindex
     INTEGER :: level , status
     REAL :: u, v, phi, LS           !1,2,3,4
     REAL :: xc , yc
     REAL :: dxp , dyp
     TYPE(cell_type), POINTER :: parent
     TYPE(cell_ptr), DIMENSION(CURRENT:MAXNBR) :: nbrindex
     TYPE(node_ptr), DIMENSION(1:4) :: cellnode
     TYPE(cell_type), DIMENSION(:), POINTER :: child
     TYPE(node_type), DIMENSION(:), POINTER :: childnode
  END TYPE cell_type

  TYPE :: cell_ptr
     TYPE(cell_type), POINTER :: cptr => NULL()
  END TYPE cell_ptr
```

Figure 3.3. Code for quadtree implementation in FORTRAN 90. The above shows the structural components of each computational cell in 2D.

(a)



(b)

a) N neighbor of NE child : SE child of N neighbor of parent cell
b) E neighbor of NE child: NW neighbor of E neighbor of parent cell
c) W neighbor of NE child: NW child of parent cell
d) S neighbor of NE child: SE child of parent cell.

Figure 3.4. Demonstration of neighbor search operation. (a) Sample of a mesh interface. The process of finding the neighbors of cell (3,40) is illustrated in the text. (b) Thumb-rule for finding the neighbors of the NE child of a cell.

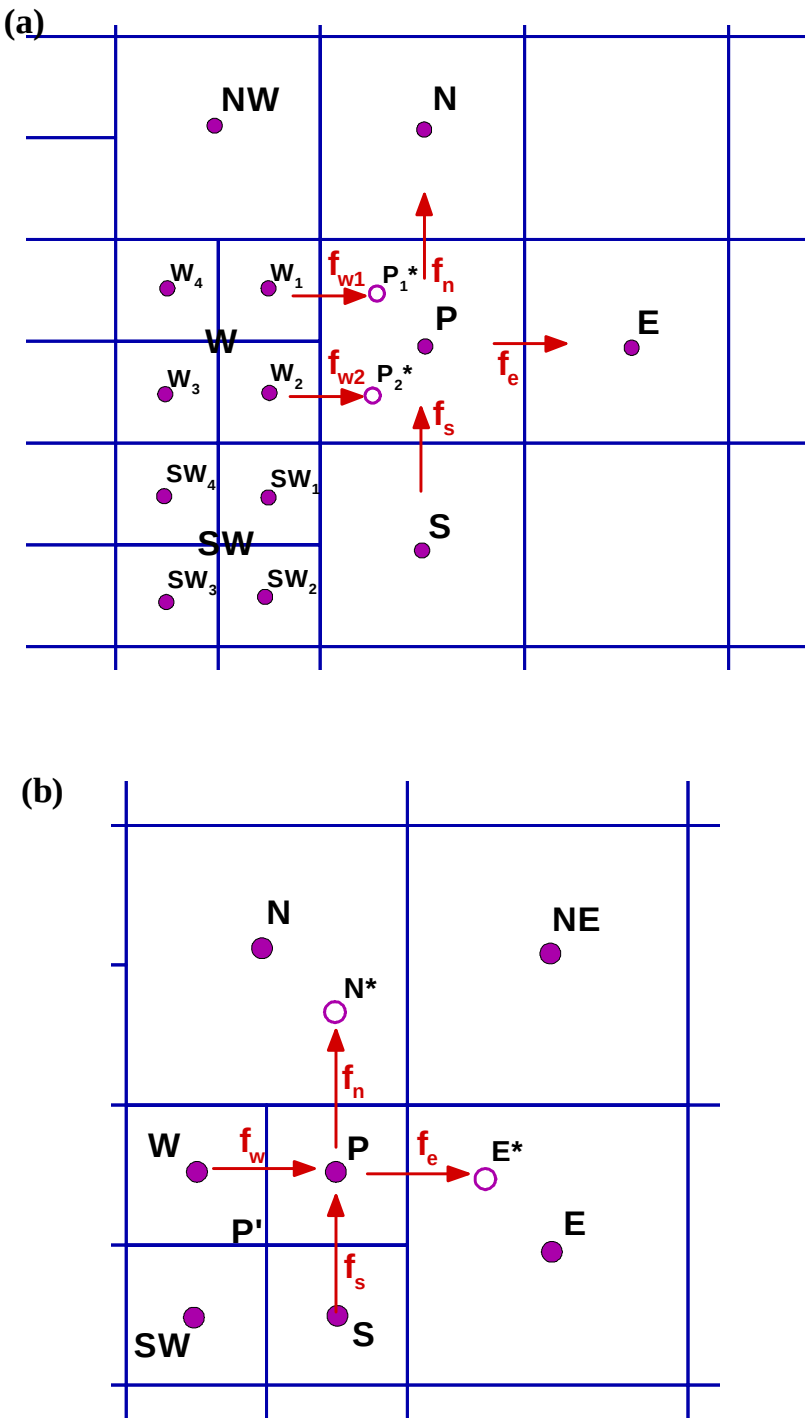


Figure 3.5. Two cases for which discretization has to be formulated. (a) Neighbor cell (WEST) is a level greater than current cell. (b) Neighbor cells (EAST and NORTH) are a level less than the current cell.

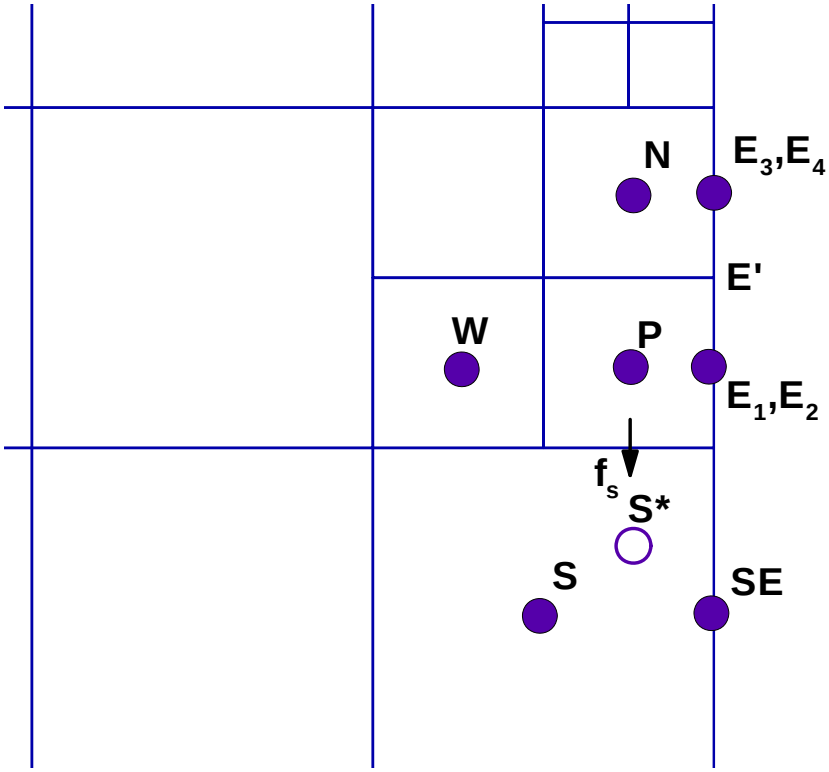


Figure 3.6. Treatment of boundary points and inclusion in the discretization scheme.

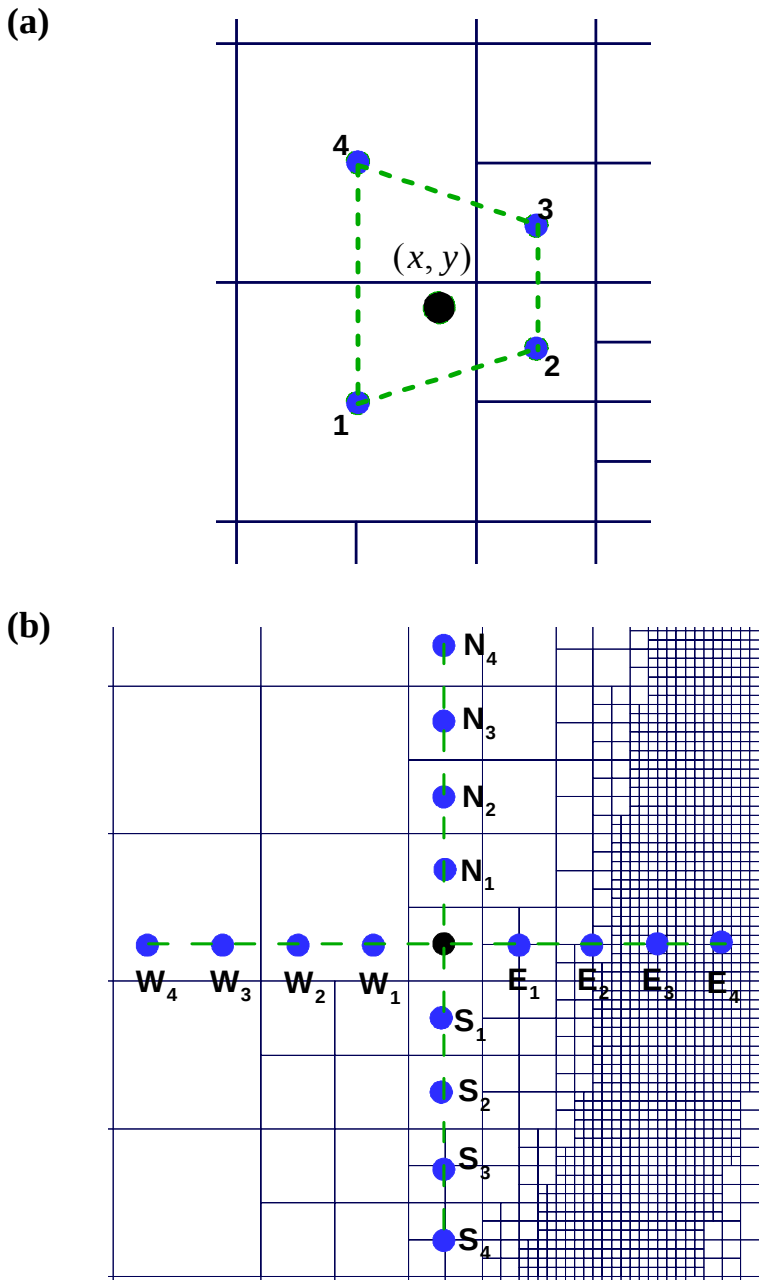


Figure 3.7. Higher order discretization. (a) Bi-linear interpolation at mesh interfaces. (b) Higher order scheme at mesh interfaces. Points used for discretization at point P are indicated with filled circles.

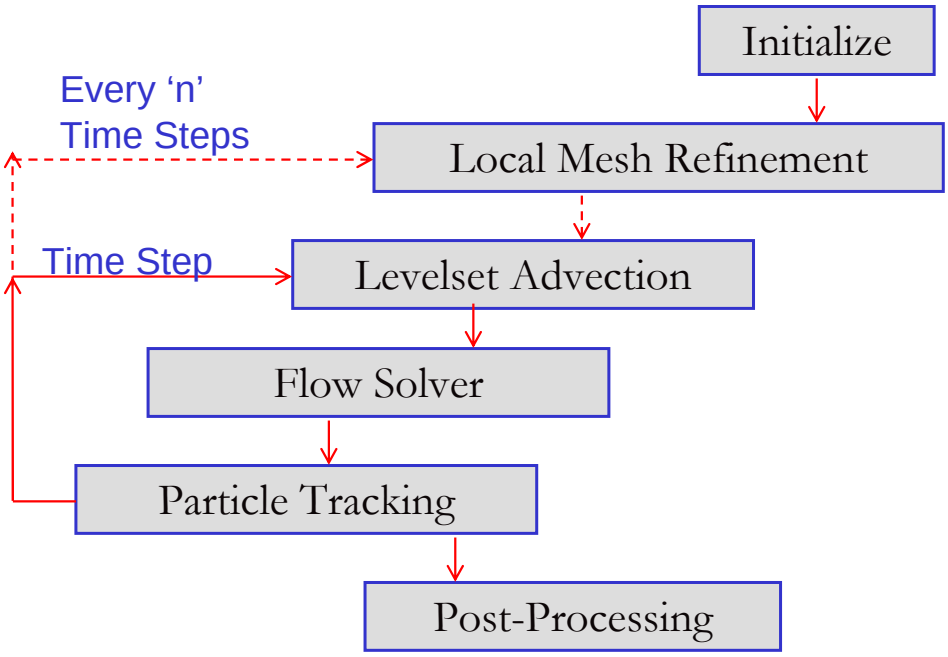


Figure 3.8. Inserting the local mesh refinement algorithm into the larger scheme of things.

CHAPTER 4

VALIDATING THE LOCAL REFINEMENT SCHEME

The local mesh refinement scheme implemented with the sharp interface Cartesian grid method is applied to a variety of applications and its efficiency and accuracy are tested. The scheme is then applied to a variety of problems that involve a wide range of length scales and which require heavy meshes (large number of computational cells) for computation in the absence of local refinement schemes. The main aim of this chapter is to demonstrate the efficacy and efficiency of the scheme when applied to a variety of moving boundary problems with sharp interface method. From the detailed analysis, the goal is to arrive at an estimate of required mesh density needed for accurate solutions, as well as the base mesh size and refinement levels to be used.

4.1 Validation and Timing Studies

Three benchmark problems ((i) lid-driven cavity flow, (ii) flow across cylinders and (iii) phase-change with dendrite growth) are considered to validate the local mesh refinement scheme. The main aim of this exercise is to optimize the scheme for base mesh cell size (Δx_{base}) and to determine the optimum number of refinement levels to optimize the fine mesh cell size (Δx_{ref}) and the tolerance limit (σ) to optimize the speed-up and accuracy obtained over a uniform fine mesh solution. When the mesh is coarsened to speed-up the solution process, it is natural that some accuracy is lost in the overall solution. The key is to find an optimum balance between solution accuracy and speed-up. The refinement, solution accuracy and speed-up are controlled by the above

parameters and hence they need to be optimized. The solution from the different exercises by varying the parameters is also tested for accuracy. The first set of studies is aimed at fixing the finest mesh resolution required to obtain grid independent solutions for three benchmark problems. The second set constitutes tests to optimize the base mesh size and the maximum number of refinement levels allowed, obtaining the fastest results for optimum accuracy. The third set of cases is aimed at fixing the tolerance limit for the refinement criteria that optimize speed up and accuracy. The parameters are optimized to obtain an error level of less than 5% in the solution.

4.1.1 Lid-Driven Cavity Flow

The lid-driven cavity flow is one of the most studied fluid problems and has been well documented [54]. This problem is chosen as the first test to benchmark the locally refined flow solver and to test its efficiency. Though this problem does not include any embedded boundaries, the factors that make it interesting for a locally refined calculation are the boundary layer separation and shear layers that arise from the moving lid of the cavity. As the flow evolves, the vorticity layers are seen to separate from the walls and concentrate in the center of the cavity. At steady state, a relatively large area of the flow domain has recirculating fluid. Based on the refinement criterion, a large fraction of the domain is likely to be refined at steady state as shown in Figure 4.1 (a) and Figure 4.1 (b) for $\mathrm{Re} = 1000$ and $\mathrm{Re} = 10000$ respectively. However, the local refinement scheme speeds up the computation time in the initial stages of calculation when the flow dynamics is in the developing stages. Though this problem does not demonstrate dramatic speed-up by using the LMR scheme, it is an ideal problem to ensure the accuracy and conservation properties of the scheme, particularly for mass conservation

on mesh boundaries. In this case, the characteristic velocity that controls the refinement criterion is fixed as 1.0, which is the horizontal velocity of the lid at the top boundary. Figure 4.2 shows the comparison of centerline velocities obtained from the current locally refined calculation with published data [54]. The data is presented for a base mesh size Δx_{base} = 0.08 and a maximum of 5 levels of refinement allowing the finest mesh cell size to be Δx_{ref} =0.005. Table 4.1, Table 4.2 and Table 4.3 document results of speed-up and accuracy studies for the lid-driven cavity flow. The first set of calculations (Table 4.1) are aimed at deciding the finest mesh size required to obtain grid independent solutions. Previous studies have reported [54] that a cell size of Δx_{ref} =0.0077 is adequate to obtain grid independent results. Starting with a base mesh size of Δx_{base}=0.16, the number of allowed refinement levels is increased until grid independent results are obtained. The solutions are compared with a uniform fine mesh (Δx_{ref} =0.005) calculation and L_2 norm of the error is calculated. The tolerance limit σ is specified as 5% for the refinement criterion. As seen from the table, a fine mesh cell size of Δx_{ref} =0.01 with 5 refinement stages is found to be adequate for grid-independent results. As the number of maximum refinement levels is increased, the speed-up obtained is decreased due to additional cells being added and tree operations increasing. Also, the percentage of time required for the refinement operations shows an increasing trend as the number of maximum refinement levels is increased.

For a maximum allowed 6 levels of refinement, almost 17% of the total CPU time is taken up by the refinement operations due to the quadtree searches. As the solution approaches steady state, the high vorticity region covers most of the computational

domain and most of the domain is covered by a fine mesh. Hence the large amount of time taken up by the refinement operations. For six levels of refinement, at steady state the number of cells is about half the number of cells in the case of a uniform grid (with the finest mesh spacing) calculation. It may be noted however that these observations are specific to the present driven cavity flow problem where the domain is closed and the vorticity produced within the domain cannot exit the domain. Table 4.2 shows optimization tests with respect to base mesh cells size while holding the fine mesh cell size fixed at Δx_{ref} =0.005. As seen from the table, 4 levels of refinement shows maximum speed-up. Beyond 4 levels, the tree operations slow down the calculation considerably. Table 4.3 shows tests to optimize the tolerance limit using base mesh size and finest mesh size optimized from 1(a) and 1(b). A tolerance limit of 10% is found to be adequate to obtain comparable error. This also shows a 10-fold speed up over a uniform mesh solution. In the lid-driven cavity case the refined solution and speed-up is found to be highly dependent on the base mesh cell size and number of refinement levels used.

4.1.2 Flow Across a Cylinder

The same tests above are performed for flow over an immersed cylinder. The sharp-interface method is tested in conjunction with the local mesh refinement scheme in this case for a stationary cylinder. The computational domain is 30x30 square units with an embedded cylinder with diameter of 1.0 unit. The domain size is much bigger than the cylinder size to avoid any effects of boundary conditions on the flow dynamics near the cylinder. Far field boundary conditions are fixed by the potential flow solution. The flow

variable that controls refinement is fixed as 1.0 which is the velocity at the flow inlet. The base mesh is uniform. Figure 4.3 shows the vorticity contours and corresponding refined mesh for flow over cylinders at $\mathrm{Re}=20$ and $\mathrm{Re}=300$. Figure 4.4 shows the drag and lift from the current refined mesh calculations for base mesh size of 0.64 and 5 levels of refinement. The steady state drag coefficient from previously published results [96, 160] at $\mathrm{Re}=20$ is between 2.03-2.08 and the length of the recirculation zone is $0.93d$ where d is the diameter of the cylinder.

From the grid refinement study shown in Table 4.4, a fine mesh cell size of $\Delta x_{ref}=0.04$ is seen to be sufficient to obtain grid independent results. A 100 time speed-up is obtained in this case with an error in calculated drag and length of recirculation zone of less than 1%. Unlike the lid driven cavity case, the fine mesh in this case is concentrated on a small portion of the domain, close to the cylinder and a very small fraction of cells are refined and the calculation speeds up corresponding to increasing base mesh cell size for the same size of the fine mesh. Table 4.5 shows studies to optimize the base mesh. The maximum base mesh cell size $\Delta x_{base}=0.64$ is fixed based on the size of the cylinder. The base mesh cell size has to be smaller than the cylinder size for the Levelset to be recognized in the initial step before a locally refined mesh is created. In this case, CPU time correspondingly decreases as the base cell size is increased without significant effect on the accuracy of the solution. Table 4.6 shows studies to establish the tolerance criterion to be used. A tolerance of 5% gives independent results with a 100-fold speed-up. A Reynolds number of 300 shows unsteady flow with periodic shedding of Karman vortices. Figure 4.3(b) shows the instantaneous vorticity contours and corresponding mesh. The shedding frequency corresponds to a

Strouhal number of 0.21 and average drag coefficient of 1.38 which correspond to previously published data [96, 160].

In the case of flow over immersed bodies, the high vorticity region is concentrated near the surface of the immersed object and this is the region that requires highly refined meshes. These kinds of problems are ideally suited to local refinement schemes for fast calculations.

4.1.3 Impulsively Started Cylinder

The impulsively started cylinder is studied as a prototype of unsteady separated flows and has been the subject of many theoretical, computational and experimental works [82]. Koumoutsakos and Leonard [82] in their work provide extensive information for quantities of interest such as vorticity field, streamlines and body forces that are valid for short times for a range of Re from 40 to 9500 through highly resolved numerical simulation by using vortex methods. Impulsively started flows present a serious challenge for numerical methods, because high resolution simulations are necessary for high Reynolds numbers to adequately resolve the singular character of the flow at early times and to resolve the details of the separation process. Two Reynolds numbers are considered in the current simulations. One on the lower end, with Re=1000 and one on the higher end with Re=9500. A rough estimate of boundary layer thickness by $\delta \approx 1/\sqrt{\mathrm{Re}}$ enables us to fix the minimum mesh size for the calculations. Typically about 10 cells are required within the boundary layer to capture the boundary layer dynamics accurately. This estimate is shown to obtain mesh independent results in the previous bench mark cases. For the simulation with Re=1000, a uniform base mesh with

a spacing of 0.64 is used. A rough estimate of boundary layer thickness shows that a fine mesh size of about 0.003 is required to capture the boundary layer dynamics.

Therefore, a maximum of 8 refinement levels corresponding to a cell size of 0.005 is allowed near the interface cells. The flow domain size is 30x30 units, while the cylinder diameter is 1. The large domain size relative to the cylinder size is required to minimize the effects of the far-field boundary conditions. For the simulation with Re=9500, a uniform base mesh with a grid spacing of 0.16 is used with 8 levels of refinement around the interface reducing the minimum cell size to 0.00125. The cylinder is impulsively started with a velocity of 1.0 and calculations in the first 5 non dimensional time units is recorded and compared with results from [82]. The cylinder velocity is the characteristic velocity used in the refinement criterion to control refinement. Figure 4.5 shows the flow development for Re=1000 for time T = 2.0, 4.0, 5.0 and 6.0 with flow patterns for the corresponding time instances presented by Koumoutsakos and Leonard [82]. The flow patterns correspond to those presented in [82].

Figure 4.6(a) shows body vorticity recorded in the current simulations compared with published results in [82] for Re=1000. Figure 4.6 shows the evolution of drag with time compared with published data. Figure 4.6 (c) and (d) show the body vorticity magnitude from the present simulations at various time instants presented in [82]. Figure 4.7 shows comparison of flow patterns for Re=9500 compared with published results in [82]. Figure 4.8 shows the same calculations for Re=9500. Figure 4.8 shows the evolution of flow at various time instants of the simulation. Figure 4.9 (a) shows instantaneous vorticity contours and corresponding refined mesh (b) for this simulation.

As seen in the figure, the calculation is well resolved with about 10 mesh points placed within the boundary layers. In Figure 4.9 (c) and (d) the boundary layer is zoomed in and details are shown. Figure 4.10 (a), (b) and (c) shows the body vorticity from the present calculation at the time instants presented in [82] for Re=9500.

Thus the method is seen to be efficient and robust in solving unsteady separated flow over a wide range of Reynolds numbers. It is capable of handling abrupt eruptions of the boundary layers associated with unsteady flows at high Reynolds numbers. These kind of flows typically need to be handled when dealing with moving boundary flows. Interplay of primary and secondary vortices are easily and efficiently handled by the algorithm as the mesh is seen to adapt according to the requirements of the solution.

4.1.4 Solidification

This problem involves phase change at the solid-liquid interface, controlled by diffusive transport, interfacial curvature and growth and instability of the interface. In this case, validation and optimization of the local refinement procedures can be performed by comparing the computed solution against theory. Microscopic solvability is the currently accepted theory for the growth of dendritic structures in solidification from the melt [135, 141]. For solidification from pure melts, the basic features of the theory are as follows. The Mullins-Sekerka instability [103] of a planar solidification front occurs in the case when the melt is under-cooled, i.e., the temperature of the melt T_∞ is depressed below the melting temperature of the planar front T_m. Linear stability analysis of phase fronts under such conditions predicts a continuum of solutions in the form of paraboloidal needle crystals, which were obtained by Ivantsov [72] as

$$\Delta = \sqrt{\pi Pe}\exp(Pe)erfc(\sqrt{Pe}) \tag{4.1}$$

where Δ is the dimensionless under-cooling $[\Delta = (T_m - T_\infty)/(L/C_p)]$. L is the latent heat and C_p is the specific heat at constant pressure, Pe is the tip Peclet number given by $Pe = \rho_{tip}V_{tip}/2\alpha$ where ρ_{tip}, V_{tip} are the radius and velocity of the paraboloidal dendrite tip and α is the thermal diffusivity. However, in nature, one observes unique crystal patterns that are selected by the growth conditions and material properties. The selection of a pattern from the continuum of unstable wavelengths becomes possible by introduction of surface tension, which provides a smoothing or restabilization mechanism and picks out a spectrum of solutions. The influence of the surface tension appears through the modulation of the interfacial temperature for solidification fronts with curvature by means of the Gibbs-Thomson condition (without including interfacial kinetics effects):

$$T_i = T_m - \frac{\gamma(\theta)T_m}{L}\kappa \tag{4.2}$$

In the above, κ is the interfacial curvature and the surface tension $\gamma(\theta)$ is a function of the crystalline anisotropy (θ being the angle made by the normal to the interface with the x-axis). The strength and directional dependency of the surface tension are determined by the structure of the solid formed. For a typical fourfold symmetric crystal (such as the popular transparent organic model material succinonitrile [70, 71], this function could assume the form [79, 111].

$$\gamma(\theta) = \gamma_0(1 - 15\varepsilon\cos 4\theta) \tag{4.3}$$

A unique operating point of the dendritic tip is then selected from the set of allowed solutions by crystalline anisotropy through the solvability mechanism [77] by stipulating that a material-dependent selection parameter σ^* be related to the tip radius and velocity as follows:

$$d_0 = \frac{\gamma_0 T_m C_p}{L^2} \tag{4.4}$$

The selection parameter σ^* is a function of the crystalline anisotropy ε and is typically of the form:

$$\sigma^* = \sigma_0 \varepsilon^{7/4} \tag{4.5}$$

The unique geometric features that are displayed by growing dendrites when a steady tip propagation state has been reached are the tip radius (ρ_{tip}) and velocity (V_{tip}), as shown in Figure 4.12. Equations (4.1-4.4) above provide the necessary relations for obtaining these two unknown selected parameters in terms of the control parameters, namely, the under-cooling Δ, capillarity parameter d_0, and anisotropy strength ε.

Details of model implementation and discretization to include interfacial boundary conditions are presented in [158]. The calculations for benchmarking the solver are reproduced here with local mesh refinement. The current calculation corresponds to an under-cooling $\Delta = 0.55$, capillarity parameter $d_0 = 0.5$ and anisotropy strength of $\varepsilon_0 = 0.03$. The analytical values of non-dimensional steady state tip velocity V_{tip} is 0.0111 and tip radius ρ_{tip}=23.3 [135]. Figure 4.11 shows the growth of the dendrites and corresponding fine meshes. Figure 4.11 (a) shows the mesh used in the simulation of

dendrite growth for comparison with the microscopic solvability theory. As seen in the figure the overall size of the domain is 4x4 with 2x2 portion covered by a uniform mesh. In the remaining part, a stretched mesh with about 50 points in each direction is used. The seed is a quarter circle of radius 0.05 placed at the origin, growing a quarter of the dendritic crystal. The bottom and left faces are symmetric while the top and right faces have a fixed temperature of -0.55. The characteristic variable that controls the refinement is 0.55 based on the boundary temperature.

As the simulation proceeds, fine meshes are built around the interface based on thermal gradients. Figure 4.11 (b) shows that the method also captures the growth of the dendritic side branches. Figure 4.12 shows the comparison of calculated tip velocity, tip radius and tip selection parameter with analytical solution of the microscopic solvability theory. The results presented are for a base mesh size of 0.026 with four levels of refinement. Table 4.7, Table 4.8 and Table 4.9 show the results on the dendrite growth simulation and comparison with microscopic solvability theory. Table 4.7 shows the results of grid refinement study. A fine mesh resolution of Δx_{ref} =0.0033 is found to be adequate to bring the error in tip velocity within 5% and tip radius to within 1% for a 5 time speed-up over uniform fine mesh solution. Table 4.8 shows studies to optimize base mesh cell size. The error in tip radius and tip velocity is around 5% till 5 levels of refinement and comparable to the fine uniform mesh solution. However, it was found necessary to limit the tolerance limit to 1% in these calculations because of the relatively thick boundary layer. When compared with the previous flow calculations, this calculation is diffusion driven while the flow calculations were convection controlled with a thinner boundary layer. Hence, the difference in the required tolerance limit to

ensure less than 5% error in solution. Increasing the refinement tolerance to 5% for 5 levels of refinement increases the error in tip radius and tip velocity to more than 5%. The difference between this simulation and the previous flow simulations is that this is diffusion dominated unlike the previous cases which were advection dominated. From Table 4.9, a tolerance of 10% is found to be sufficient to achieve less than 5% error in solution.

4.2 Other Applications

Based on the insights obtained from the benchmark calculations discussed in the previous section, the algorithm is demonstrated on a variety of applications involving moving boundaries. All the applications demonstrated below involve steep flow gradients or constricted geometries. In each case, resolution of the finest features requires use of very fine meshes making these calculations computationally very slow and inefficient in the absence of local refinement schemes.

4.2.1 Simulation of Mixing in the GI Tract

The pylorus (meaning 'gatekeeper') is a star shaped valve like structure at the junction of the stomach and duodenum. This organ is a critical part of the digestive system with many functions. The Pylorus is controlled by a complex sphincter muscle that sets the resistance to the bulk fluid flow out of the stomach (by affecting the resting diameter or tone of the open/relaxed outlet), and closes the outlet intermittently, by a forceful contraction of the pyloric segment that occurs as the peristaltic contraction of the stomach invades the terminal antrum. In man, when fully relaxed, the pyloric orifice forms a narrow waist between the much wider duodenal bulb and the antrum (luminal

diameter about 1 cm, length less than 0.5 cm). Closure of the lumen is completed by mucosal folds, which narrow the lumen into a star-like slit, then plug it. If the pylorus malfunctions, the stomach discharges larger particles into the intestines. Discharge of larger particles by the stomach in turn leads to failure of digestion and absorption in the intestines. Pylorus and duodenal bulb are the high ridge over which fluids swap out of the stomach into the depending part of the duodenum. A second function is that the pylorus prevents bile reflux into the stomach.

In the present study, the mechanics of flow through the pylorus and its effect on mixing for the homogenization and the effective exposure of gastric chyme to pancreatic and biliary secretions is examined. The current simulations model the pylorus as a notch like structure in 2D. The notch functions like a valve opening and closing synchronous with pulsating inlet of fluid. A typical Reynolds number calculated from data obtained for the cat-gut is around 100. Resolving the fine boundary layers and vortex interaction requires very fine meshes. The local mesh refinement scheme speeds up the calculation significantly by optimizing usage of meshes for the calculation.

The action of vortices generated at the pylorus is responsible for the complex fluid dynamics that aids in rapid mixing of chyme. A marker fluid is placed upstream of the pylorus and advected with the flow by solving the scalar transport equation with low diffusion coefficient. A 4^{th} order ENO scheme discussed in Section 3.10 is adopted along with the local refinement scheme for discretizing the advection terms in the scalar transport equation. Figure 4.13 shows vorticity contours and the mixing of the marker dye with the flow at different time instances. Note that the flow is entirely laminar, but within a distance of a few centimeters of the pylorus the marker fluid is fully mixed.

Panels (c), (d) and (e) of Figure 4.13 show the complex vortex structures and vortex break-up into small structures captured by the calculation as the flow progresses.

The main reason for the intense mixing in this region is the break up of the jet emerging from the pylorus notch due to interaction with the boundary layer from the wall and its subsequent break-up into smaller structures. The jet emerging from the pylorus hits the top wall and is reflected back into the duodenal bulb. Interaction with the wall boundary layer cause break up into small structures while also turning the jet and causing it to impinge against the bottom wall with further break-up. Small vortex structures cause the intense mixing as seen from the evolution of the marker fluid. The main reason for the complex flow dynamics in this region is the asymmetry of the flow emerging from the pylorus notch. The small dimension of the closed pylorus causes a strong jet to emerge from the notch. The jet is directed upwards because of the asymmetry of the notch geometry. The upwardly directed jet then hits the top wall and draws in the boundary layer from the top wall. Asymmetry is thus seen to be a major factor that aids in mixing and this insight may also have significance in the design of engineering systems. Figure 4.14 shows instantaneous vorticity contours and corresponding refined mesh for a well resolved and optimized calculation.

4.2.2 Particles Settling Under the Effect of Gravity

The efficacy of the local refinement scheme is demonstrated in an application involving multiple moving immersed objects of varying dimensions. 20 cylinders with a median radius of 0.4 and median object to fluid density ratio of 1000 are placed in a domain of size 10x50. The cylinders are allowed to fall freely under the effect of gravity and pressure and shear forces acting on them. Initially the fluid is stationary. Once the

cylinders are released and allowed to fall, flow is induced by the effect of the falling cylinders. As the flow develops, gravitational force as well as fluid forces govern the movement of the particles. Reynolds number is varied as $Re=10$, 100 and 1000 by altering the fluid viscosity suitably. The cylinder length scale is about $1/100^{th}$ of the domain size. To capture the fine flow features, especially at higher Re, the small length scales need to be adequately resolved. Use of stretched meshes is not practical for this case, because the cylinders pass through the entire domain at some point in the simulation and stretched meshes can cause fine length scales to be diffused because of numerical diffusion. Without the LMR scheme, this is a very heavy calculation requiring cells of the order of million cells allowing about 20 cells across the cylinder diameter. With the LMR scheme, refining to a maximum of 5 levels of refinement and coarsening the base mesh accordingly the average number of cells required for the calculation is reduced by 3 orders of magnitude with corresponding speed-up of the unsteady calculation. Most of the critical region as far as the flow is concerned is restricted to the region around the cylinder and this region is automatically detected by the refinement criterion. The area where there are no large gradients is coarsened.

Figure 4.15 shows three stages in the calculation for the three Reynolds numbers. As seen in the figure, the calculation at $Re=10$ is characterized by thick boundary layers and high drag. The higher drag causes the larger cylinders to slow down as they descend in the fluid, causing the cylinders to cluster and collide against each other. Elastic collisions are assumed between the cylinders and in the event of collision of cylinder 1 with cylinder 2, the velocities of the two objects are modified based on elastic collision theory as:

$$V_{1,new} = \frac{m_1 - m_2}{m_1 + m_2} V_1 + \frac{2m_2}{m_1 + m_2} V_2$$
$$V_{2,new} = -V_{1,new} \tag{4.6}$$

where, m_1, m_2 are the masses and V_1, V_2 are the velocities of the two colliding objects respectively. $V_{1,new}$ is the velocity of object 1 after the collision.

The simulation with Re=100 shows longer wakes behind the cylinders and beginning asymmetry in the wake. No distinct shedding of Karman vortices is observed at this Reynolds number. The third case with Re=1000 shows intense shedding of Karman vortices. The high Reynolds number is characterized by thin boundary layers and periodic shedding of vortices. At Re=1000, the particles get caught up in the wake of the particles in front and the shedding vortices are clearly captured in the current scheme. Figure 4.16 shows the vorticity contours and the refined mesh at a particular time instant in the simulation. As seen, the LMR scheme captures the intricacies of the boundary and shear layers at different scales with great detail in a highly optimized and accurate calculation.

4.3 Conclusions and Discussions

A local mesh refinement scheme is implemented complementing the Eulerian Levelset based Sharp Interface Method for fast and efficient simulations of moving boundary problems. The mesh is adapted based on solution gradient and curvature and naturally captures shocks and boundary layers. No user intervention is necessary at any point in the calculation except in determining the initial parameters that control the refinement process. The method is applied to a variety of benchmark problems involving

moving and stationary boundaries and its accuracy and efficiency are demonstrated. The parameters that control the refinement process are optimized to obtain maximum speed-up while maintaining less than 5% error in the solution. Depending on the application, 5-100 times speed-up is recorded compared to uniform fine mesh calculations.

For some applications such as the mechanical heart valve simulations, local mesh refinement algorithm is essential to obtain reasonable results in real time. Particularly, the simulations require highly resolved meshes in some localized regions of the domain. The efficacy of the current method in these simulations is demonstrated in the following chapters.

Fine Mesh Cell Size, Δx_{ref}	**Maximum number of allowed refinement levels (** n_{ref} **)**	**Number of computational cells/ Number of cell in uniform fine mesh.**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**Error Compared to Uniform Fine Mesh Solution**
0.16	1	0.00157	5961.6	0.34	5.53e-3
0.08	2	0.00627	1471.9	3.8	4.60e-3
0.04	3	0.023	357.45	4.6	2.43e-3
0.02	4	0.06	83.53	5.5	8.29e-4
0.01	5	0.15	16.29	7.6	3.07e-4
0.005	6	0.4186	4.14	17.37	3.01e-4

Table 4.1. Optimizing finest mesh size for Lid Driven Cavity Flow at Re=1000. Optimized refined mesh size for maximum speed-up for less than 5% error is found to be Δx_{ref} =0.005.

Note: CPU time is compared with a uniform fine mesh (Δx_{ref} =0.005) calculation. Base mesh size, Δx_{base} = 0.16 and tolerance limit σ = 5% are held fixed while maximum number of allowed refinement levels n_{ref} is varied.

Base Mesh Cell Size, Δx_{base}	**Maximum number of allowed refinement levels (** n_{ref} **)**	**Number of computational cells/ Number of cell in uniform fine mesh.**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**Error Compared to Uniform Fine Mesh Solution**
0.005	1	1.0	1.0	0.0	0.0
0.01	2	0.3229	2.52	1.59	3.61e-5
0.02	3	0.1735	7.10	2.57	1.22e-4
0.04	4	0.1720	8.32	4.33	2.45e-4
0.08	5	0.2488	7.19	7.53	2.96e-4
0.16	6	0.4186	4.14	17.37	3.01e-4

Table 4.2. Optimizing base mesh size for Lid Driven Cavity Flow at Re=1000. Δx_{ref} =0.005, σ = 5% are held fixed and Δx_{base} and n_{ref} are varied. Optimized base mesh size is found to be Δx_{base}=0.04.

Specified tolerance limit, σ	Number of computational cells/ Number of cell in uniform fine mesh.	Relative CPU Time (Speed-Up)	% of Time Required for Refinement Operations	Error Compared to Uniform Fine Mesh Solution
10%	0.15	10.91	3.28	2.51e-4
5%	0.17	8.32	4.33	2.45e-4
1%	0.51	1.02	5.01	2.34e-4

Table 4.3. Optimizing tolerance limit for Lid Driven Cavity Flow at Re=1000. Δx_{base}=0.04, Δx_{ref} =0.005, n_{ref} =4 and σ is varied. From Tables 4.1, 4.2 and 4.3, optimized parameters for maximum speed-up for less than 5% error in solution are Δx_{base}=0.04, Δx_{ref} =0.005, n_{ref} =4 and σ = 10%.

Fine Mesh Cell Size, Δx_{ref}	**Maximum number of allowed refinement levels** (n_{ref})	**Number of computational cells/ Number of cell in uniform fine mesh.**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**% Error in Drag Coefficient**	**% Error in Length of re-circulation zone**
0.64	1	0.0016	6089.7	0.0	49	-
0.32	2	0.0036	3479.18	3.7	29	-
0.16	3	0.00815	2551.5	3.9	14	4.3
0.08	4	0.010	1448.09	4.6	4.3	2.1
0.04	5	0.019	114.04	4.51	0.14	0.035
0.02	6	0.028	87.07	5.08	0.14	0.03

Table 4.4. Flow across cylinder, $\mathrm{Re} = 20$. Optimizing finest mesh size. $\Delta x_{base} = 0.64$, $\sigma = 5\%$ are fixed and n_{ref} is varied.

Note: Drag force of the cylinder and length of the wake are compared with published results [96, 160]. The drag force on the cylinder is expected to be between (2.03-2.08) while the length of the wake is around $0.93\,d$, where d is the diameter of the cylinder. Note that the cylinder diameter is 1.0 and this fixes the maximum allowed base mesh cell size.

Base Mesh Cell Size, Δx_{base}	**Maximum number of allowed refinement levels** (n_{ref})	**Number of computational cells/ Number of cell in uniform fine mesh.**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**% Error in Drag Coefficient**	**% Error in Length of recirculation zone**
0.04	1	1.0	1.0	0	0.096	0.001
0.08	2	0.25	2.23	1.05	0.096	0.004
0.16	3	0.067	12.94	1.63	0.14	0.016
0.32	4	0.022	52.68	2.64	0.14	0.03
0.64	5	0.019	114.04	4.51	0.14	0.035

Table 4.5. Flow across cylinder, $\mathrm{Re} = 20$. Optimizing base mesh size. Δx_{ref} =0.04, σ = 5%, Δx_{base} and n_{ref} varied. The method shows scalable speed-up with increasing base mesh size.

Specified tolerance limit, σ	**Number of computational cells/ Number of cell in uniform fine mesh.**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**% Error in Drag Coefficient**	**% Error in Length of recirculation zone**
10%	0.011	165.57	3.95	0.0	1.3
5%	0.019	114.04	4.51	0.14	0.035
1%	0.053	45.8	6.71	0.14	0.02

Table 4.6. Optimizing tolerance limit for flow across cylinder with Re=20. Δx_{base}=0.64, Δx_{ref} =0.04, n_{ref} =4 and held fixed while σ is varied. Δx_{ref} =0.04, σ=10% are found to be the optimum values for less than 5% error and maximum speed-up for this calculation from Tables 4.4, 4.5 and 4.6.

Fine Mesh Cell Size, Δx_{ref}	**Maximum number of allowed refinement levels (** n_{ref} **)**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**% Error in Tip Radius**	**% Error in Tip Velocity**
0.052	1	680.27	0	51.8	150.9
0.026	2	100.70	3.03	45.5	103.8
0.013	3	16.84	3.12	27.3	51.6
0.00667	4	4.93	3.47	19.1	28.9
0.0033	5	2.07	4.54	1.8	5.4

Table 4.7. Dendrite growth simulation and comparison with microscopic solvability theory [135]. Optimizing finest mesh size, Δx_{ref} . Δx_{base} = 0.052, σ = 1% are held fixed and Δx_{ref} is varied. Optimum mesh size is found to be Δx_{ref} =0.0033.

Note: For the simulation conditions ($\Delta = 0.55, d_0 = 0.5, \varepsilon_0 = 0.03$), tip velocity is expected to reach a steady value of 0.011 and tip radius is expected to reach a steady value of 23.3. Initial seed diameter of 1.0 fixes the maximum Δx_{base}.

Base Mesh Cell Size, Δx_{base}	**Maximum number of allowed refinement levels (** n_{ref} **)**	**Relative CPU Time (Speed-Up)**	**% of Time Required for Refinement Operations**	**% Error in Tip Radius**	**% Error in Tip Velocity**
0.0033	1	1	0	0.45	3.4
0.0667	2	2.83	3.2	0.73	4.3
0.013	3	4.45	3.3	0.91	4.7
0.026	4	4.93	3.47	1.36	4.9
0.052	5	5.07	3.5	1.8	5.4

Table 4.8. Dendrite growth simulation and comparison with microscopic solvability theory [135]. Optimizing base mesh size. Δx_{ref} =0.0033, σ = 1%.

Specified tolerance limit, σ	Relative CPU Time (Speed-Up)	% of Time Required for Refinement Operations	% Error in Tip Radius	% Error in Tip Velocity
10%	3.5	2.5	7.2	8.4
5%	4.45	2.9	4.5	6.7
1%	5.07	3.5	1.8	5.4

Table 4.9. Optimizing tolerance limit for Dendrite growth simulation and comparison with microscopic solvability theory [135]. Δx_{base}=0.052, Δx_{ref}=0.0033 are fixed and tolerance limit is varied. Optimized parameters from Tables 4.7, 4.8 and 4.9 are found to be Δx_{ref}=0.0033, Δx_{base}=0.052, σ=5%.

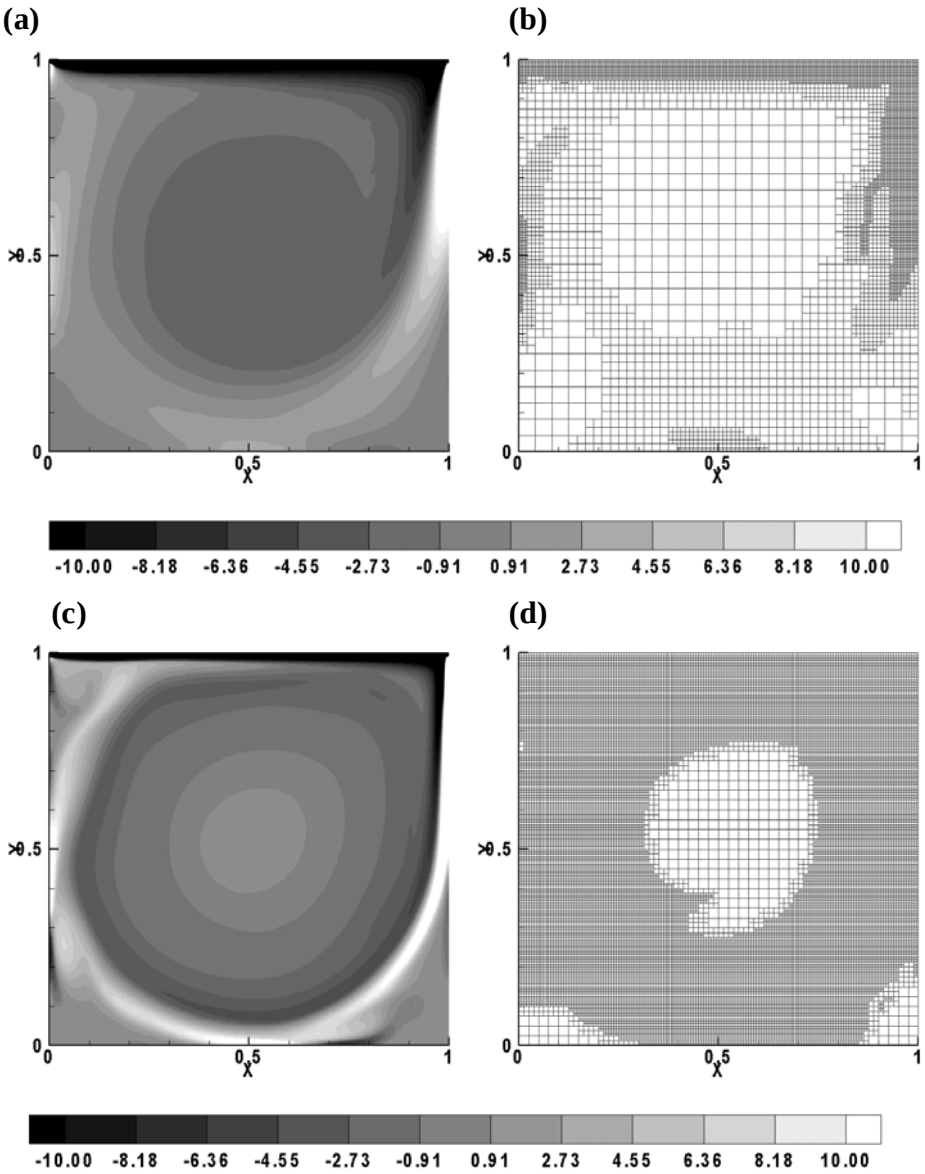


Figure 4.1. Vorticity contours and corresponding refined mesh. (a) Re=1000, vorticity contours at steady state. (b) Refined mesh corresponding to (a). (c) Re=10000, vorticity contours at steady state. (d) Refined mesh corresponding to (c).

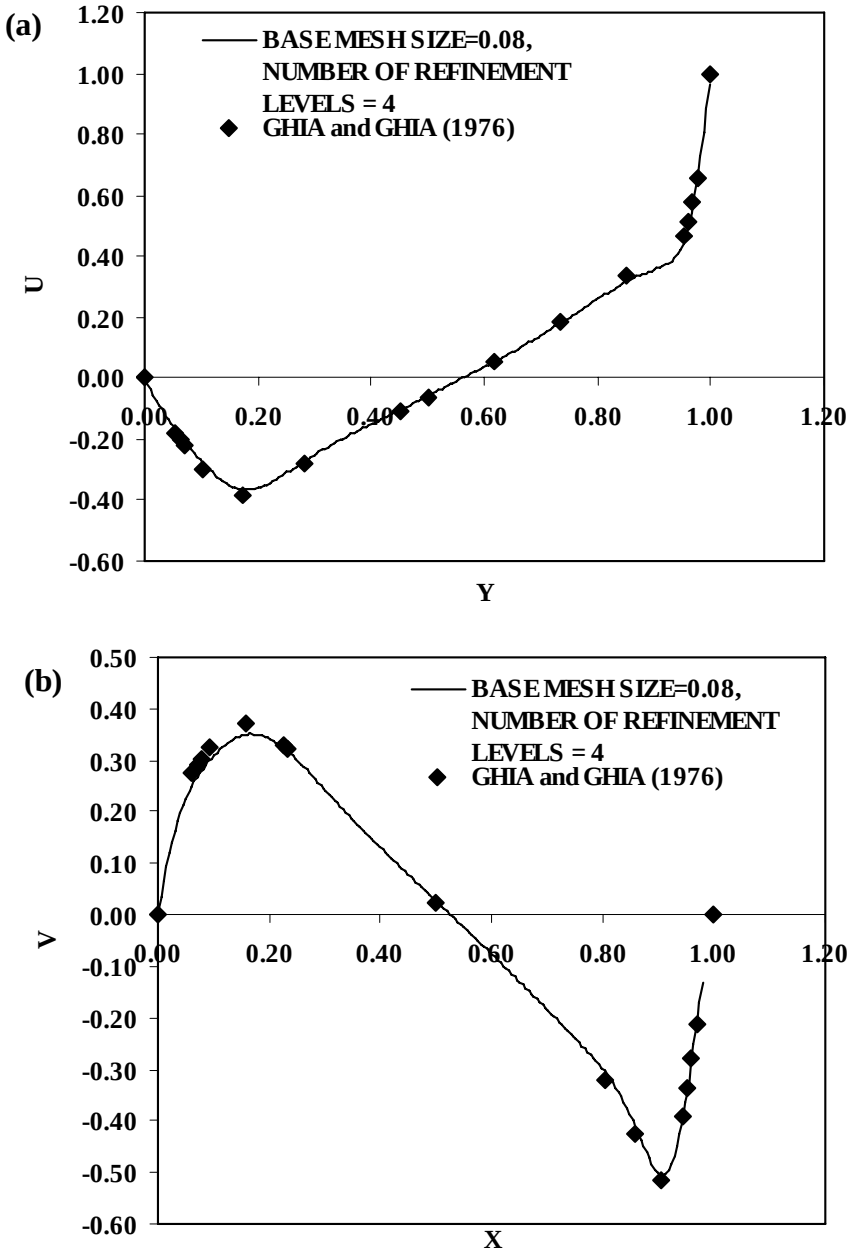


Figure 4.2. Lid-driven cavity flow at Re=1000. (a) Comparison of centerline u-velocity with published results [54] for different based meshes and refinement levels. (b) Comparison of centerline v-velocity with published results.

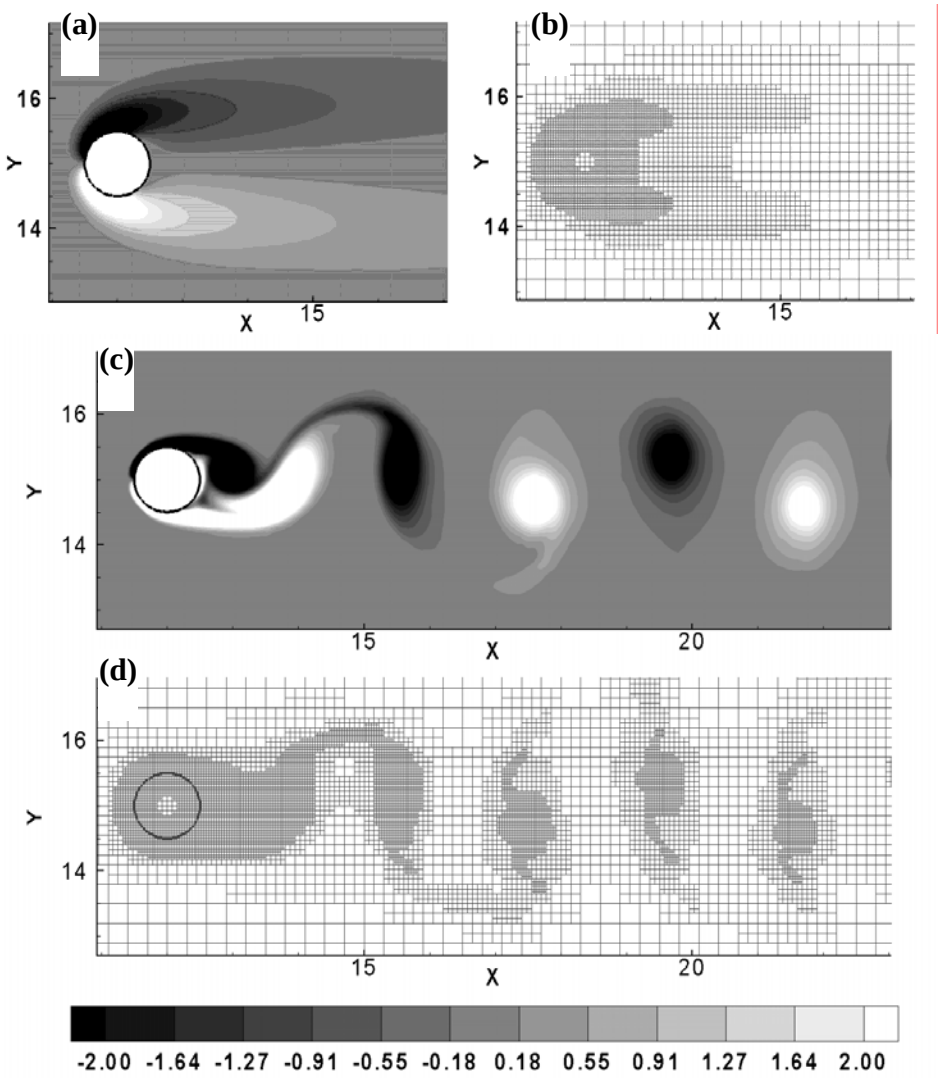


Figure 4.3. Flow over cylinder. (a) Steady state vorticity contours for Re=20. (b) Refined mesh corresponding to (a). (c) Instantaneous vorticity contours for Re=300. (d) Refined mesh corresponding to (c).

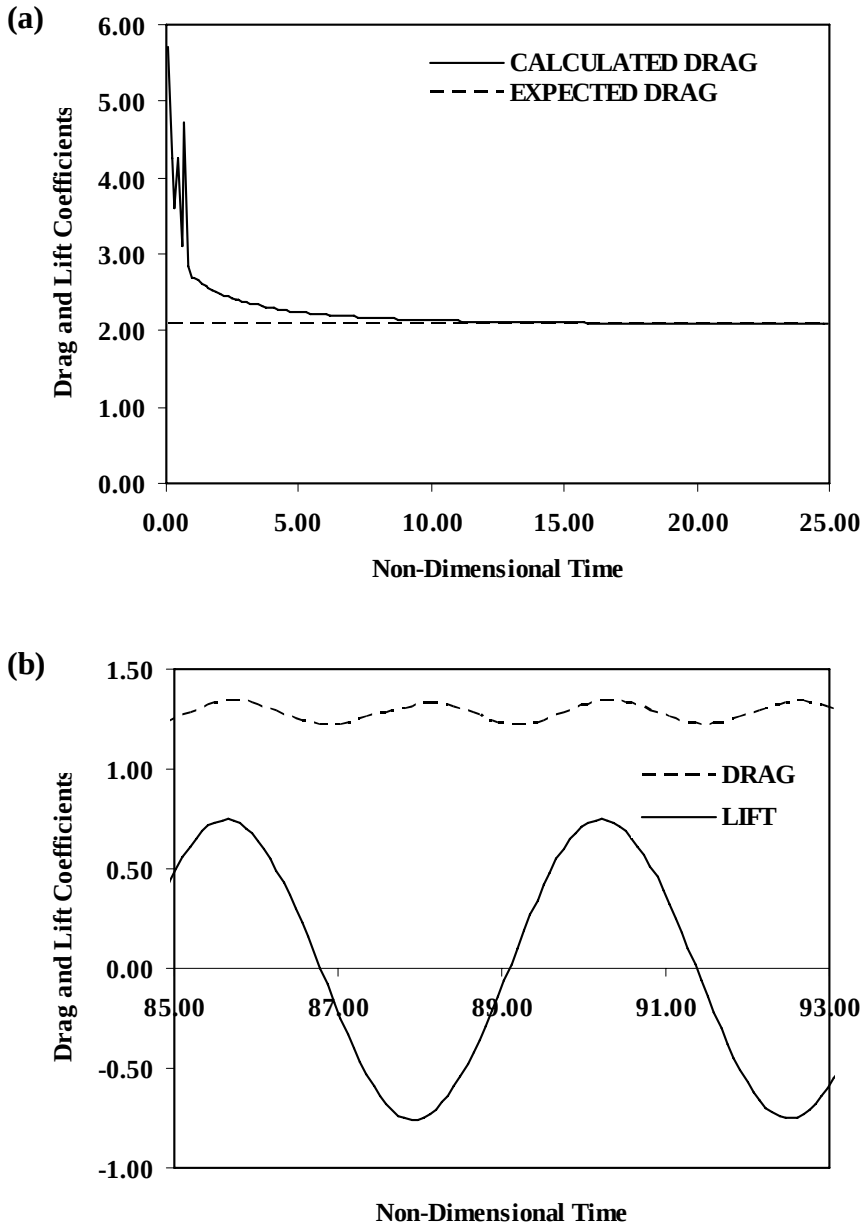


Figure 4.4. Comparison with validated results. (a) Re=20, (b) Re=300. Drag and lift are compared with validated results.

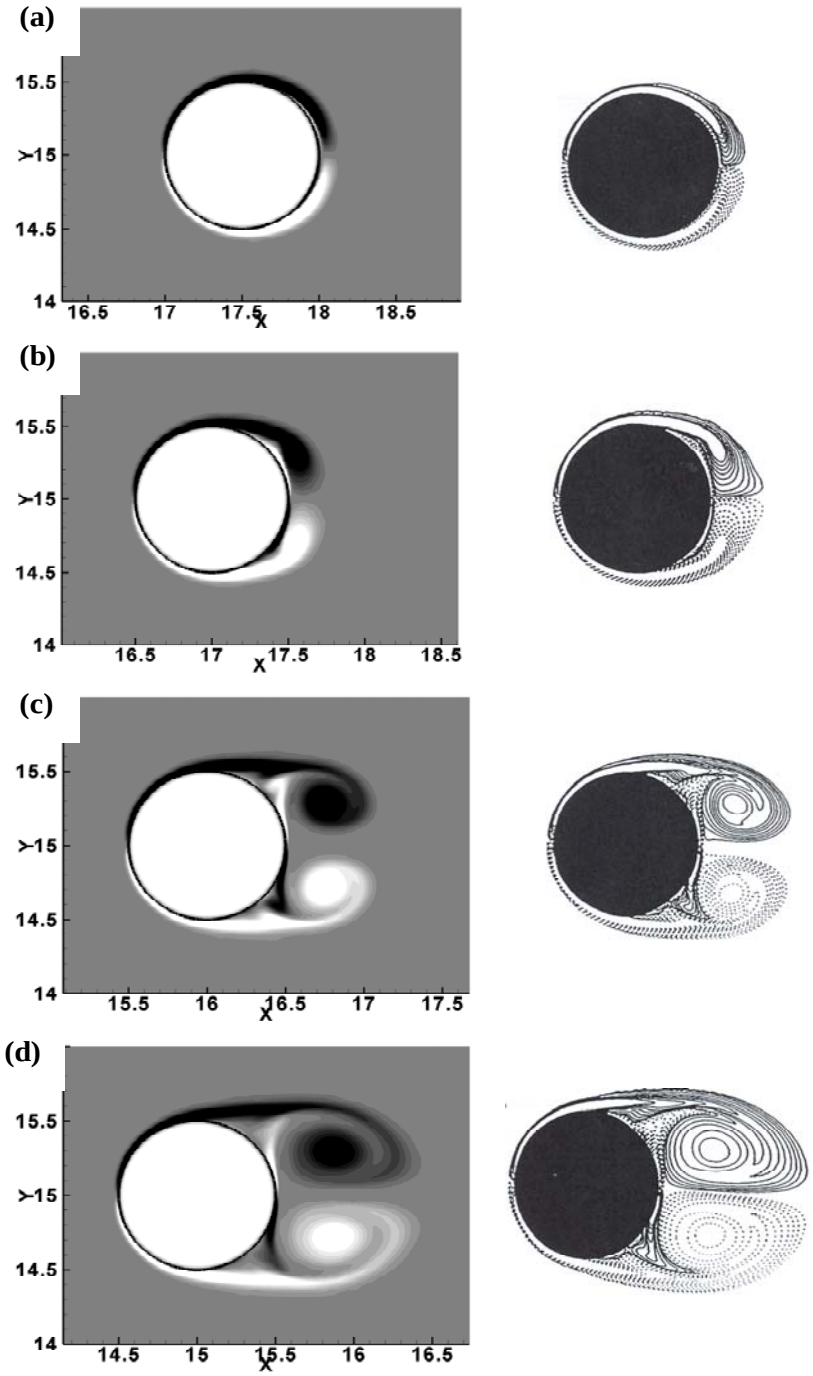


Figure 4.5. Vortex shedding from an impulsively started cylinder for Re=1000. The time instances are $T = 2.0,4.0,5.0,6.0$. These flow patterns are compared with published data [82] on the right side.

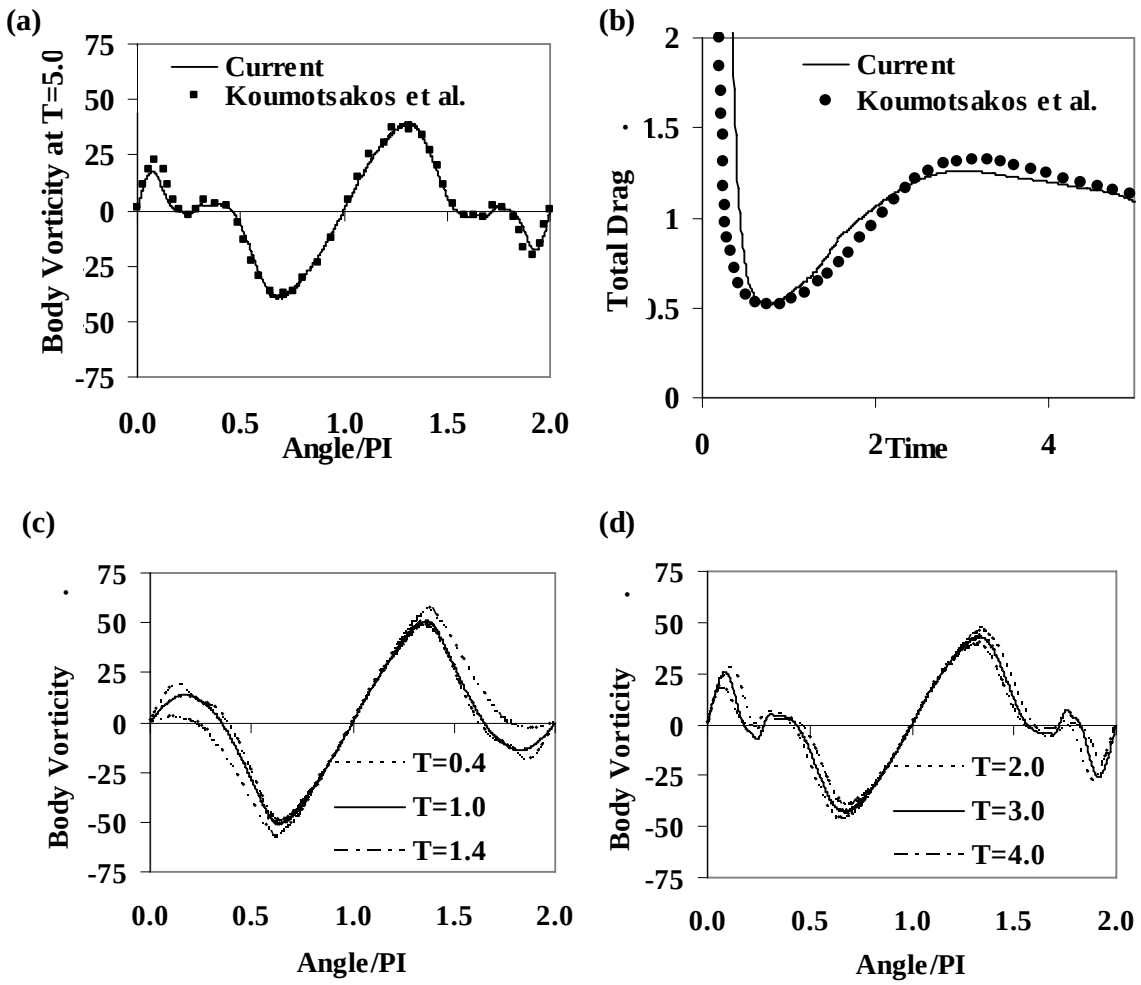


Figure 4.6. Comparison with Koumoutsakos and Leonard [82]. (a) Body vorticity from the current calculation compared with published results for Re=1000 and (b) Evolution of drag compared with published results. (c) and (d) Body vorticity at different time instances.

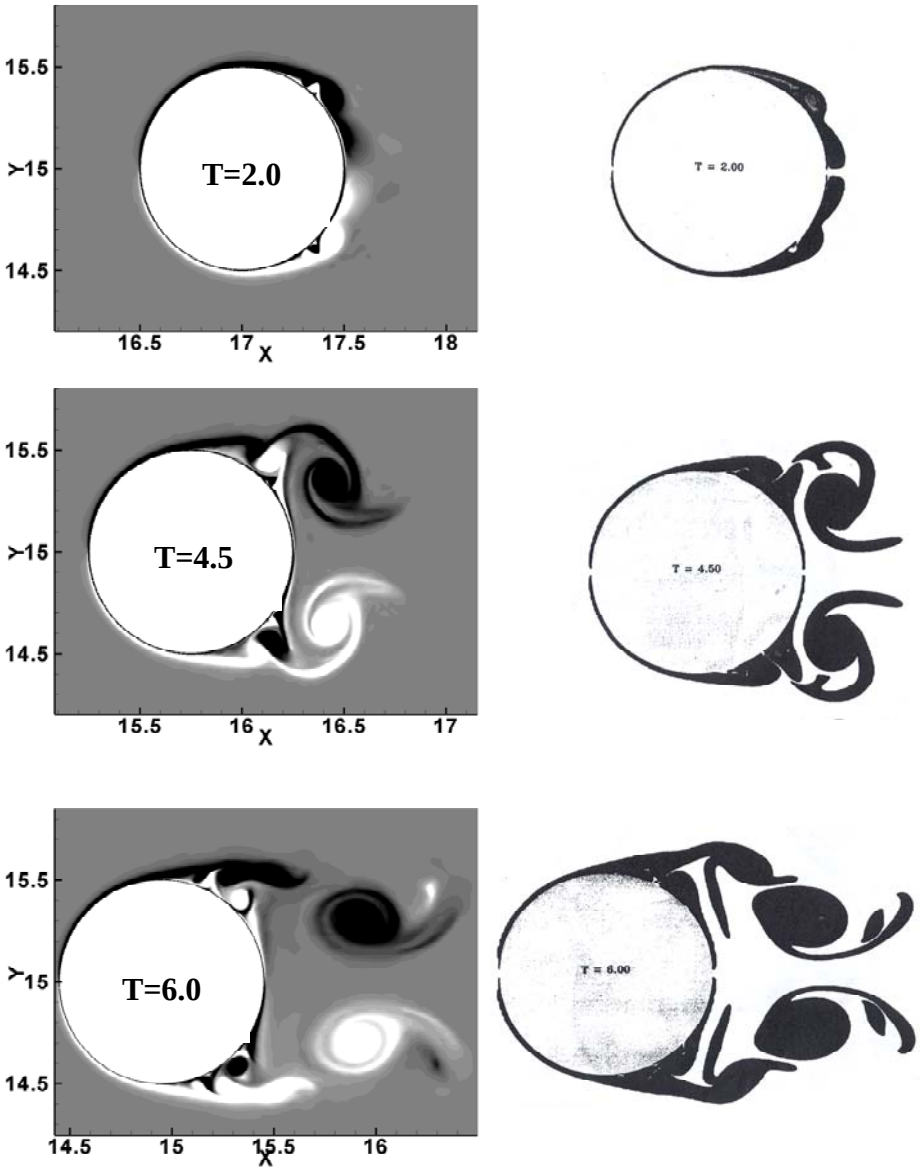


Figure 4.7. Comparison of flow patterns are different time instants with published results from [82] for Re=9500. On the right side the flow patterns at corresponding time instances from published results are shown

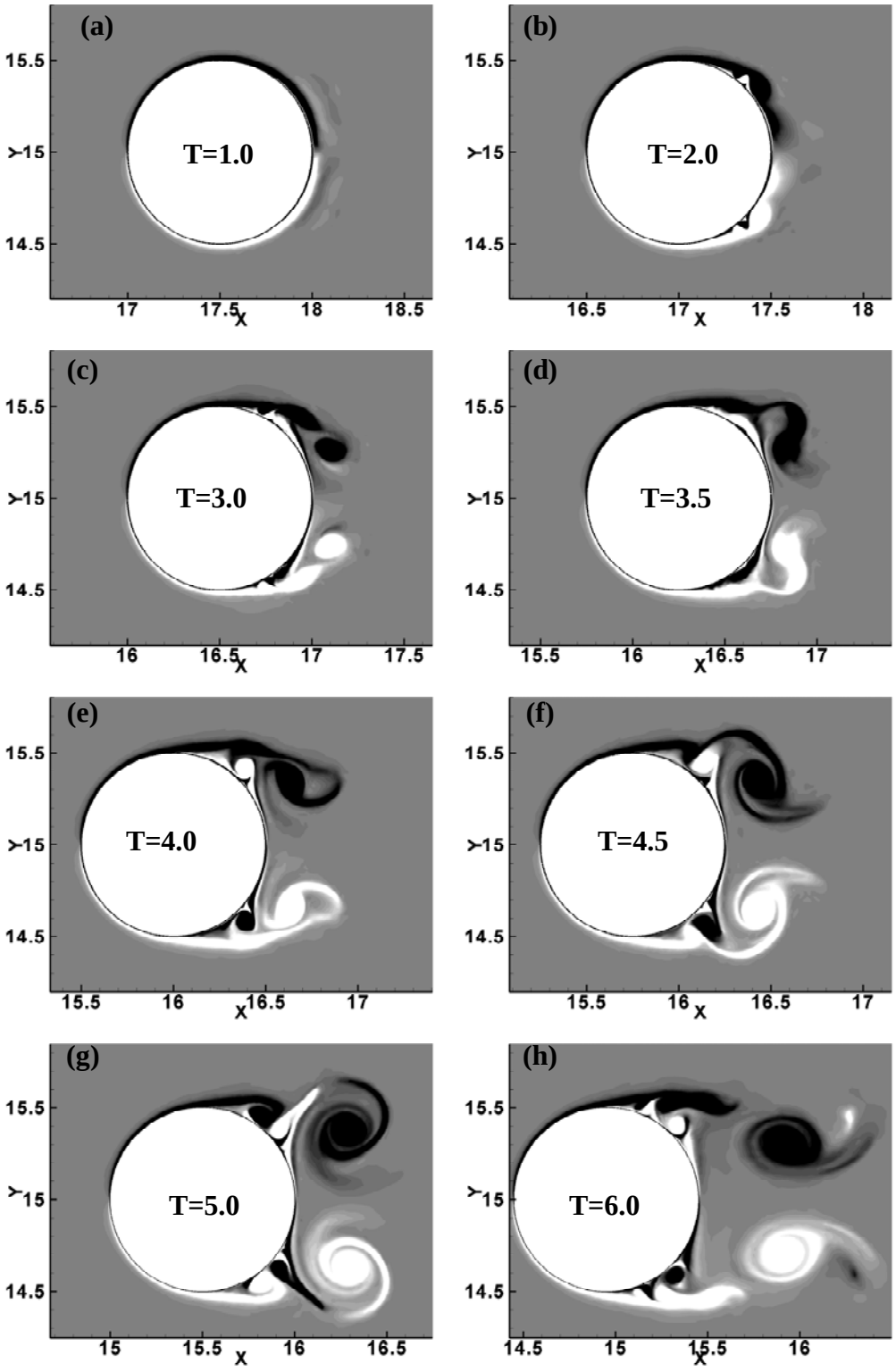


Figure 4.8. Vortex shedding at Re=9500. The flow patterns are comparable to published results [82].

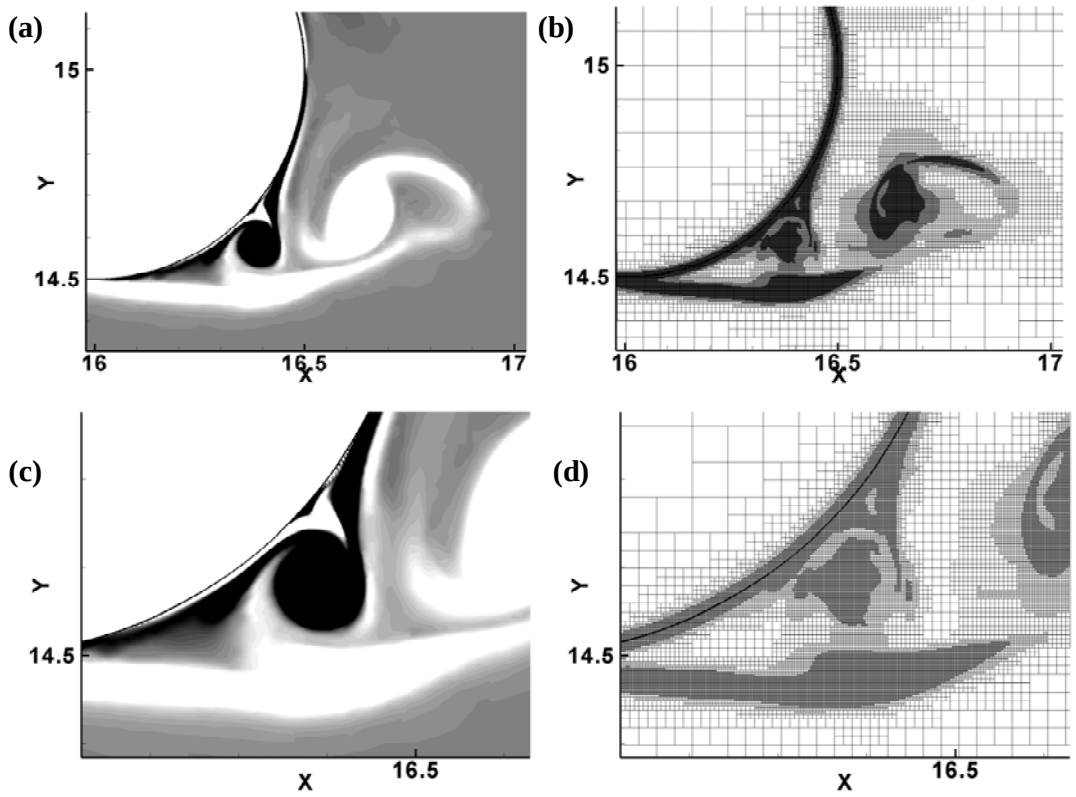


Figure 4.9. Impulsively started cylinder at Re=9500. (a) Instantaneous vorticity contours, (b) corresponding refined mesh. (c), (d) Zoomed in boundary layer and corresponding refined mesh.

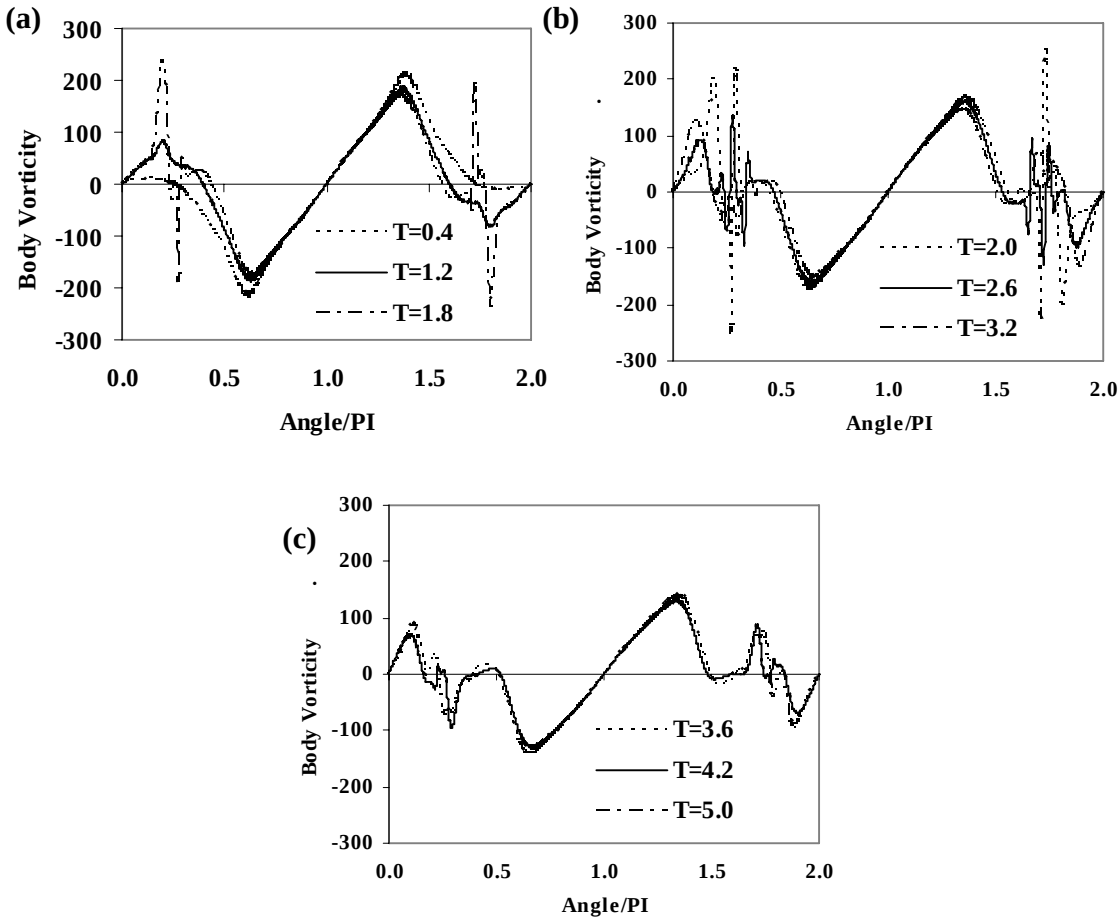


Figure 4.10. Body vorticity at different time instances for an impulsively started cylinder at Re=9500. The plots are shown at the same time instances as shown in [82].

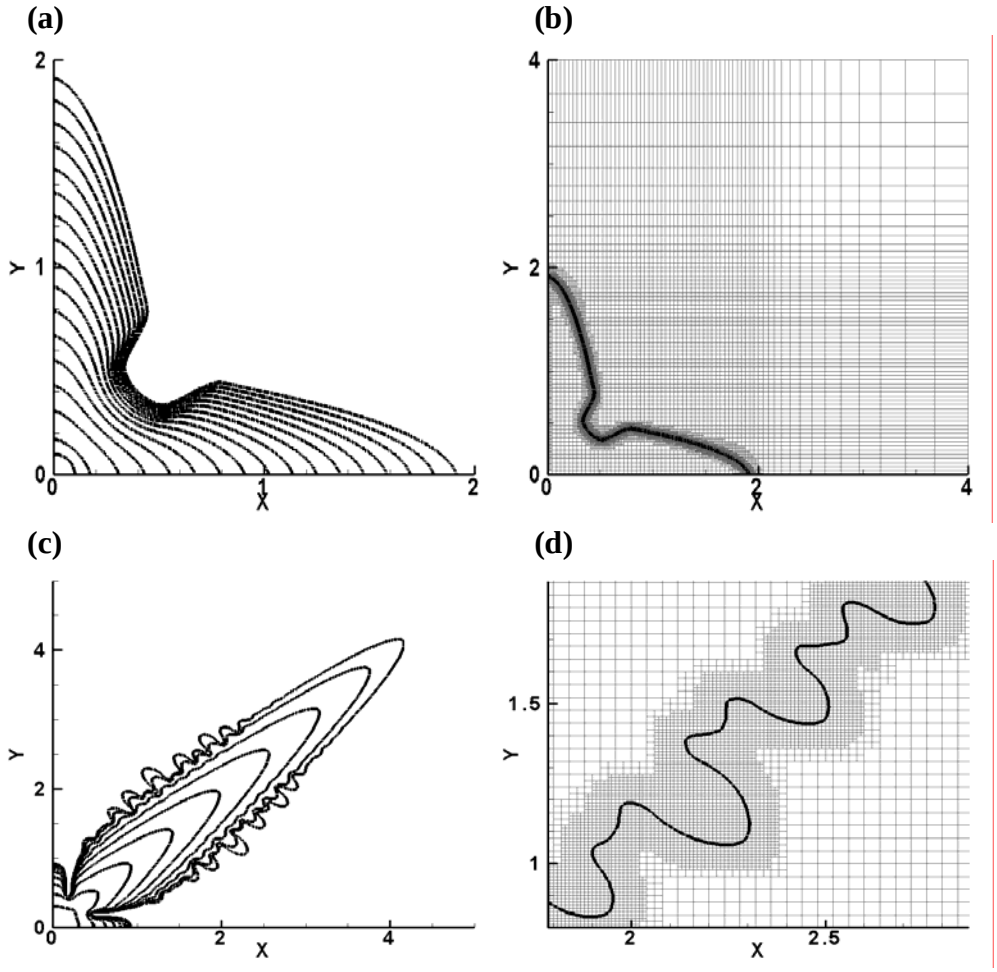


Figure 4.11. Phase change problem. (a) Dendrite growth (under-cooling $\Delta = 0.55$, capillarity parameter $d_0 = 0.5$ and anisotropy strength of $\varepsilon_0 = 0.03$) sequence. (b) Mesh used for calculation. The refined mesh tracks the interface. (c) Dendrite growth along diagonal axis. Note the secondary dendrites. (d) Refined mesh tracks the smaller scale features.

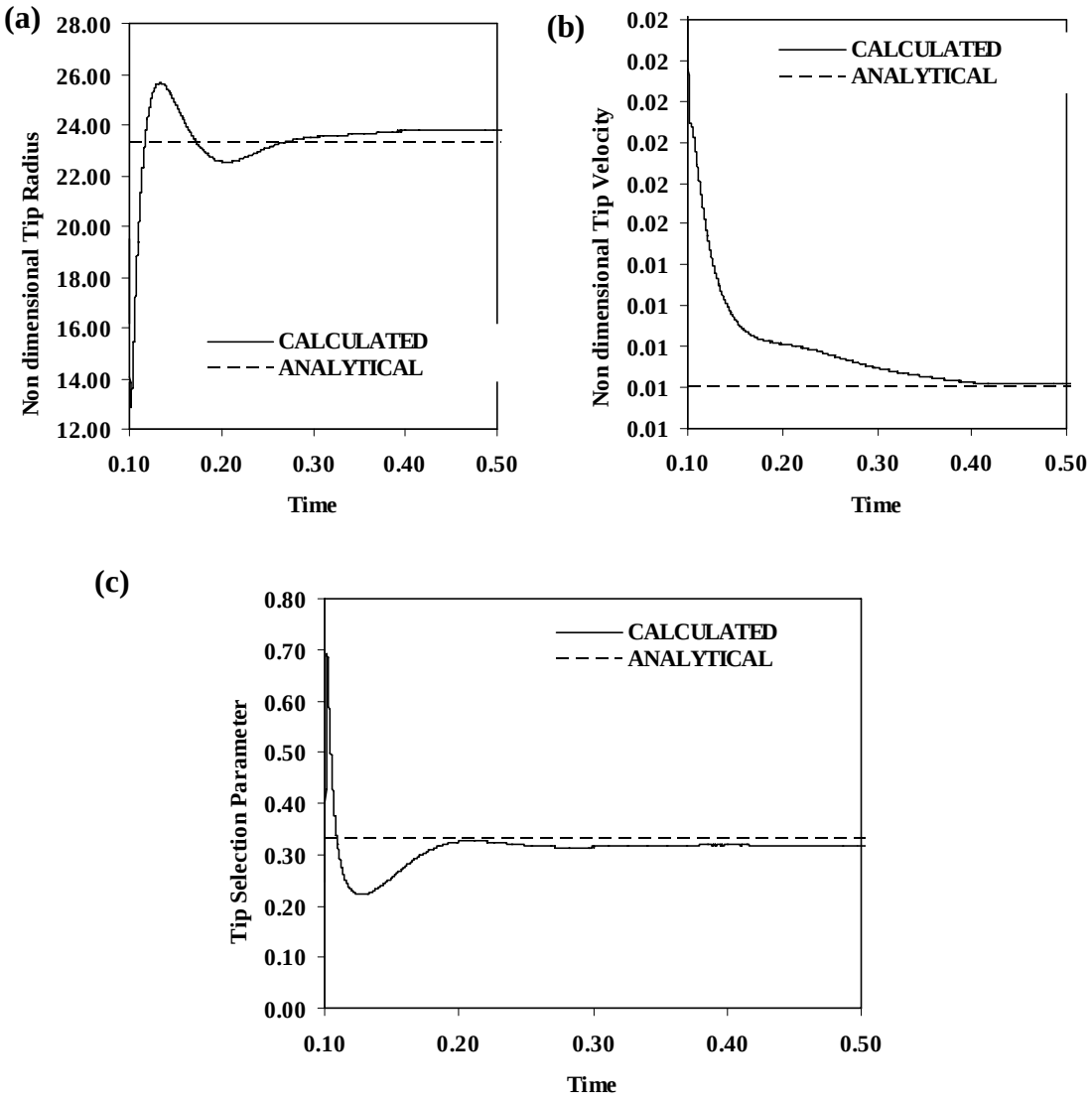


Figure 4.12. Comparison with microscopic solvability theory [135]. (a) Tip velocity and (b) Tip radius and (c) Tip selection parameter.

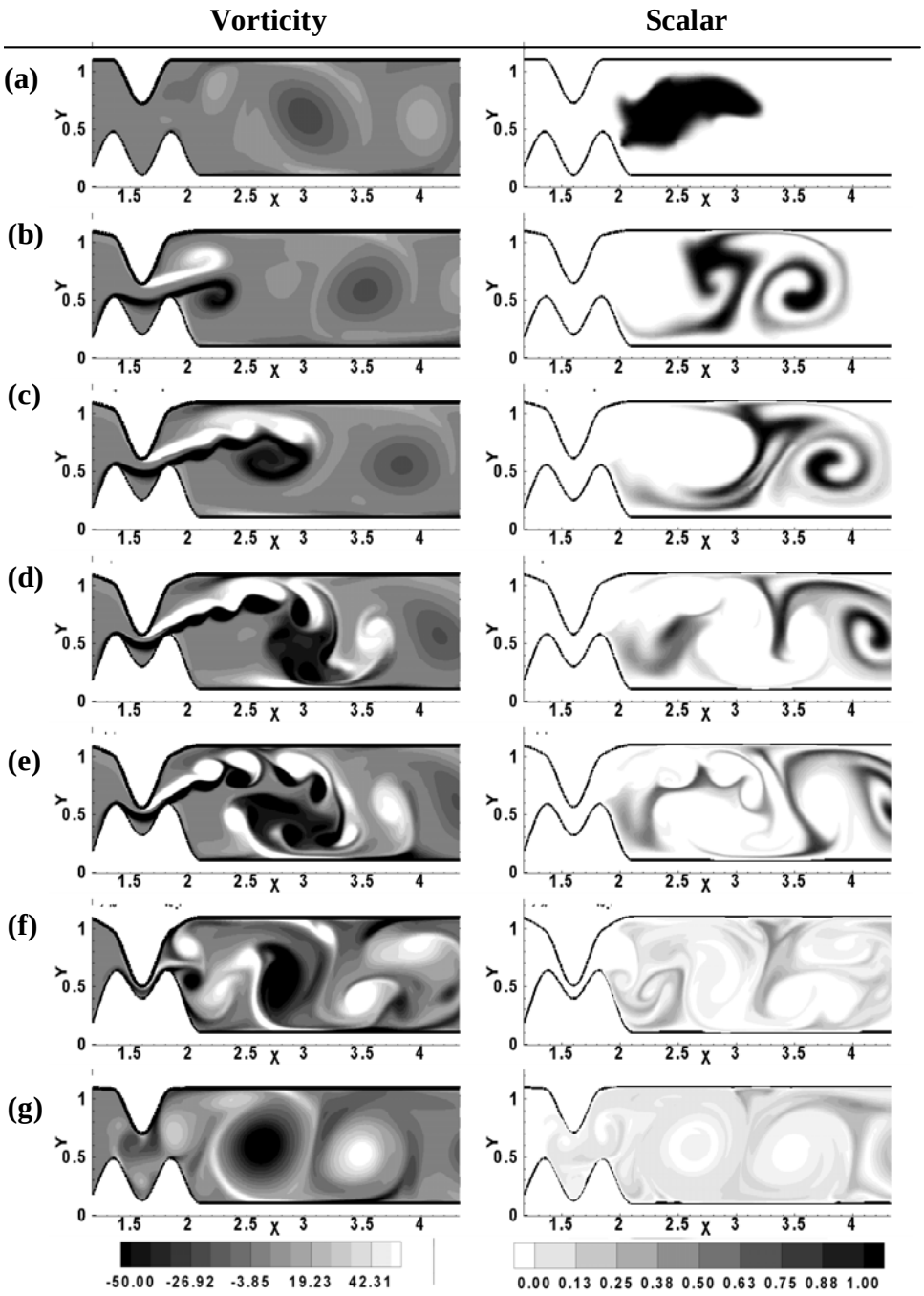


Figure 4.13. Effect of the pylorus on mixing in the GI tract. The left panels show vorticity contours at different points in the cycle while right side panels show the scalar contours indicating mixing. Note that the jet emerging from the closed pylorus notch impinges against the upper wall Interacting vortices lead to a large amount of mixing.

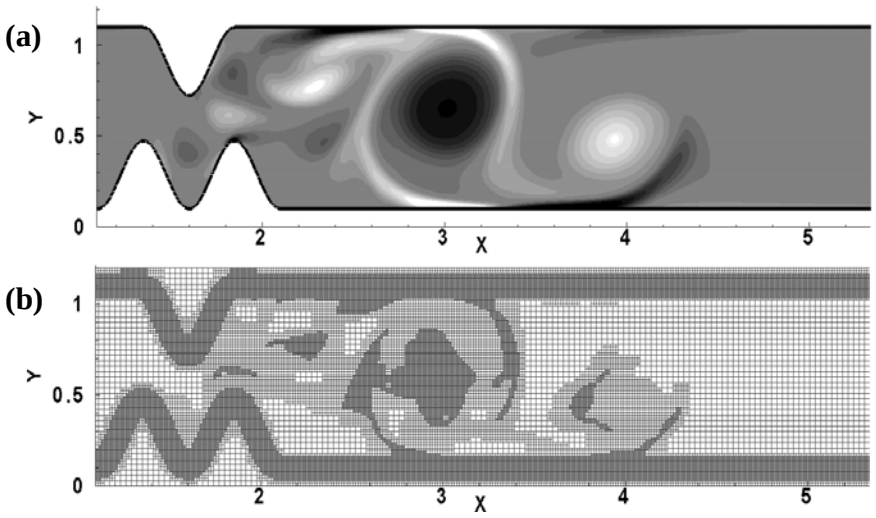


Figure 4.14. Formation of refined mesh. (a) vorticity contours and corresponding (b) refined mesh for the pylorus simulation.

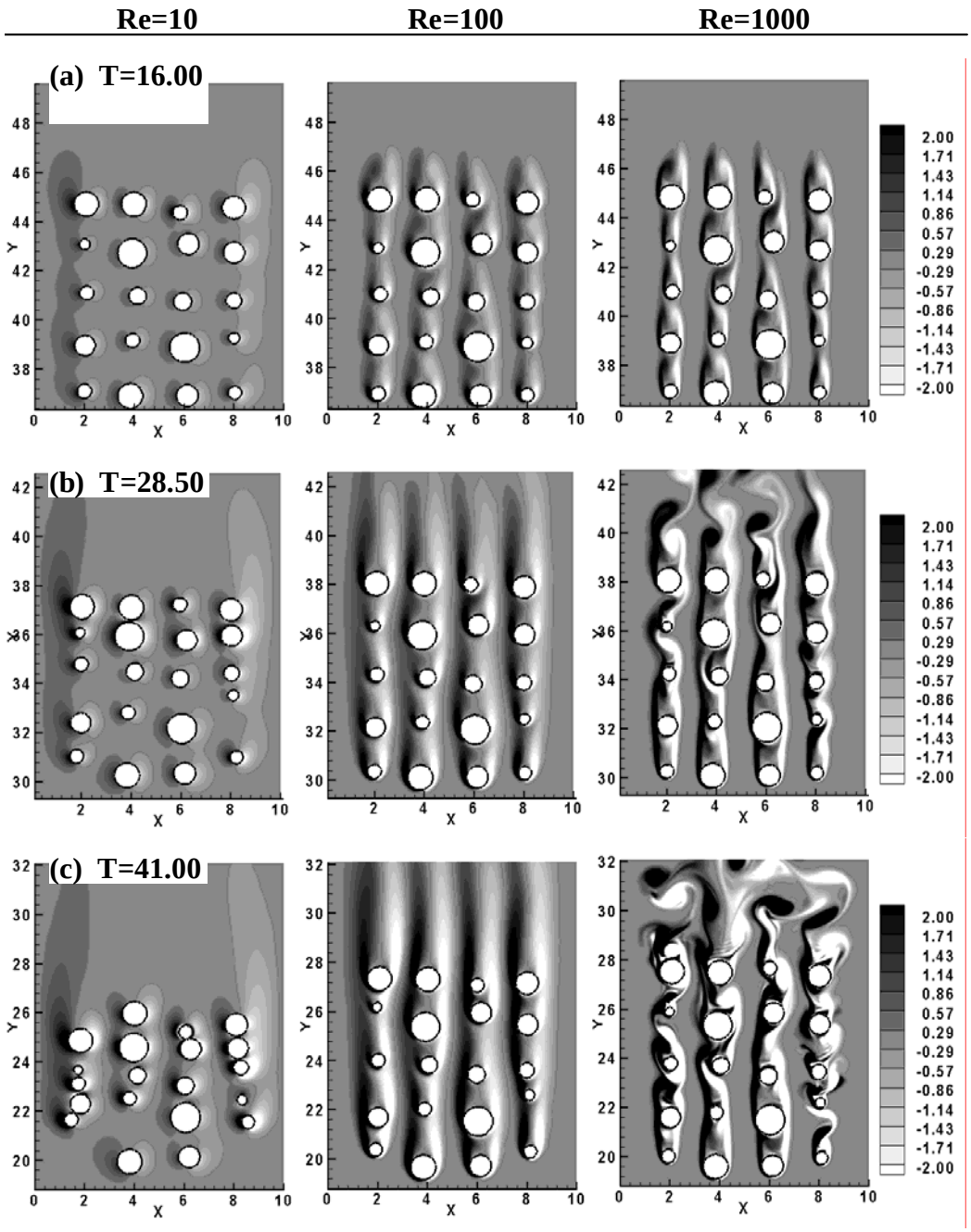


Figure 4.15. Cylinders falling under the effect of gravity. Multiple Levelsets of different scales simulated by local mesh refinement. (a), (b) and (c) show instantaneous vorticity contours at different points during free fall for the three Reynolds numbers calculated.

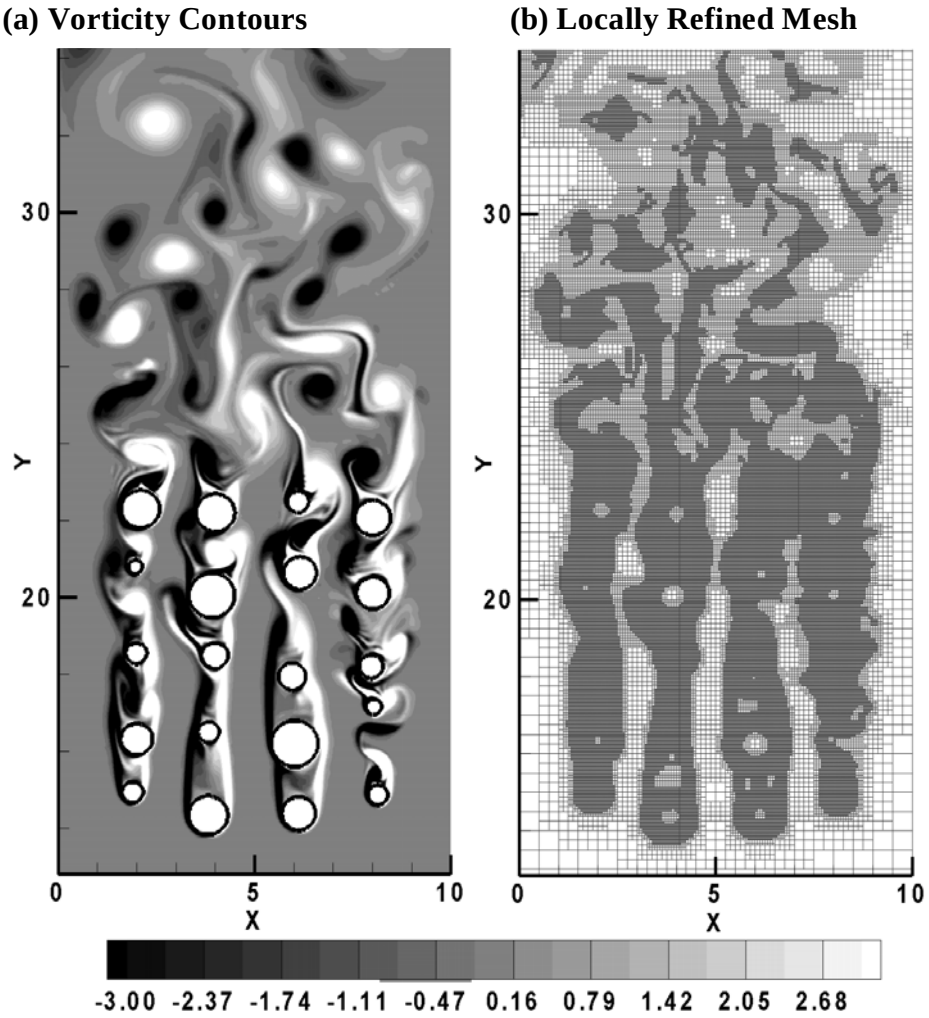


Figure 4.16. Instantaneous vorticity contours and corresponding refined mesh for freely falling cylinder at Re=1000.

CHAPTER 5

PLATELET ACTIVATION DURING HEART VALVE CLOSURE

5.1 Introduction

The method developed in Chapters 2 and 3 is now applied to perform simulations of the operation of a mechanical heart valve closure. Implantation of mechanical heart valves is one of the most common forms of treatment of valvular diseases. Due to the disturbances to the blood flow and non physiological shear stresses produced in the heart chambers due to the presence of the rigid mechanical valve, formed elements in the blood such as platelets are damaged and cause further complications in the patients. Understanding the fluid mechanics of flows through mechanical valves is thus important in design improvements. A full FSI simulation of the valve closure by the application of pressure in the ventricular chamber is demonstrated here. One significant difficulty in modeling the heart valve mechanics is the wide disparity in length scales from the large-scale motions with length scales in the valve orifice ($\sim 25\,mm$) to the leakage flow through the gap ($\sim 0.4\,mm$) to the flow in the hinge regions of mechanical valves ($\sim 100\,\mu m$). To capture the flow dynamics around the heart valve accurately, the above length scales need to be resolved adequately. The leakage flow is a significant part of the valve dynamics that can only be adequately resolved for meaningful flow analysis by employing state-of-the-art computational techniques.

Leaflet motion is another important feature that must be properly captured. The valve sweeps an angle of almost 64° in the process of valve closure. It is well established by previous studies [33, 113] that flow patterns recorded from fixed valve models are

entirely different from moving valve models. Use of body-fitted meshes may be able to adequately resolve the leakage flow by use of highly stretched meshes [15, 28, 33, 52, 53, 64, 75, 76, 80, 81, 85, 118, 119, 149, 161]. However, use of moving meshes presents additional complications of re-meshing, in particular the rather complex task of mesh generation to accommodate changes in geometry in order to prevent mesh skewness, entanglement, extreme disparity in cell sizes and aspect ratios and so on. All these factors impact negatively on the accuracy and robustness of body-fitted mesh schemes. The natural way of handling moving boundary problems by entirely obviating the complexity of mesh generation is by using a Cartesian grid approach. Cartesian grid methods have been previously employed in modeling of heart valve dynamics [108, 109] but ensuring sufficient resolution in the small gaps can make the computations extremely tedious because of the fine meshes that have to be used. A further consideration is that algorithms for solving the Navier–Stokes Equations must employ at least second-order accurate numerics both in space and time to capture the complex flow dynamics near the heart valve. First-order methods are more stable but introduce excessive artificial viscosity into the numerical solutions and tend to smear out most scales of motion, except perhaps the largest ones, and could dramatically misrepresent hemodynamically relevant flow features—such as the intensity of regions of high shear, the existence of pockets of reversed flow and flow separation, etc [163]

The valve is modeled as a smoothed rectangle hinged close to the left edge. The left boundary of the domain is symmetric giving the effect of two leaflets. The right boundary is the valve housing and is modeled as a wall. Pressure on the top boundary representing the atrium is held fixed while the pressure on the bottom boundary is

increased linearly to 120mm Hg over a period of 60 ms, representing the pressure applied by the ventricle in a heart beat. Based on maximum velocity recorded in the gap between the leaflet and the housing, and considering the gap dimensions and the properties of blood, the maximum Reynolds number is calculated to be around 1800. The pressure gradient across the valve leaflet induces blood flow, and causes the leaflet to swing shut due to the hemodynamic forces acting on the valve.

5.2 Fluid-Structure Interaction

The leaflet rotation required as input for solving the Levelset equation as discussed in Section 2.2.3 can be described by the relationship [28, 29, 85],

$$\frac{d^2\theta}{dt^2} = \frac{M}{I_o} \tag{5.1}$$

In the equation, $\theta(t)$ is the opening angle, indicating the leaflet position at any instant t; I_o is the moment of inertia of the leaflet about the pivot ($3.3e^{-9} kg-m^2$), M is the total momentum applied on the leaflet from the external forces inducing the leaflet motion. The external momentum can be calculated as:

$$M = M_G + M_P + M_F \tag{5.2}$$

M_G is the momentum resulting from the buoyancy and the gravitational force and is given by:

$$M_G = gV\left(\rho_l - \rho_f\right)\left(\frac{l}{2} - a\right)\cos(\theta) \tag{5.3}$$

Here g is the acceleration due to gravity, V is the leaflet volume, ρ_l (2000 kg/m^3) and ρ_f (1056 kg/m^3) are the leaflet and fluid densities respectively, l (1.316 cm) and a are the leaflet radius and pivot length (distance between the hinge location and the left-side edge of the valve, 0.179 cm). The leaflet thickness is 0.0899 cm. M_P is the momentum resulting from the blood pressure, and M_F is the momentum resulting from shear forces. These quantities are calculated from the flow field by integrating the normal (pressure) and tangential (shear-stresses) fluid forces acting on the valve surface.

5.3 Leaflet Rebound

The leaflet will impact against the valve seating lip at the instant of valve closure. After impact, the leaflet will bounce back from the housing. The governing equation of leaflet dynamics during impact can be expressed as [28]:

$$\omega_2 = -\sigma\omega_1 \tag{5.4}$$

where σ is the coefficient of resilience that depends upon the material of the leaflet and the valve housing. ω_1 and ω_2 are the angular velocities before and after impact, respectively. The coefficient of resilience σ is specified as 0.5 [28].

5.4 Platelet Activation

Exposure to high shear and entrapment in re-circulation regions with high residence time in the vicinity of the valve are expected to contribute to platelet deposition and thrombus formation [18, 85]. A dynamic platelet activation model that includes the

effects of shear stress and exposure time is employed. Statistical observations of activation levels of platelets at various stages of valve closure are made throughout the period of calculation.

The primary cause of platelet activation is exposure to high shear stress over a sufficiently extended period of time. Therefore, a primary measure for level of platelet activation can be defined as [18]:

$$\alpha_T = \int_0^T |\tau| dt \tag{5.5}$$

5.5 Simulation Conditions

The geometry for a typical bi-leaflet valve is shown in Figure 5.1(a). Figure 5.1(b) shows the valve leaflet dimensions and Figure 5.1(c) shows the pressure applied on the ventricular side. In the fully opened position, the valve is aligned at 0.2° with the vertical axis. To close completely, it swings through an angle of 63.8° with the fully closed position being aligned at 64° with the vertical axis. Experiments record the closure time for this valve to be around $32\,ms$. The large angle of rotation does not provide any challenge to the Eulerian method since the mesh does not have to be modified to account for the rotation. However, the challenge lies in adequately resolving the small gaps between the two leaflets and the leaflet and the wall. The gap between the two leaflets as well as between the leaflet and the wall is specified as $0.03\ cm$ while the overall width of the valve itself is $1.3116\,cm$.

The fluid is assumed to be incompressible, laminar, and Newtonian with the density of 1056 kg/m^3 and viscosity of 0.0035 $kgm^{-1}s^{-1}$, which is representative of

human blood properties at $37\,^{\circ}C$. The ventricular pressure at the inlet is raised from 0 to 120 $mmHg$ at a constant pressure rise rate of 2000 $mmHg/s$ as described in Figure 5.1(c) and the atrial pressure at the outlet is maintained at 0 $mmHg$ consistent with physiological conditions. The ventricular and atrial pressures prescribed above are sufficient to be used as the boundary conditions at open boundaries. At the symmetry boundary, normal velocity component is set to zero and all other quantities are extrapolated assuming zero normal gradient. The remaining boundaries are solid walls, and the standard no-slip condition is used. Note that the leaflet rotation is calculated from the fluid forces acting on it. Platelets are assumed to be point spheres of radius $2\,\mu m$ and density same as that of blood.

5.6 Results

The nominal local Reynolds number based on properties of blood is calculated to be around 1800 based on the gap dimensions and the maximum velocity computed in the gap after closure. The above Reynolds number indicates that the flow through the valve is expected to be laminar for the bulk of the flow. However, the disparate length scales (i.e. between the overall valve dimensions and the leakage region) pose a challenge in adequately meshing the leakage gap (~15 cells at least are required to resolve the jet flows in these gaps) to obtain reliable solutions in reasonable time making optimal use of computational resources. The local mesh refinement algorithm solves this problem and refines meshes only in regions that need to resolve high solution gradients or curvature.

Figure 5.2(a) shows the mesh being locally refined in regions where the algorithm for adaptive refinement detects high velocity gradients and gradients of vorticity. The left

panel in the figure shows the refined mesh and the right panel shows the vorticity contours. The maximum allowed refinement level for a fixed base mesh is estimated from a grid refinement study tracking the closure dynamics. Figure 5.2(b) shows the closure angle recorded with time for 3,4,5 and 6 maximum levels of refinement allowed in the calculation. A pilot study with 3 levels of refinement was seen to be unstable beyond a certain point of valve closure, presumably because the mesh was too coarse for adequate resolution of the large gradients present in the boundary and shear layers in the gap region. As seen in the figure, refinement of the base mesh up to 5 levels of refinement shows converged solution of closure dynamics and is adopted in the present calculation.

5.6.1 Closure Phase

In the fully open position, the valve leaflet is aligned at an angle of 0.2° with the vertical axis. Initial closure of the valve is very slow as the open valve does not present much obstruction to flow. In this position, the forces acting on the valve are relatively low because its orientation is in the direction of flow. As the valve swings shut pressure buildup on the upstream (ventricular) side causes the net force acting on the valve to increase, thus increasing the angular velocity of the valve. The net effect is that the leaflet motion is very small for a considerable time after the ventricular pressure rise is initiated; in the final 25% of the time there is exponential increase of angular velocity and corresponding rapid closure of the valve. Figure 5.3(a) compares the experimentally measured closure behavior with that of the current calculation. Closure time measured from the current calculation is about 30.6 ms while the experimental curve shows about 32 ms. The discrepancy could be because the computational model is restricted to 2D

and does not incorporate the effect of 3D flow from which experimental data was obtained. Figure 5.3(b) shows the corresponding applied pressure on the ventricular side. (The points marked on Figure 5.3(b-c) correspond to the time instants at which the contour plots are presented in Figure 5.4(a-f). Figure 5.3(c) shows the angle through which the valve has moved over time while Figure 5.3(d) shows the angular velocity. Once the valve hits the housing it rebounds with a resilience factor of 0.5 characterized by the reversal of the angular velocity (Figure 5.3(d)) and re-opening of the valve. In the following results, the valve leaflet edge at the centerline will be referred to as the "left edge" and that on the housing side will be termed the "right edge". Note that these terms apply specifically to the present case of two-dimensional simulations with a symmetry line at the left boundary.

Figure 5.4 shows vorticity contours at various stages of leaflet closure. In Figure 5.4(a), the valve is almost fully open. Even at this stage, boundary layer separation is evident at the left and right edge of the leaflet. The next stage in Figure 5.4(b) shows the wake starting to oscillate as it becomes unstable. In the third stage in Figure 5.4(c) the unstable wake rolls up into vortices which shed periodically from the leaflet at the left edge. At this point in time the gap between the two leaflets at the left side (i.e. the centerline) is the smaller one and the flow velocity is higher there, causing vortex shedding from the left edge of the leaflet. This periodic shedding of vortices occurs at the left edge throughout the valve closure process but this appears to have little consequence for platelet activation or thrombus formation as discussed later.

The dynamics at the right edge of the leaflet are more interesting and consequential for platelet activation and localization. As seen in Figure 5.4(a-c) the

boundary layer on the right edge of the leaflet starts thickening and separating from the edge. A weak vortex with recirculating flow is observed on the distal side of the leaflet at the right edge as well. The boundary layer on the (Figure 5.4 (d)) wall is clearly distinguishable at this stage. As the valve closes further, the anti-clockwise vortex on the right edge of the leaflet starts thickening and separating. But, this boundary layer never appears to roll up and shed as it does on the left edge of the leaflet throughout the closure phase. This is because the right edge of the leaflet has a significant velocity in the same direction as the flow due to leaflet rotation around the hinge. Also, as long as the gap between the right edge and the housing is large the predominant flow velocities at this stage of valve closure lie at the centerline of the valve; this leads to higher shear rates and hence enhanced shear layer instability and roll-up at the left edge of the leaflet. The leaflet rotation has the effect of pushing the fluid distal to the valve towards the central axis such that the jet flow from the gap between the leaflet and the valve is pushed towards the central axis. In addition, the leakage jet and shear layer emerge into a region of increased flow area, leading to a further deviation of the jet into the space distal to the valve. Because of the orientation of the leakage jet, the wall boundary layer on the valve housing thickens and separates. In Figure 5.4 (f) which is very close to the closure point, the positive vorticity from the valve edge and the negative vorticity from the wall approach each other closely. If held in this position for a longer time, there is the possibility of interaction between the boundary layers as shown in the stages after closure. To summarize, the closure stages are characterized by boundary layer separation from the right edge of the leaflet, instability and roll up of the wake, thickening of boundary layers on the right edge of the leaflet and the valve housing, and drawing in of

the housing wall boundary layer by the vortex emanating from the leaflet's right edge. The interaction of the shear layer at the right edge of the wall with the boundary layer on the valve housing has been well resolved in the current calculations using the locally refined mesh in this region. This interaction proves to have significant impact on the platelet activation as described in the results below.

The vortex roll-up from the valve edges has been reported by several previous computational and experimental studies [16-18, 95, 113] and has been implicated as one the main reasons for platelet activation, thrombo-emboli formation and cavitation effects that are detrimental to the valve performance. The small gaps between the valve leaflets and between the right edge of the leaflet and the wall after valve closure are regions of high speed jet flow. Figure 5.5 shows the flow characteristics in this region at valve closure. Figure 5.5 (a) shows the vorticity contours at the closure instant. Pressure contours in Figure 5.5 (b) shows the pressure build-up on the upstream (ventricular) portion of the gap. Figure 5.5 (c) and (d) show the horizontal and vertical velocity contours. The peak velocity at this instant is around 20m/s. As seen in Figure 5.5 (e) which shows the stream lines, the jet between the valve leaflets, i.e. on the symmetry axis, is oriented in vertical direction with periodic shedding of vortices while the jet between the housing wall and the leaflet (on the right edge) is oriented towards the central orifice. The orientation of this jet is caused by a combination of factors, including the swinging movement of the valve during closure phase which pushes the fluid towards the central axis and the sudden area expansion experienced by the leakage flow. Additionally, the central jet is stronger for most of the closure stage because of the smaller orifice area and this jet draws in most of the fluid towards it. The orientation of

the leakage jet towards the central orifice is critical because as seen in Figure 5.5 it has a significant "pulling" effect on the boundary layer on the housing wall, causing it to thicken and separate. These two oppositely oriented vorticity layers on the right edge of the valve are critical in the behavior of the jet in the post-closure stage. The jet orientation will not be captured in its entirety unless the swinging action of the valve is simulated. Therefore, as noted by previous researchers, flow characteristics will be entirely different for a moving valve in comparison to a fixed valve [33, 113].

Figure 5.6(a) and (b) show the velocity and shear stress magnitude in the small gaps through the valve closure and rebound stages respectively. As seen, the peak velocity in the gaps until the valve rebounds is around $20\,m/s$ in agreement with results from previous computational work by the present group and others [28, 29, 85]. However, at the first valve rebound, velocity magnitudes of $120\,m/s$ were computed at the leaflet tip. During this period of rebound the energy which was previously responsible for valve rotation is now diverted into pushing the fluid through the small gap. With the high velocities built up the flow results in very large shear stresses in the gap flow during this period. As seen in Figure 5.6 (b), absolute shear stress reaches high values at this instant, much higher than the shear levels required for platelets to become activated. After the instant of rebound, the velocity normalizes to around $20\,m/s$ again. Due to the high resolution of the mesh enabled by the local mesh refinement, intricate flow features are captured adequately in spite of high velocity magnitudes.

5.6.2 Effect of Flow During Closure and Rebound Phases

Figure 5.7 shows vorticity contours, absolute shear stress and the simulated platelet activation in the region of the gap between the leaflet and the housing at time

instants near the valve closure and rebound phases. The most significant feature observed in the figure is the interaction between the boundary layers separating from the wall and the leaflet edge. As seen in Figure 5.7 (I-a), the boundary layers from the wall and the leaflet edge separate and come closer. Due to the mutual interaction of these vortex sheets of opposite sign, secondary vortex layers are formed below the main vortex structures, both on the wall and on the valve. Instability of these vortex sheets causes the sheets to roll up, thus leading to a self-sustaining regeneration of pockets of vorticity that lead to local recirculating flows in the region distal to the valve. However, the interaction of the vortices in this region does not allow vortices to shed periodically and to be carried away downstream as in the central jet on the left edge of the valve. Figure 5.7 and Figure 5.8 show the various stages of vortex interaction in the valve closure and rebound stages with the corresponding instantaneous shear stress and activation parameter. Again, in terms of energy budget, the pressure differential across the valve at this time results in the creation of recirculating fluid pockets rather than a strong jet with a vertically directed velocity field. The leaflet edge separated boundary layer vorticity draws in the vorticity from the housing wall and the two structures roll around each other. As the flow evolves, these pockets of vorticity break up but remain in the same region. The dynamics of flow in this region causes shear stresses to be high and also causes recirculations that are localized in the region near the gap distal to the valve. This high shear coupled with high residence time makes this a potential site for platelet activation and deposition. Corresponding contours of the activation parameter indicate the distal (atrial) side of the right edge of the leaflet as a high risk region for platelet activation and deposition. This is because particles entrained in the vortex interaction zone are subject to high shears

from the circulating flow, but are unable to escape quickly from the region. It may be noted that if the interaction between vorticity generated at the housing wall and valve edge does not occur, the particles are likely to get activated by the high shear region in the small gap but may rapidly advect out of the housing region with the leakage jet and would then be unlikely to cause significant deposition. The resolution of the flow in the gap is therefore of crucial importance. Elucidation of the flow in this region is possible due to the high local resolution provided by the present numerical technique and has not been demonstrated previously. Eventually the strong vortex structures with high shear rates are carried downstream by the flow, away from the valve leaflet. However, weak vortex structures with re-circulating flow on the downstream edge of the leaflet persist even after 5-10 ms after valve closure. Figure 5.7 and Figure 5.8 shows the flow characteristics and corresponding shear rates and activation parameter a few milliseconds after the initial impact of the valve against the housing. At this stage, though the high shear region has moved downstream with the vortex structure, the downstream edge of the valve continues to show significant activation levels of platelets.

Figure 5.9 shows a statistical quantification of the maximum shear stress that platelets have been exposed to at various stages of valve closure. The maximum shear stress that the platelets are exposed to is plotted on the x-axis and the y-axis shows the percentage of platelets that were seeded in the domain that became exposed to that shear stress at different points of valve closure and rebound. The different curves indicate the position of the valve. As seen in the Figure, in the final 10° of valve closure there is a sudden increase in the shear stresses that platelets are exposed to and also in the percentage of platelets exposed to that shear stress. One explanation for this is the

exponential increase in the angular velocity of the valve in the last stages of closure. In the last 10° of closure, there is also a significant percentage of platelet exposed to shear stresses of $500\,Pa$ and higher. After the rebound stage, there is a small percentage of platelets that have been exposed to over $1000\,Pa$.

The periodically shedding vortices in the central orifice are likely regions of free emboli formation as reported by other researchers [16, 17] but are not likely to cause deposition on the valve itself. But interaction of vorticity from the housing wall and valve develops a potential thrombogenic region on the valve leaflet edge and housing. The high vorticity strength in the recirculating flow region also has significance as potential sites for cavitation because of highly negative pressures that develop at the vortex core [8, 29, 85].

5.7 Discussion and Conclusions

The present highly resolved calculations on a fixed Cartesian mesh show flow details in the leakage flow that have been hitherto unobserved in numerical simulations. To capture the details of the local fluid dynamics during valve function, it is essential to adequately resolve macro-scale and micro-scale dynamics simultaneously. Highly resolved 2D calculations of flow valve closure dynamics as presented in this work allows focus on the leakage flow between the valve leaflet and the housing. Platelet dynamics is tracked through the flow by a Lagrangian particle tracking method. A preliminary shear stress based platelet activation model is included to predict likely regions of platelet activation and thrombus formation and an attempt has been made to explain the vulnerability of valve housing to thrombus formation.

The interaction between separated boundary layers from the valve housing and the valve leaflet is mainly responsible for complex flow system observed just downstream of the gap between the wall and the valve leaflet in the near-closure and rebound stage. The flow in this region is highly dynamic with high stresses due to vortex stretching, breakup and roll-up. Platelets caught in this system are trapped and stay in the vicinity of the valve leaflet and housing till the vortices are diffused or advected away from the region. High shear stresses and high residence time in this region make this region highly vulnerable to thrombus formation. The high strength vortices also make this a potential region for cavitation [8]. The gap between the two leaflets on the other hand shows periodic vortex shedding. Though platelets passing through this gap are likely to experience high stresses and aggregate forming thromboemboli [16], thrombus deposition is unlikely in this region. The difference in behavior of the two gaps is partly due to the orientation of the leaflet and the intense movement in the flow direction experienced by the right edge of the leaflet and the relatively mild movement by the left edge of the leaflet opposing the flow direction. On the right edge of the leaflet, the flow is oriented towards the central axis of the valve and is responsible for drawing in the housing wall boundary layer and causing it to separate from the wall. The resulting interaction of the oppositely signed vortex sheets leads to creation of intense, yet localized pockets of vorticity with high residence times in the region distal to the valve near the leakage gap. The combination of high shear stress and locally recirculating flows can have rich significance for platelet activation and deposition.

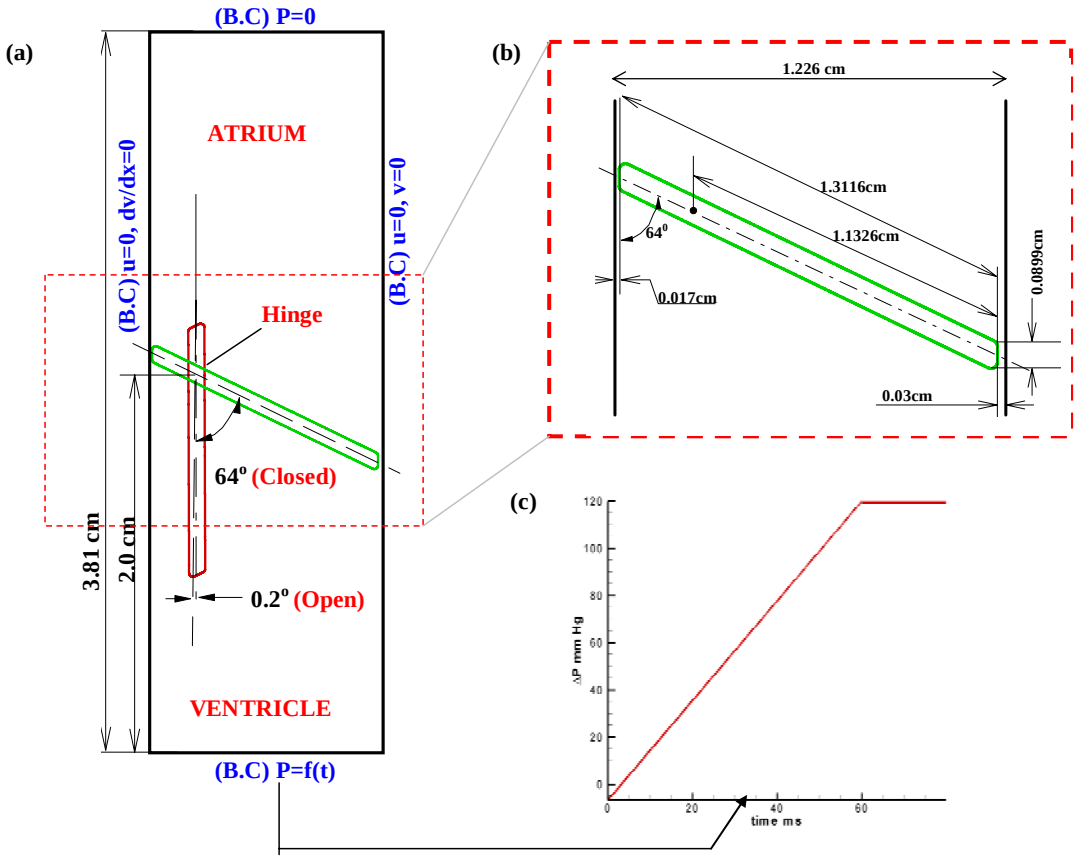


Figure 5.1. Schematic of numerical model. (a) Full view of domain with dimensions and valve in fully closed and open positions. Also indicated are the applied boundary conditions. (b) Zoomed view of the area marked in (a) with typical valve dimensions. (c) Profile of the time varying pressure applied at the ventricular side.

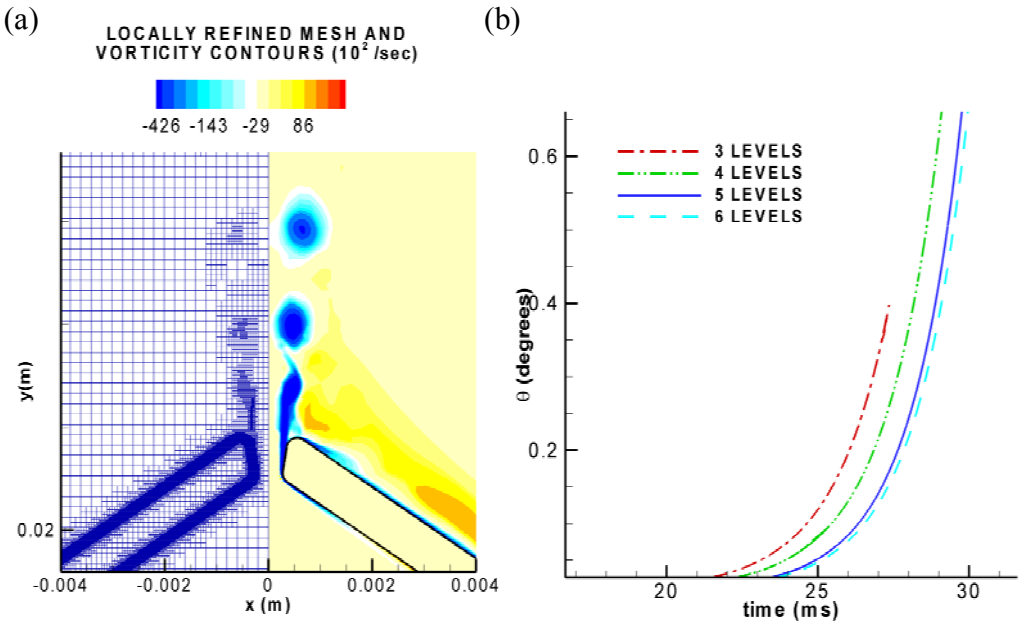


Figure 5.2. Deciding on an optimum mesh. (a) Locally refined Cartesian mesh. Note that the mesh is refined in areas of high velocity gradients. Mesh refinement enables optimum resolution of boundary layers and shear layers. (b) Grid refinement study deciding maximum number of refinement levels allowed for optimizing computation time and accuracy.

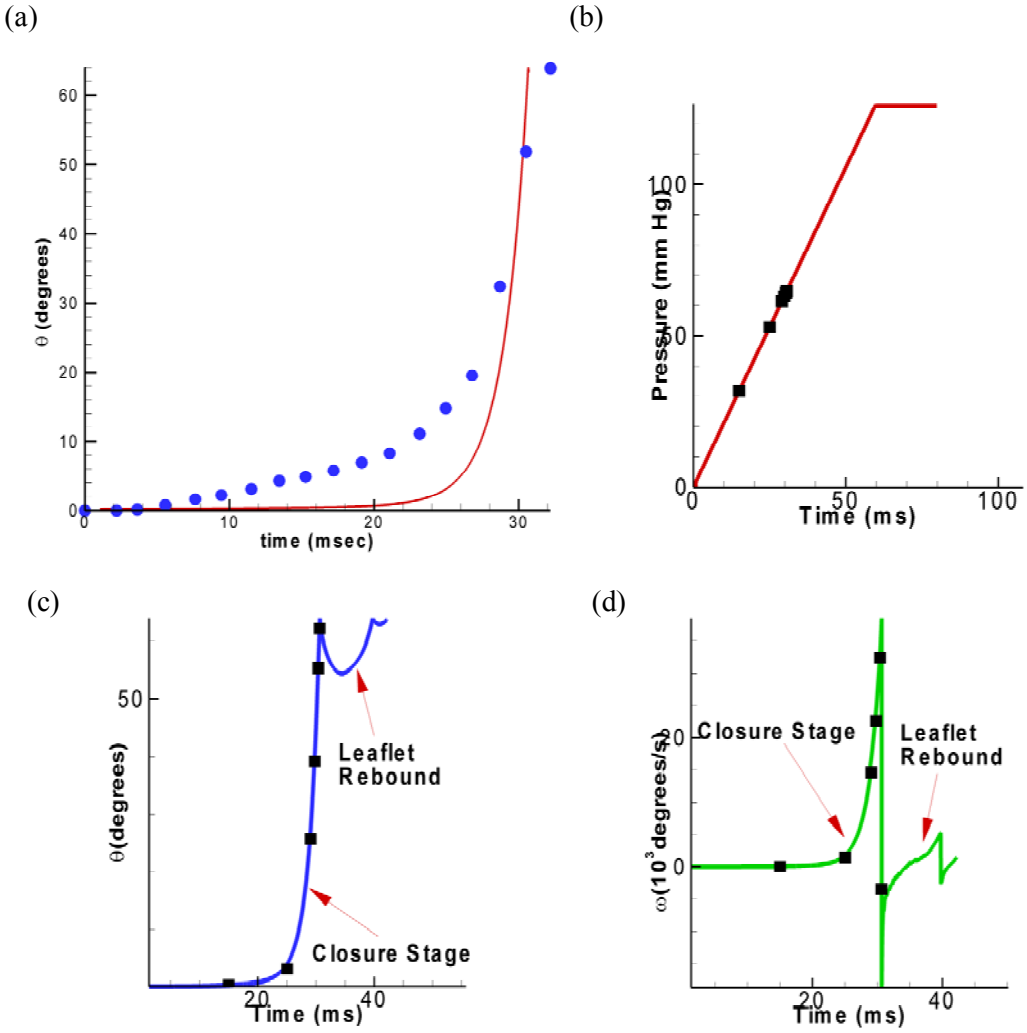


Figure 5.3. Quantitative analysis. (a) Comparison of valve closure stages from calculation with experimentally measured values. (b) Applied pressure profile on the ventricular side. (c) Angle made by valve with vertical axis. (d) Angular velocity of the valve as the valve swings shut. The points marked in (b)-(d) are the stages of valve closure at which vorticity contours are plotted in Figure 5.4.

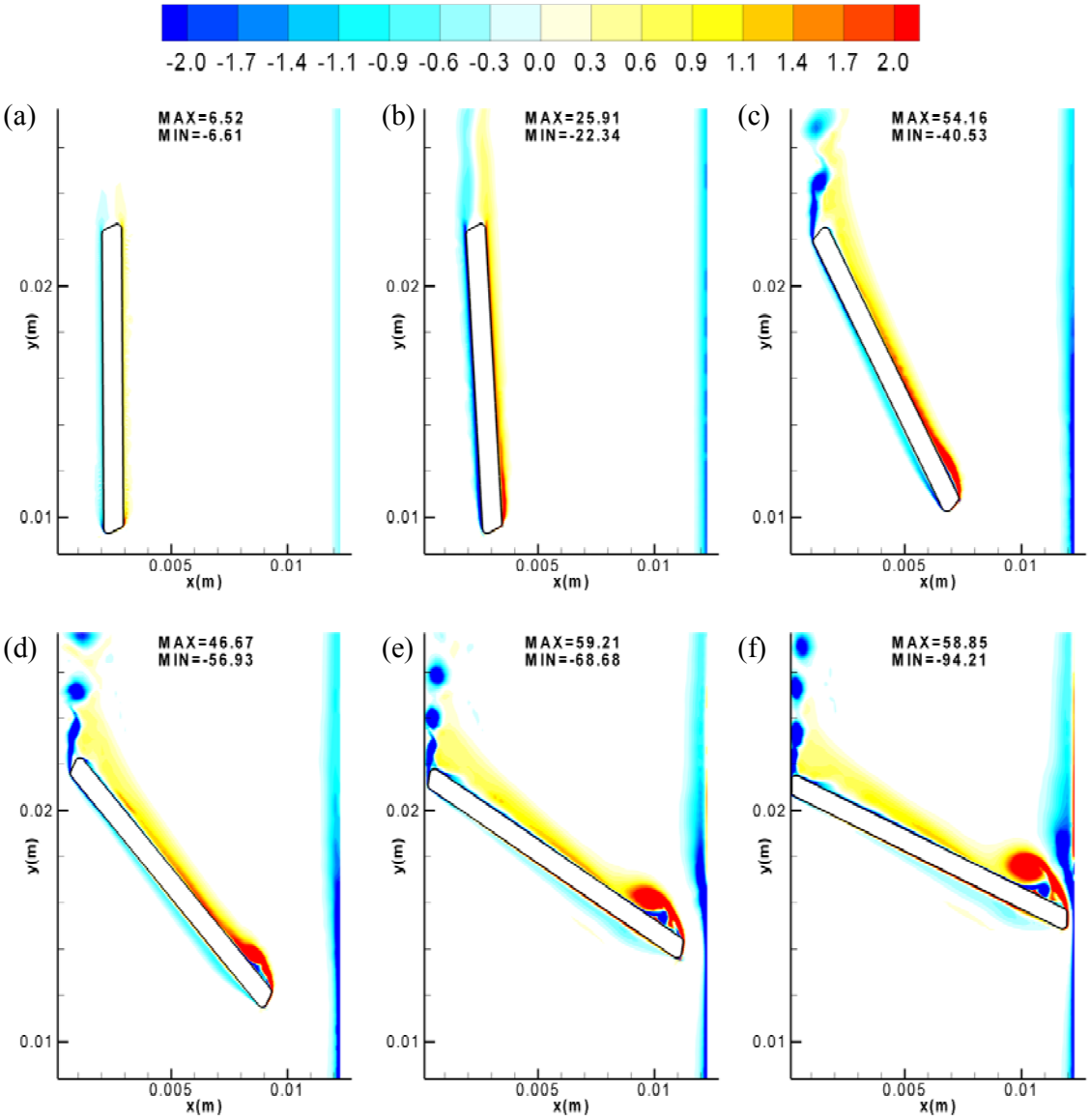


Figure 5.4. Snapshots of closing phase of leaflet showing vorticity contours. Maximum and minimum vorticity strengths are indicated as well. (a) $t = 15.01ms, \theta = 0.4^0, \omega = 0.28 \times 10^{2\ 0}/s$ .Valve is almost in open position. (b) $t = 25.05ms, \theta = 3.2^0, \omega = 0.14 \times 10^{4\ 0}/s$. (c) $t = 29.07, \theta = 25.7^0, \omega = 0.15 \times 10^{5\ 0}/s$. The unstable shear layer breaks up and rolls up into periodically shed vortices. (d) $t = 29.81, \theta = 39.2^0, \omega = 0.23 \times 10^{5\ 0}/s$. Vortices are shed periodically from the downstream edge of the valve. (e) $t = 30.4, \theta = 55.3^0, \omega = 0.32 \times 10^{5\ 0}/s$. Vorticity layer at the wall is pulled in by the oppositely oriented vorticity at the leaflet edge. (f) $t = 30.6, \theta = 61.1^0, \omega = 0.15 \times 10^{5\ 0}/s$.

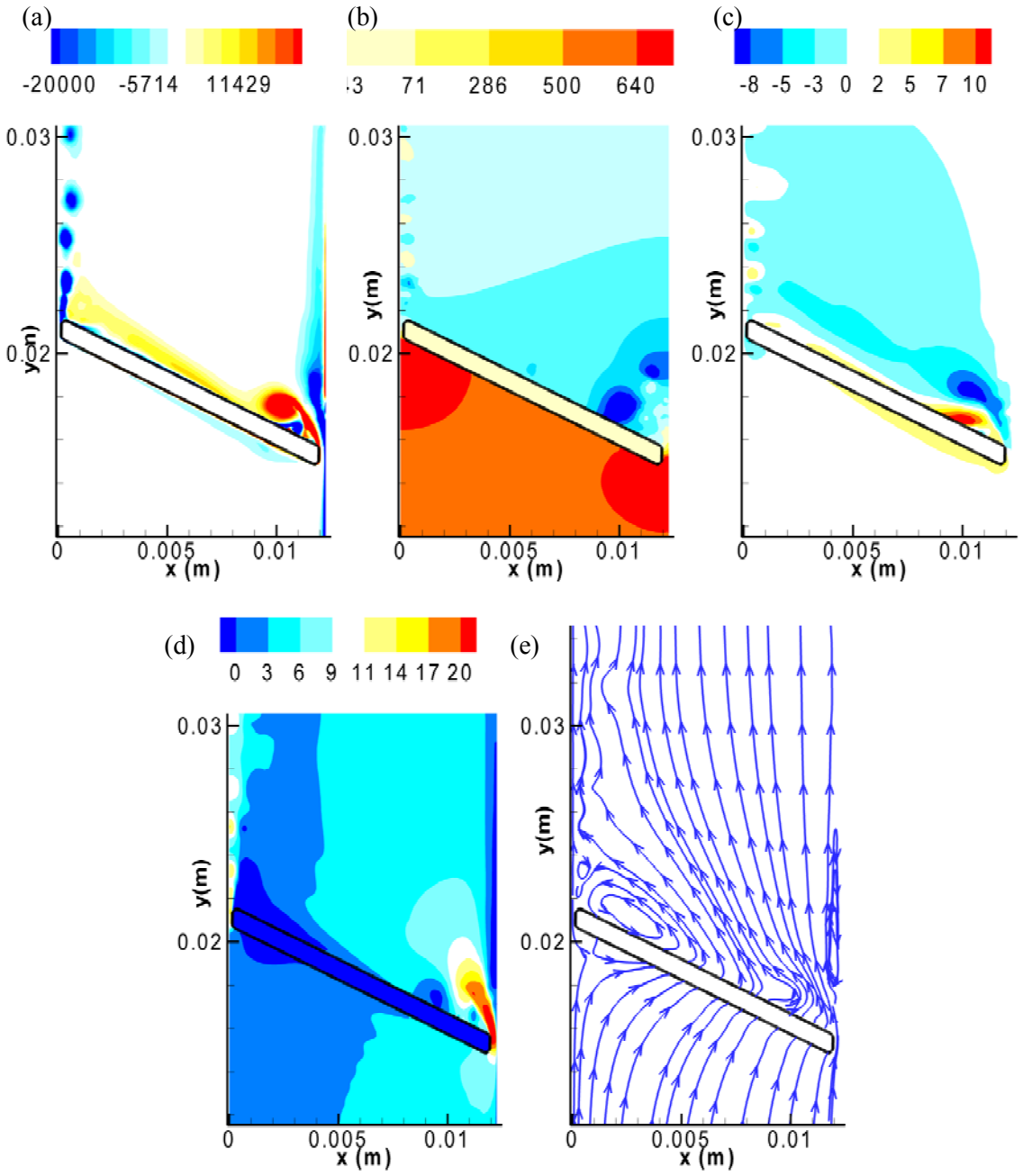


Figure 5.5. Flow characteristics at valve closure. (a) Vorticity contours. On the right side of the leaflet, shear layer separation is observed both from the leaflet and from the wall. (b) Corresponding pressure contours. Once the valve is closed, pressure in the ventricle is mostly uniform. The leakage flow is very strong and here there is enormous pressure gradient across the valve at both the gaps. (c) and (d) show the horizontal and vertical velocities respectively. Maximum velocity is of the order of 20 m/s. (e) Stream lines show the direction of flow. A large recirculation region is formed downstream of the valve.

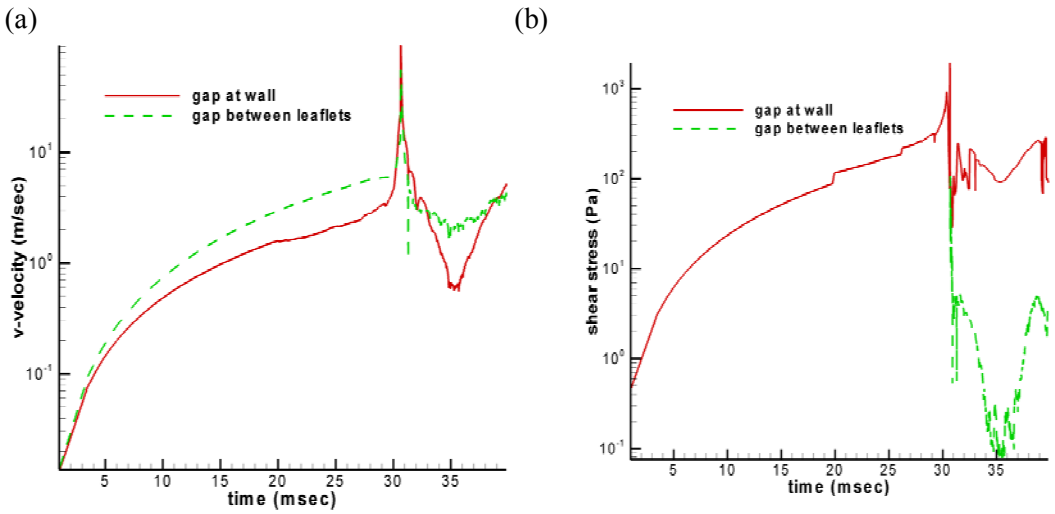


Figure 5.6. Flow parameters recorded in the gap. (a) Velocity and (b) absolute shear stress in the gap between the valve leaflets and the gap between the leaflet and the valve as the valve swings shut.

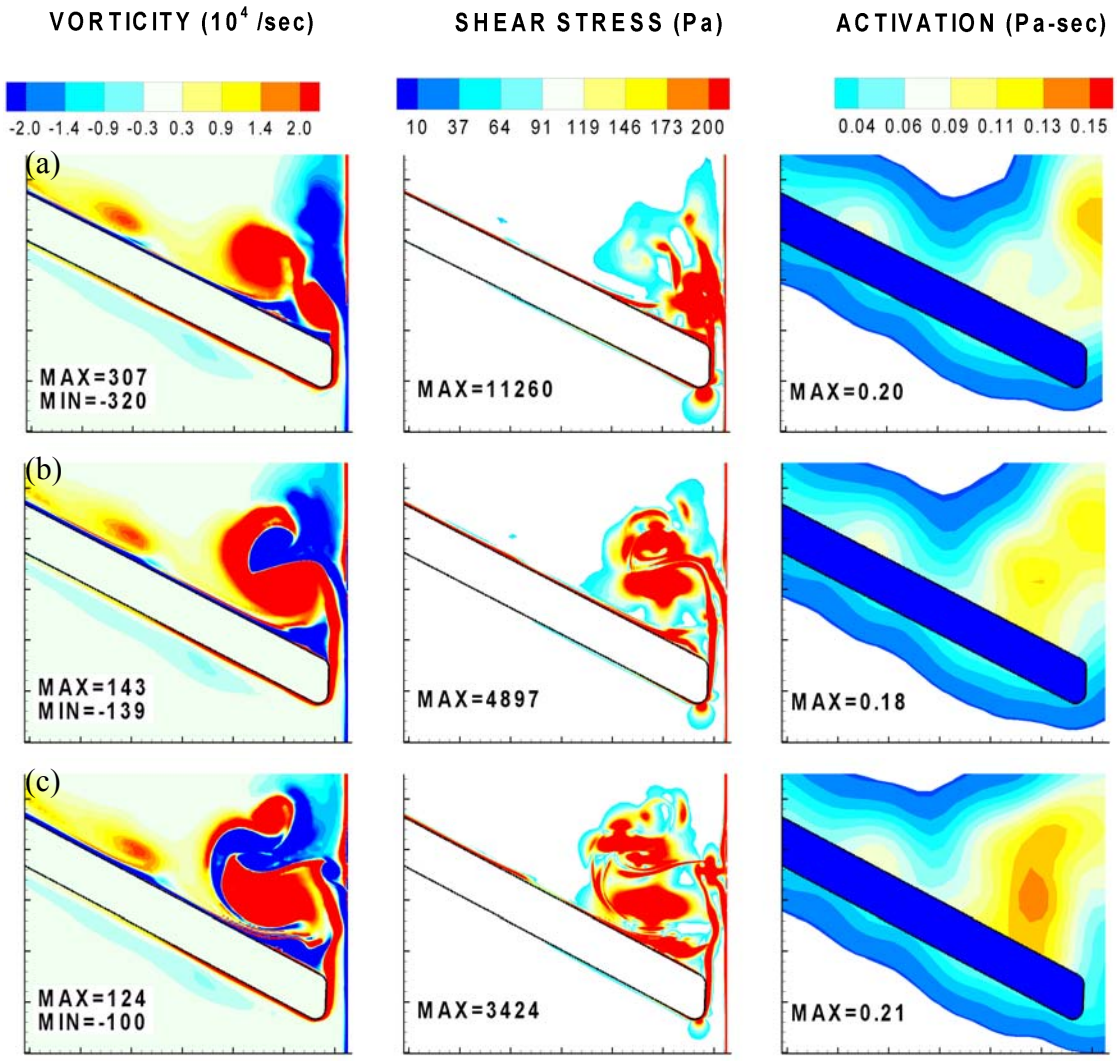


Figure 5.7. Vorticity, shear stress and activation at closure and rebound stage. The first panel in each Figure shows vorticity, the second shows absolute shear stress and the third shows the activation parameter. (a) $t = 30.71, \theta = 63.27^0$, (b) $t = 30.78, \theta = 62.54^0$, (c) $t = 30.86, \theta = 61.95^0$.

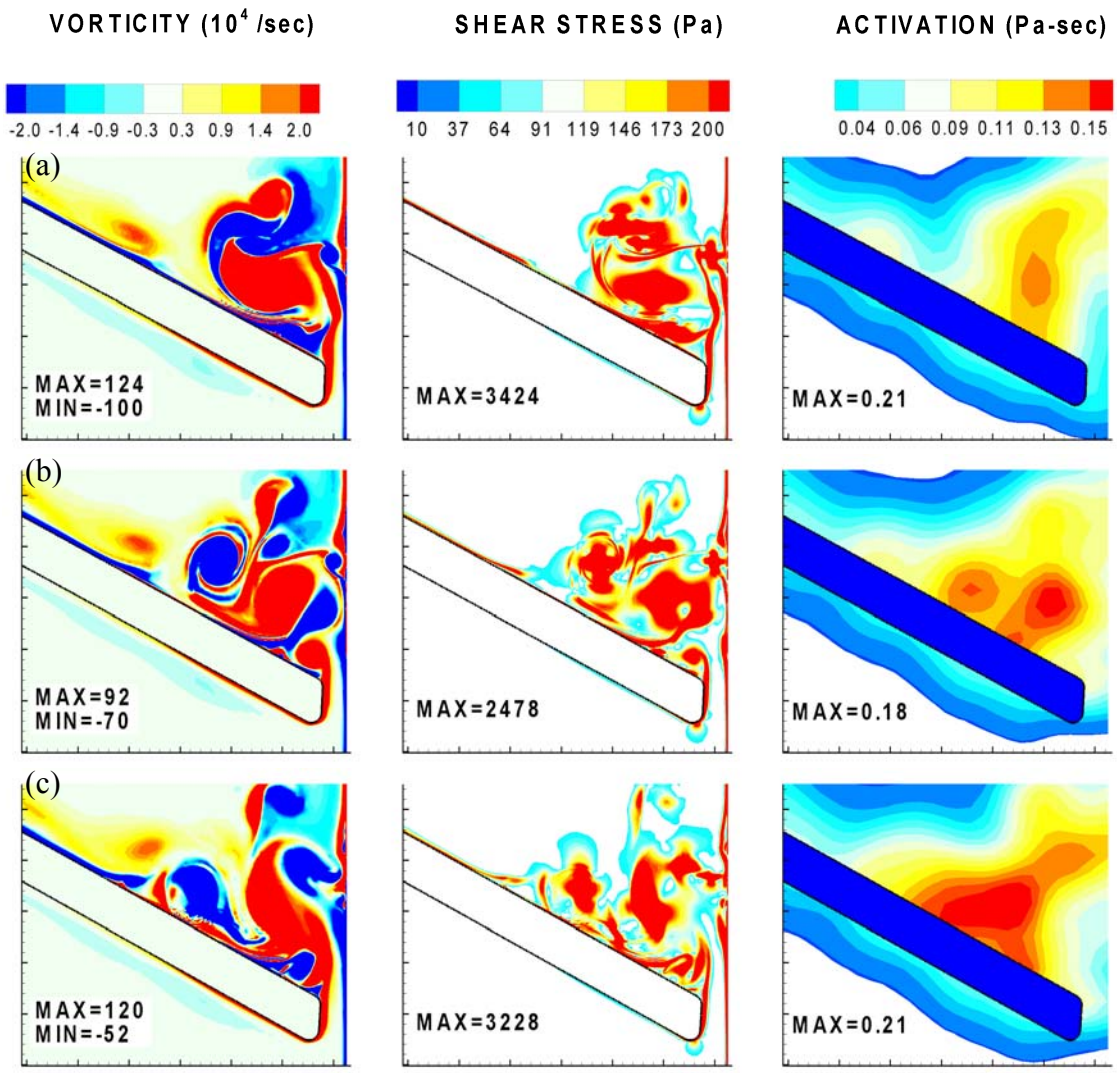


Figure 5.8. Vorticity, shear stress and activation at closure and rebound stage. The first panel in each Figure shows vorticity, the second shows absolute shear stress and the third shows the activation parameter. (a) $t = 30.95, \theta = 61.37^0$, (b) $t = 31.05, \theta = 60.8^0$, (c) $t = 35.2, \theta = 54.69^0$. This position is 5 milliseconds after the first impact of the valve with the housing. The strong vortices are advected away from the leaflet and diffused over time.

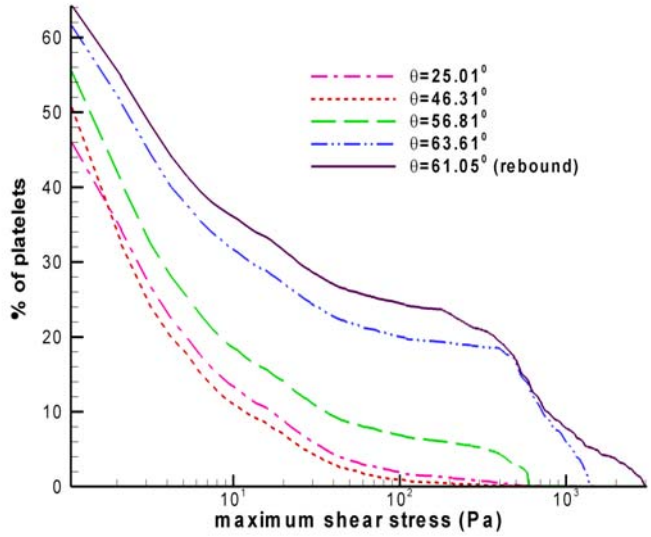


Figure 5.9. Percentage of platelets exposed to the range of shear stress indicated on the x-axis

CHAPTER 6

EFFECT OF LOCAL GEOMETRY ON VALVE PERFORMANCE

6.1 Introduction

Despite years of research, problems associated with heart valve prostheses have not been eliminated and new designs continue to be developed. It has been observed that some models of valves perform better than others with respect to thrombus formation due to small changes in geometry and operation parameters. Differences in geometry include the shape of the edges of the leaflets, the hinge location, thickness and length of the leaflets and orientation of the leaflets in open and closed positions. Operation parameters can cause the valve closure velocity and impact velocity to be different for different valves and this may have major effect on the performance of the valves. The methods developed in Chapters 2 and 3 are applied to comparison of the performance between two models of the bi-leaflet mechanical valve.

Flow velocity and shear stress fields can be significantly different for various prosthetic heart valve designs. Elevated levels of shear stress lead to lethal damage to blood cells as well as platelet activation. Platelets have been shown to be activated when subjected to shear stresses of about $10\,Pa$ [67] and this will also be further affected by presence of foreign surfaces of the valve housing and leaflets. This critical stress level is also affected by the residence time of the cell in the damaging fluid environment, which further complicates the damage mechanism [16-18, 163, 164]. Furthermore, the regions of flow stagnation and flow separation that occur adjacent to the valves could promote the deposition of damaged blood elements, leading to thrombus formation on the

prosthesis.

In bi-leaflet valves, thrombus formation is mostly observed in the hinge region and also on the valve housing [45-47, 66]. It is hypothesized that the local flow conditions in these regions contribute to the thrombus formation. Local flow conditions can be influenced by local geometry and performance parameters of the valve. All the studies on bi-leaflet valves, experimental as well as computational, report flow separation and vortex shedding at both ends of the valve leaflets [17, 95]. The high velocity leakage flow generates regions of high shear stress which are likely sites for platelet activation. The activated platelets have high residence times in the valve vicinity, when caught in regions of stagnant flow or re-circulation regions which are the likely sites of thrombus formation [156, 157]. Computational methods provide an efficient way of analyzing performance of valves by incorporating minor changes without actually going through the process of manufacture and testing. Design changes can be easily tested and validated in a cost-effective way before prototype development and experimental assessment. However, hardware and software constraints force many simplifying assumptions on the computations.

This chapter aims at applying an optimized computational model to compare performance of two valves with respect to shear stresses generated, platelet residence time, intensity of vortices and amount of time for which the vortices persist, all of which are directly related to the amount of platelet activation observed at the valves. The incorporated local mesh refinement enables calculation of macro-scale closure dynamics as well as more detailed flow features of the leakage flow regions.

6.2 Simulation conditions

The two geometries of the valve used in the current simulations are shown in Figure 6.1. The two valves will be henceforth referred to as valve-1 and valve-2. The 2D geometry has been extracted from 3D models of the actual valves and the dimensions correspond to actual valves available in the market. The left boundary of the domain is symmetric while the right boundary represents the valve housing. The leaflet edge at the symmetry side will henceforth be referred to as the left edge and the housing side will be referred to and the right edge of the leaflet.

The differences in geometry and dimensions of the valves are clearly marked in Figure 6.1. The first main difference is in the orientation of the valves with reference to the vertical axis. In the fully open position, valve-1 makes an angle of 5° with the vertical axis and valve-2 makes an angle of 0.2°. Valve-1 sweeps and angle of 60° for complete closure while valve-2 sweeps an angle of 63.8°. Therefore valve-2 has to sweep a larger angle for complete closure. The gaps between the leaflet and the valve housing and between the leaflets is adjusted to be 0.04 cm for both valves. The hinge locations, thickness and length of the valves are slightly but not significantly different. The other major difference between the valves is the shape of the leaflet edge. As shown in Figure 6.1(c) and Figure 6.1(d), valve-1 has a sharper and tapered right edge while valve-2 has a smooth and rounded edge. The left edge of both leaflets are similar looking with slightly rounded edges. The rest of the simulation conditions for the valves are exactly alike.

The region below the valve represents the ventricular side and the region above is the atrium. The pressure on the top boundary on the atrial side is held fixed at 0 $mmHg$ while the pressure on the bottom boundary on the ventricular side is increased linearly

from $0\,mmHg$ to $120\,mmHg$ in a time period of $60\,ms$ at a constant pressure rise rate of $2000\ mmHg/s$. The large angle of rotation does not provide any challenge to the Eulerian method since the mesh does not have to be modified to account for the rotation. However, the challenge lies in adequately resolving the small gaps between the two leaflets and the leaflet and the wall. The gap between the two leaflets as well as between the leaflet and the wall is specified as $0.04\,cm$ while the overall width of the domain itself is of the order of $1.0\,cm$.

The fluid is assumed to be incompressible, laminar, and Newtonian with the density of 1056 kg/m^3 and viscosity of 0.0035 $kgm^{-1}s^{-1}$, which is representative of human blood properties at $37\,°C$. At the symmetry boundary, normal velocity component is set to zero and all other quantities are extrapolated assuming zero normal gradient. The remaining boundaries are solid walls, and the standard no-slip condition is used. Note that the leaflet rotation is calculated from the fluid forces acting on it. Platelets are assumed to be point spheres of radius $2\,\mu m$ and density same as that of blood.

6.3 Results

Figure 6.2 shows the closure characteristics of the two valves. Figure 6.2 (a) shows the orientation of the two valves with time. Valve-1 closes significantly faster than valve-2 because of its initial orientation and lower closure angle. Valve-1 is observed to close in about 28 ms while valve-2 closes in $37\,ms$. The simulation is then continued till 10 ms after the initial closure in the rebound stage. Once the valve strikes the valve housing in the initial closure stage, the angular velocity is reversed as discussed in Section 5.3 with a resilience factor of 0.5. The valve re-opens slightly in this stage before

closing again. Figure 6.2 (b) shows the angular velocity of the two valves in the closure and rebound stages. As seen in the figure, the angular velocity at the instant of closure for valve-1 is about $15000^\circ/s$ while that of valve-2 is $20000^\circ/s$. In the initial stages of valve closure till about 10°, the angular velocity is very low. After this point, there is exponential rise in the leaflet rotation rate. As the leaflet swings shut, it obstructs the flow from the ventricle to the atrium causing enormous pressure build-up on the upstream side of the leaflet. This causes the forces acting on the leaflet to increase correspondingly, leading to exponential increase of angular velocity after a certain closure angle. Added to that is the effect of increased ventricular pressure and increased flow from the ventricle to the atrium.

The higher angular velocity of the valve-2 is due to many factors. Initially, it is aligned at 0.2° with the vertical axis. Starting from this position, due to the alignment of the leaflet along the flow direction, the forces causing the valve to swing shut are very low and correspondingly, the angular velocity is very low. From this position, as seen in Figure 6.2 (a), the leaflet moves very little for the first 30 ms of the closure cycle. By the time, the leaflet is aligned at 5° with the vertical axis after $30\,ms$, the ventricular pressure has already risen to about $60\,mmHg$. For the same leaflet alignment for valve-1, the ventricular pressure is $0\,mmHg$. Hence the force experienced by the valve-2 in the exponential part of the closure curve is much higher than that experienced by valve-1. Valve-1 is completely closed at 60°, while valve-2 has to swing through another 4° to close completely. The longer the leaflet stays in the exponential part of the curve, the higher the angular velocity rise. All these factors contribute to the much higher angular velocity of valve-2. Higher angular velocity at the instant of closure will cause a much

more forceful impact against the valve housing and a violent rebound with implications to higher shear stresses and consequent increased platelet activation. Figure 6.2 (c) shows the valve tip velocity of the valves as the leaflet swings shut. Tip velocity is dependent on both the angular velocity of the valve and the distance of the leaflet tip from the hinge. As seen in the figure, the tip velocity of valve-1 is around 2.8 m/s and that of valve-2 is 3.6 m/s. The tip distance of valve-2 from the hinge is slightly higher than that of valve-1.

Figure 6.3 shows the velocity and pressure at probe points placed in the middle of the gap between the right edge of the leaflet and the housing. Peak velocities from Figure 6.3 (a) at the instant of valve closure and initial stages of valve rebound are seen to reach as high as 25 m/s for valve-1 and 35 m/s for valve-2. Peak negative pressures are recorded at the instant of closure in Figure 6.3 (b). Overall peak values are higher for valve-2 because of the longer cycle time and consequential higher ventricular pressure at closure point. The high velocity during this period of rebound is because the energy which was previously responsible for valve rotation is now diverted into pushing the fluid through the small gap. With the high velocities built up the flow results in very large shear stresses and negative pressure in the gap flow during this period. This has implications in likelihood of cavitation and platelet activation in this stage.

Figure 6.4(a) and Figure 6.4 (b) record maximum shear stress and minimum pressure recorded in the domain as the valve closes. The data has been processed to smooth out the peaks. But two things are immediately evident. One, that the shear stress and pressure recorded for valve-2 is higher than that of valve-1. Second, the shear stress peak stresses are much higher that that required for platelet activation. Figure 6.4 (c) shows a statistical representation of the activation parameter for the two valves, 10 ms

after the valve closure. A larger fraction of platelets are seen to have reached higher activations levels in the case of valve-2 as compared to valve-1.

Figure 6.4 and Figure 6.5 shows the comparison of vorticity contours for valve-1 and valve-2 at the same angle of orientation made with the vertical axis. Note that valve-2 reaches the same orientation as valve-1 about 10 ms after valve-1 does. As explained previously, this is because valve-1 starts out at 5° in the fully open position. Since valve-2 starts out at 0.2° aligned with the direction of flow, the initial stages of closure are very slow. Judging by the vorticity patterns, the flow structure in both cases is very similar. Both valve exhibit similar patterns of flow separation from the leaflet edges and eventual orientation of flow towards the central axis. On the left edge of the leaflet, both valves exhibit periodic shedding of vortices. As seen in Figure 6.5 (f), which is at a time instant very close to the closure point, the shear layer from the left edge of the leaflet is seen to pull in the wall boundary layer. The only difference in the flow as seen from the Figure is in the intensity of the vortices. For the same angle of orientation of the leaflets, the intensity of vorticity is higher in the case of valve-2. The angular velocity at the instant of closure is also higher for valve-2. Because valve-2 closes about $10\,ms$ after valve-1, the high shear region near the right edge of the leaflet can also be expected to have persisted for a longer period of time compared to valve-2. Similar flow patterns seem to indicate that the local geometry of the leaflet edges do not seem to have much of an effect in changing the pattern of flow in the region. The intensity of vortices can be directly related to the higher angular velocity acquired by valve-2 in the closure period.

Figure 6.6 shows the vorticity contours, shear stress and activation parameter at time instants, $2\,ms$, $4\,ms$ and $8\,ms$ after the closure instant in the rebound stage of valve-

1. Figure 6.7 shows the same parameters as above for valve-2 at the same time instants. Both valve-s exhibit the highly dynamics region at the right tip of the valve where the shear layer from the leaflet edge interacts with the boundary layer separating from the housing. This is a region of high shear due to the highly rotating nature of the flow. In each of these instants, it can be observed that the activation parameter for valve-2 is at least 3 times the value of the parameter for valve-1. A comparison of the vorticity contours show that though the flow patterns are similar, the vorticity magnitude is higher in case of valve-2. In the case of valve-1, the vortices diffuse quickly and are almost cleared out by the end of $8\,ms$. For valve-2, the vortices are seen to be still strong at this time. Correspondingly, the shear stress is also higher for a longer period of time for valve-2. Overall, valve-2 exhibits about 3 time higher activation that valve-1. The higher vorticity magnitude shown by valve-2 at the instant of closure also contributes to its persistence for a longer time and consecutive higher shear stress and activation rate.

6.4 Conclusions

Two models of bi-leaflet mechanical heart valves are compared in terms of their effectiveness against platelet activation and thrombus formation. It is found that local geometry does not seem to have a critical effect on the flow patterns generated in the valve-closure cycle. Instead, the global characteristics of the valve such as closure time are critical in deciding the performance of the valve. Flow patterns are found to be qualitatively similar for both the valves compared but closure time, angular velocity and generated stresses are different. One of the valves has a longer closure time and higher magnitudes of closure velocity and consequently higher vorticity strength and shear stress. The difference in magnitudes of critical parameters is mainly found to be affected

by the initial orientation of the valve. Therefore, the most critical parameter that affects the valve closure and subsequent shear stresses and platelet activation in the models under consideration is the initial and final orientation of the leaflets and one of the valves is found to exhibit three times higher platelet activation than the other. It is possible that knowing the exact patterns of flow, tweaking the operational parameters of the valves further could reduce platelet activation by significant amounts. Some recommendations to improve the valve performance would be (i) change the geometry of the edge of the leaflet so as to break up the shear-layer formed at the edge before it interacts with the boundary layer from the valve housing, (ii) minimize the valve tip velocity at closure by changing the closure pattern especially the exponential increase of velocity in the final stages of closure, (iii) optimize the closure time to minimize the valve tip velocity at closure.

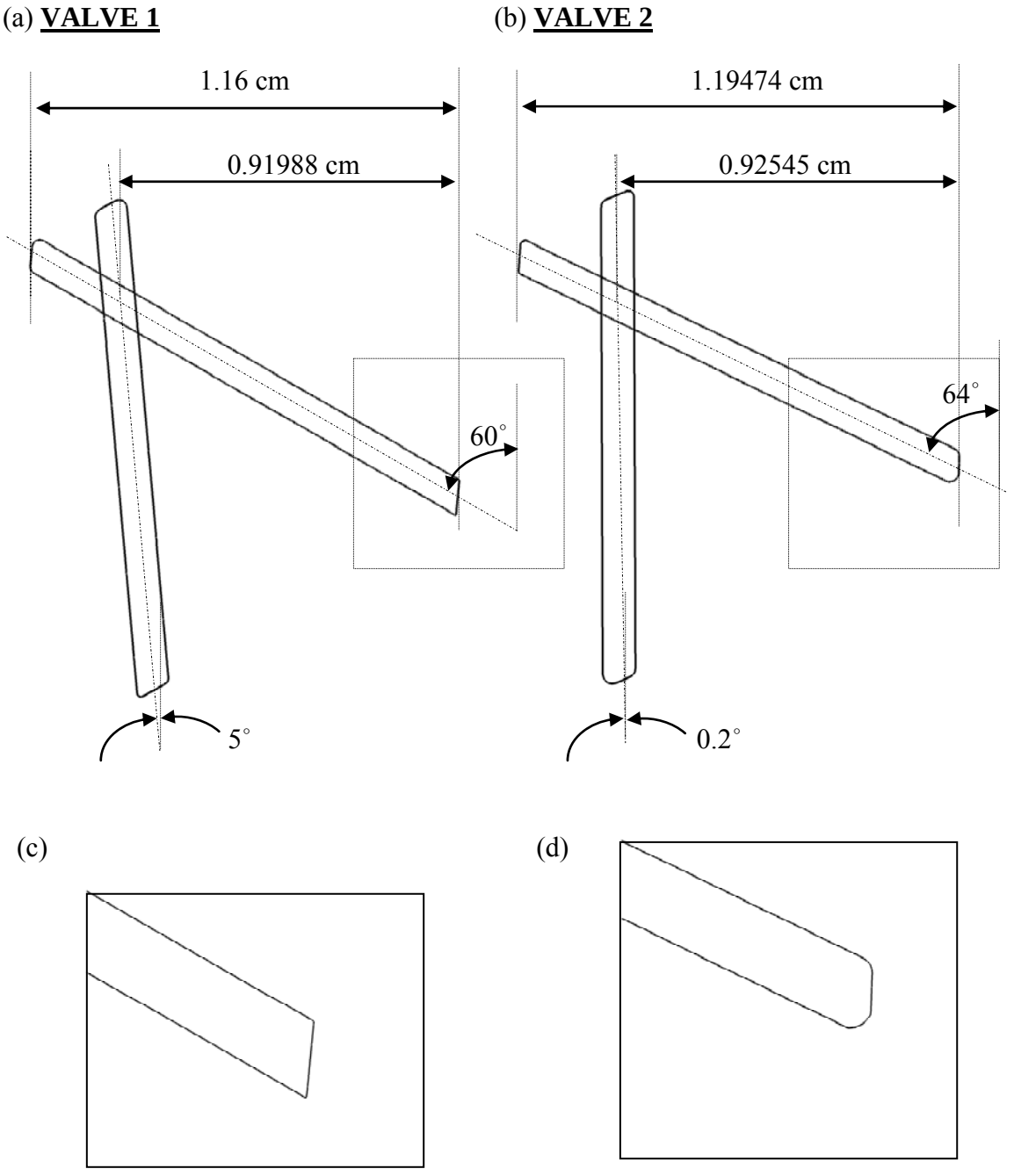


Figure 6.1. Comparison of valve geometries. (a) This is indicated as valve 1. The edge geometry of this valve (c) is sharper than that of valve 2 shown in (d). The angle made by the valves in fully open and closed positions are also indicated in the Figure. The valve dimensions also slightly differ from each other.

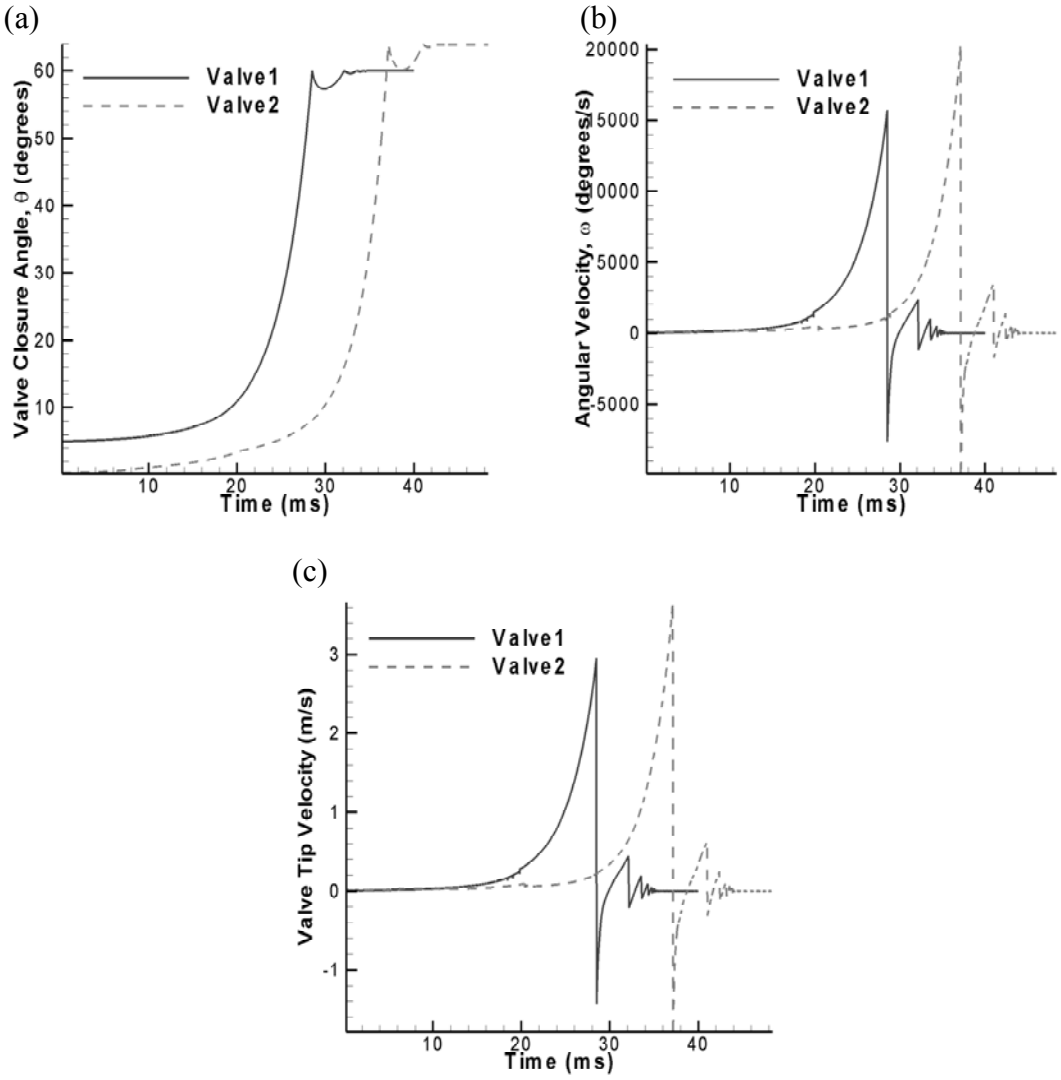


Figure 6.2. Valve closure characteristics of valve 1 and valve 2. (a) Closure angle, (b) angular velocity, (c) valve tip velocity. Valve 1 closes much faster than valve 2. The angular velocity and valve tip velocity of valve 2 are much higher.

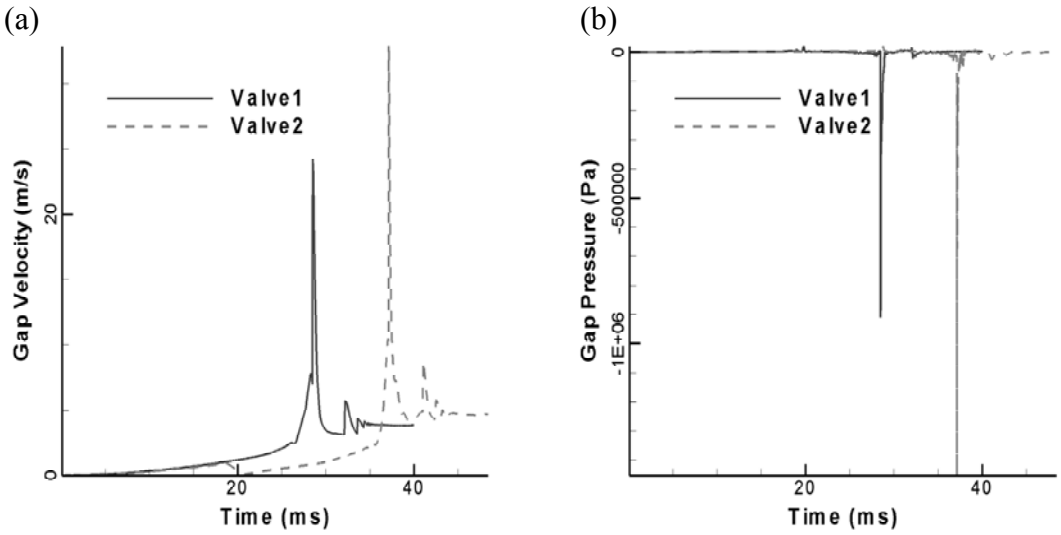


Figure 6.3. Flow parameters in gap between leaflet and wall through closure cycle. (a) Velocity, (b) shear stress and (c) pressure. Corresponding to the data in Figure 6.2, the flow parameters in the gaps are much higher for valve 2.

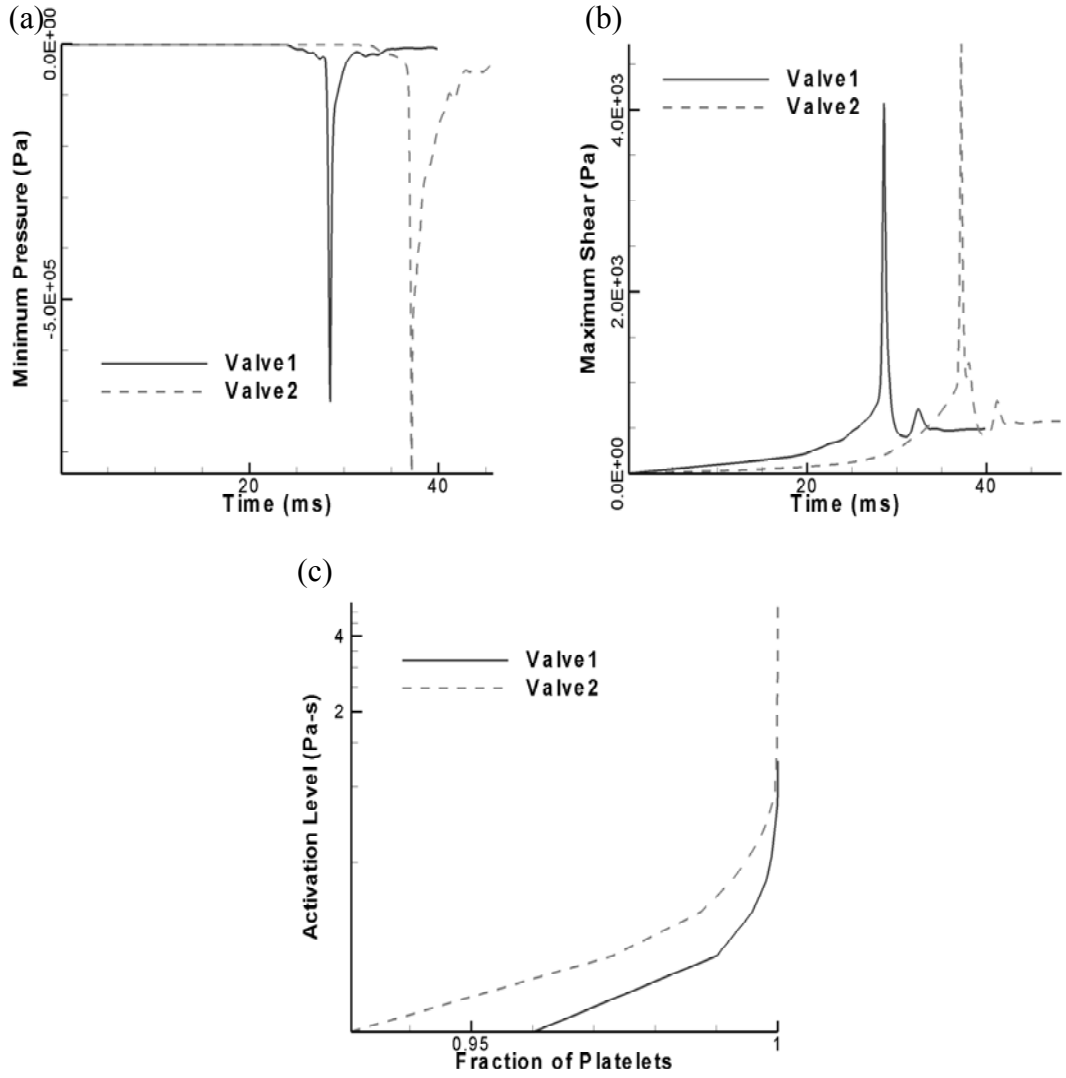


Figure 6.4. Comparison of flow parameters between the two valves. (a) Minimum pressure recorded in domain. (b) Maximum shear stress. (c) Activation level as a function of fraction of platelets. Valve 2 shows lower negative pressures indicating higher likelihood of cavitation and higher shear stresses indication higher likelihood of platelet activation.

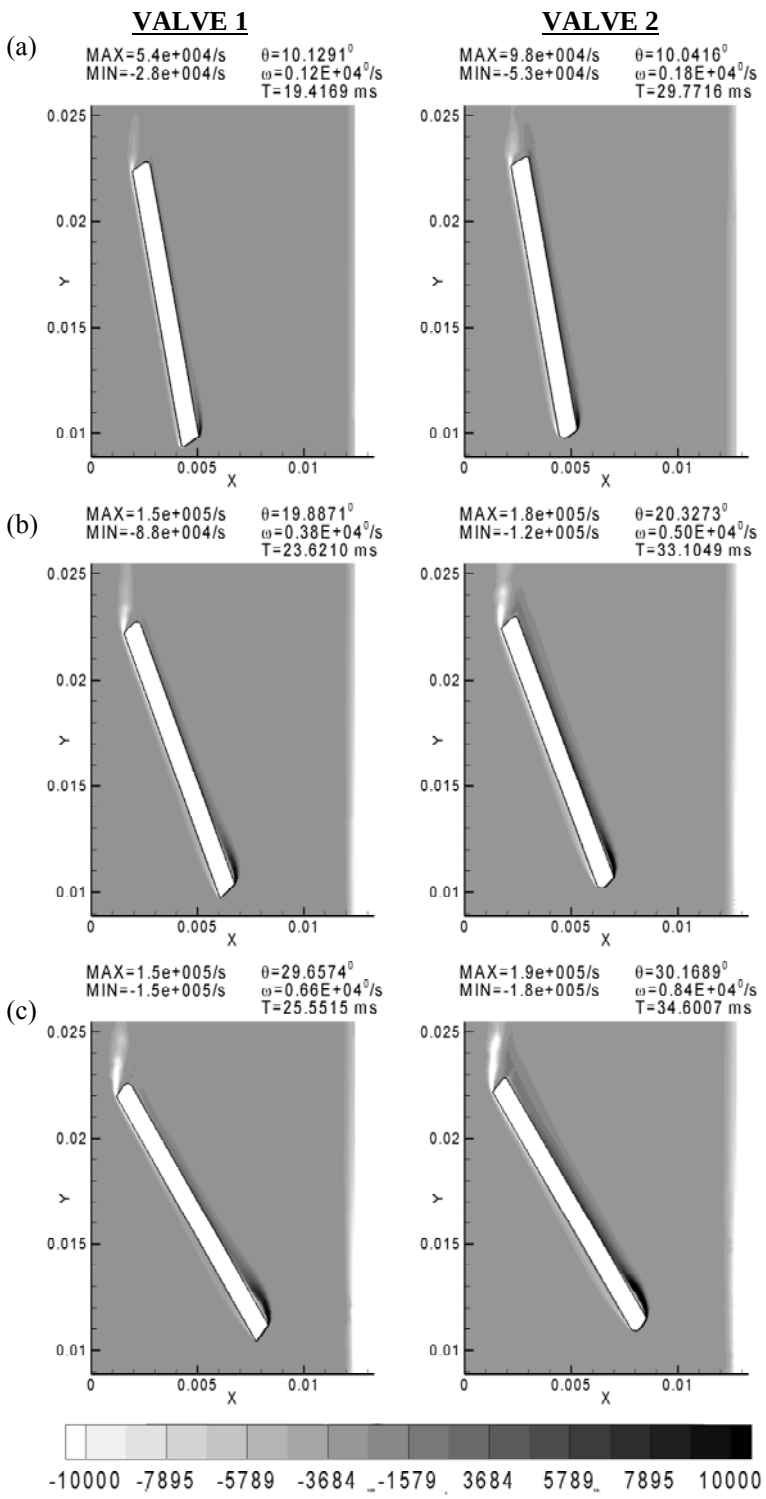


Figure 6.5. Comparison of early closure stages of valve 1 and valve 2. Note that the intensity of vortices is much lower for the first valve. Qualitatively, flow patterns in both cases are similar.

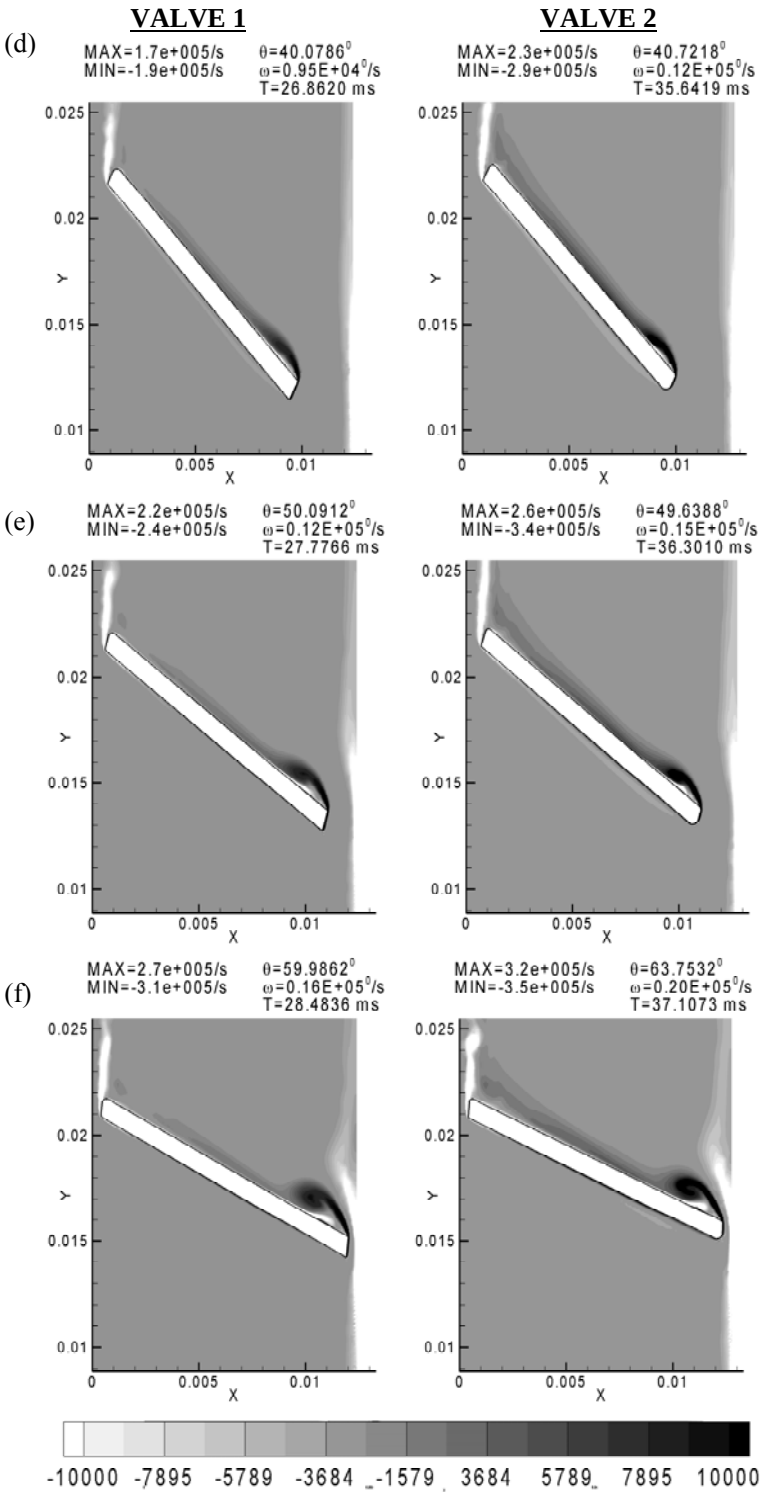


Figure 6.6. Continuation of Figure 4. Comparison of later stages of valve closure between valve 1 and valve 2.

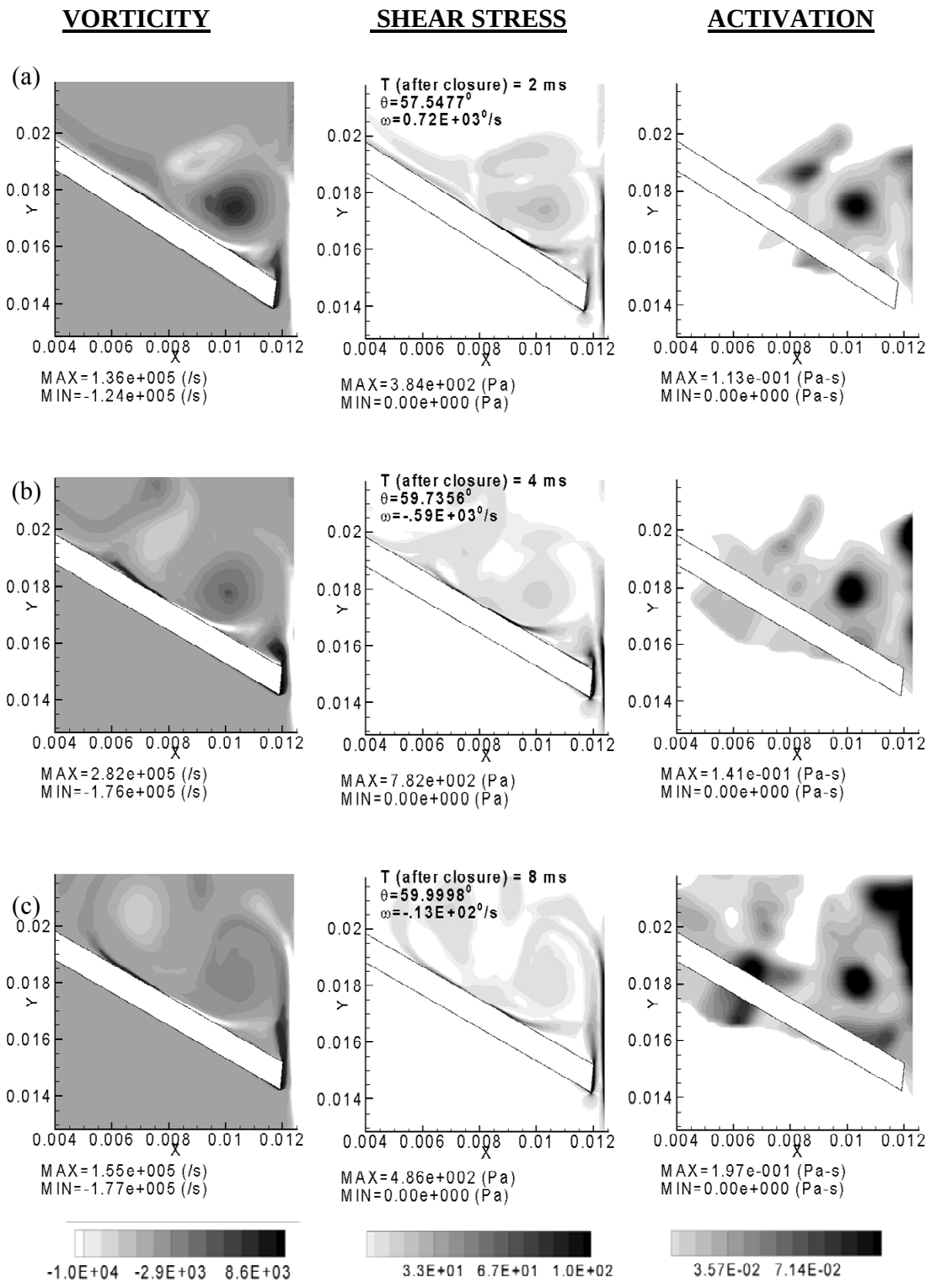


Figure 6.7. Rebound stages of valve 1. The first panel shows vorticity contours, the second shows shear stress and the third shows activation level. Qualitatively, both valve show similar flow behavior. However, intensity of vorticity is much higher for valve 2.

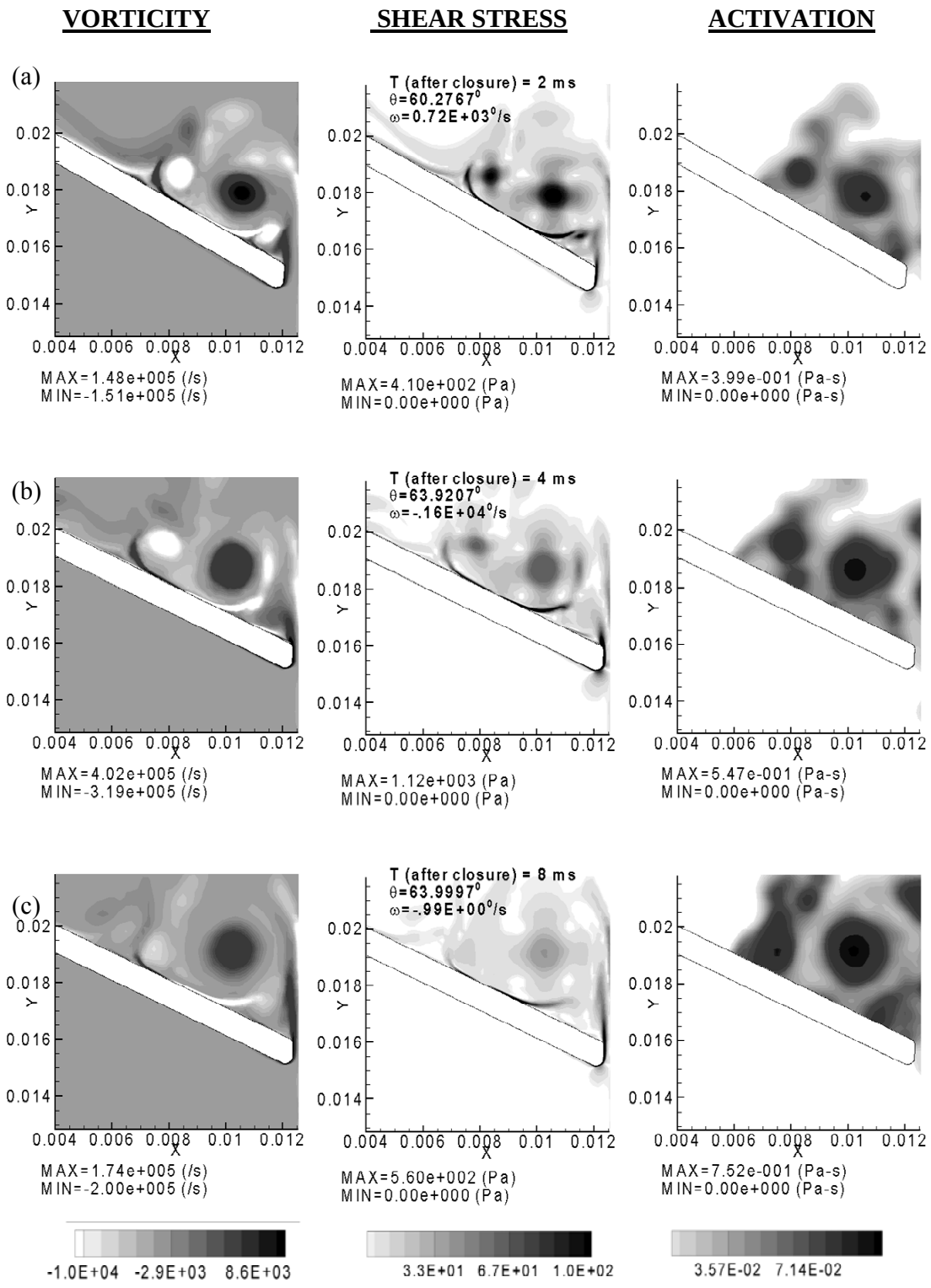


Figure 6.8. Rebound stages of valve 2. The first panel shows vorticity contours, the second shows shear stress and the third shows activation level.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Contributions of the Current Thesis

The contributions of the current thesis can be summarized as follows:

1. A generalized and highly efficient Sharp Interface Cartesian Grid Method for simulating moving boundary problems has been developed in this thesis. The formulation allows the method to be applied to both solid-fluid and fluid-fluid interfaces with equal ease. The method has been extensively validated on moving boundary problems involving solid-fluid and fluid-fluid interfaces as well as phase-change applications.
2. An algebraic multigrid solver has been implemented to complement the sharp interface method to speed up the solution of the pressure Poisson equation. Local coarsening has made the re-generation of grids more efficient in the presence of moving boundaries.
3. A particle tracking algorithm has been integrated with the sharp interface flow solver for calculations involving transport of particulate matter and their interaction with moving boundaries.
4. A quadtree based local mesh refinement scheme has been formulated and implemented to complement the sharp-interface Cartesian grid solver in 2D and 3D to considerably speed-up and optimize the solution process. Mesh generation is automatic and solution dependent.

5. The local refinement scheme has been validated extensively against benchmark solutions on a wide variety of applications and tested for optimizing the parameters that control the refinement process.
6. The above developed method has been applied to detailed analysis of flow during mechanical heart valve closure process. Platelets in blood have been modeled as particles and tracked with the particle tracking algorithm. Shear history on the platelets has been recorded to predict regions with a high likelihood of platelet activation and subsequent thrombus growth. Two geometries of mechanical vales are compared and the factors that affect their performance are explored.

7.2 Future Work on the Numerical Front

A generalized and highly efficient Sharp Interface Cartesian Grid Method for simulating moving boundary problems has been developed in this thesis. The formulation allows the method to be applied to both solid-fluid and fluid-fluid interfaces with equal ease. The above method has been extended by including a Local mesh refinement scheme which considerably reduces the mesh requirement for the solution of complex moving boundary problems. As the solution evolves, the mesh is adapted according to pre-set criteria which minimize user intervention during initial meshing. The optimum criteria for mesh adaptation have been explored by the solution of varied applications and comparison with benchmark solutions and speed-ups have been recorded. The refinement scheme has proved to be highly efficient in reducing memory CPU time requirement for several applications with some cases showing a 100 time speed-up over uniform fine mesh solutions.

The solution time for the simulation of mechanical heart valve closure through the

entire closure and rebound phases which required several months previously has been reduced to the order of several days with the refinement scheme. Particularly, the refinement scheme has been shown to be essential for resolving the small gaps at valve closure. There are however, several extensions to the current method that need to be implemented to make it more versatile. The immediate improvements suggested on the numerical aspects are as follows:

1. Though the local refinement scheme has been implemented in 3D, the actual solution of moving boundary problems in 3D within the framework of sharp interface Cartesian grid method is still formidable in terms of the heavy mesh requirement. These calculations are still time intensive owing to the large scales of these problems. Parallelization of the code is essential for effective solution of 3D problems in real-time. Hence parallelization of the solver is a necessity if calculations are to be performed with accuracy in real time. Parallel algorithms have been shown to achieve almost linear speed up of computational time by distributing tasks among multiple processors optimally with minimum communication requirements between them. To achieve optimum scalability, parallel simulations need the computational domain to be partitioned equally between the processors such that no processor will remain idle while it waits for the other processors to finish their job. Hence load balancing or domain partitioning is an important consideration when applied to locally refined meshes. Efficient grid partitioning and load balancing strategies have been investigated by many researchers for locally refined meshes [99, 100]; [50, 51]
2. Currently, solution gradient and curvature based criteria have been adopted to control mesh adaptation. Every problem is different and intelligent choices of the refinement

control parameters may be necessary to obtain fast reliable solutions. Improvement of the refinement criteria are needed to make mesh adaptation more user-independent such that boundary layers and shear layers can be resolved effectively. Wavelet based criteria are being explored for this purpose.

3. Formulation of finite volume schemes are infinitely complicated especially in 3D. The current work adopts a finite difference scheme for the discretization around interfaces. For mass conservation, this entails the mesh around immersed objects to be of the same refinement level. In some problems, this places severe restriction on the mesh refinement scheme. An improved formulation around immersed interfaces needs to be explored to allow mesh interfaces across immersed objects.
4. A second order central difference scheme is used for the discretization of the governing equations. In certain cases, this can be highly restrictive because of its dispersive nature. A natural way of switching between higher order and lower order schemes needs to be adopted as the complexity of the problems under consideration is increased.
5. The current particle tracking algorithm assumes a dilute particle loading. Improvement of the particle tracking algorithm is required to account for the effects of particle-particle interactions as well as effect of the presence o particles on the fluid flow.
6. Most of the flows encountered in nature are turbulent in nature. The current method needs to be extended by including turbulent flow models.
7. Extending the framework of the current method to solve compressible flow problems will enormously increase the range of applications that can be tackled with the code.

7.3 Improving the Mechanical Heart Valve Simulations

Flow analysis of mechanical heart valve closure indicates that the interaction of boundary layer separating from the valve housing with the boundary layer separating from the valve leaflet during the rebound phase causes the formation of a region of high shear stress at the tip of the valve leaflet. This is also a region of highly re-circulating flow indicating high platelet residence time. The current simulations indicate that this is a region with high likelihood of platelet activation and thrombus formation. Comparison of performance of two commercially available replacement heart valves indicates that the orientation of the valve in open and closed positions and the angle through which the leaflet swings during closure is the major factor that affects the intensity of interaction of wall and leaflet boundary layers. Consecutively, this may be the single important factor that determines the performance of replacement valves vis-à-vis platelet activation and thrombus formation. Further insights into the valve performance may be obtained by simulating the entire valve cycle including the opening phase. Simulation of consecutive valve cycles may further indicate pockets of flow stagnation and persisting high shear stress regions.

Previous studies [28, 29] have shown that while 2D flow analysis is able to capture the valve closure dynamics qualitatively, quantitative comparison will require 3D models. Meaningful 3D simulations, i.e. those that will capture the details of the leakage flow, require very dense meshes and will require parallel computing to obtain results with reasonable computational effort. Work is currently underway to accomplish this.

With regard to potential sites of platelet activation, a limitation of the 2D model is that hinge geometry cannot be incorporated in the flow simulation. Platelet activation and

thrombus formation is reported to occur due to the leakage flow in the hinge region and much previous research has concentrated on this aspect [47, 64, 88, 120]. The dimension of the hinge region is two orders of magnitude lower than the valve dimension and will need to be considered in the computational analysis.

With regard to the flow field features, it has been suggested [163, 164] that turbulent flow can ensue in the leakage jet and the resulting turbulent stresses can have a significant impact on the platelet activation and thrombus initiation. In the present simulations, the average Reynolds number based on average velocity in the gap is around 300 while that based on the peak velocity of the leakage jet was approximately 1800. These Reynolds numbers indicate that, at best the flow in the valve may approach the transitional regime for brief durations during the flow. The local velocities in the leakage region approach high values in confined regions for short time durations. The advent of turbulence for flow under such conditions is unlikely. The computations performed in the present work did not require any models to represent the increased dissipation due to turbulent Reynolds stresses. However, the finest (5-level) mesh explored was found necessary to capture the rather steep gradients that occur as the flow from the ventricular side turns the corner into the atrial side at the right edge of the leaflet.

The present platelet activation model is purely based on the shear stress-time integral experienced by the platelets in flowing through the gap width between the leaflet edge and the valve housing. Previous studies have suggested a specific shear stress-time relationship for platelet activation in arterial flows [68, 73]. Tambasco et al., [134] suggested a minimum shear stress beyond which the platelets will be activated in the shear stress-time integral employed in the current thesis. However, the process of platelet

activation, aggregation, and thrombus initiation particularly in the presence of foreign surfaces such as a mechanical valve leaflet is not clear.

There are many other factors like agonist synthesis and release by activated platelets and concentration, platelet-phospholipid-dependent thrombin generation, and thrombin inhibition by heparin that need to be incorporated to build a comprehensive activation model [122, 123]. Inclusion of biochemical effects on platelets accounting for all these factors will improve the prediction of thrombus formation. In this study, a particle dynamics analysis was incorporated in the computational fluid dynamic analysis code in order to compare the concentration of platelets and their residence time in the gap between the leaflet edge and the valve housing and the central gap between the leaflets during the closing phase of a bi-leaflet valve. It was shown that the flow dynamics and the behavior of the vortices are significantly different in the two regions and the platelet activation and deposition is more likely in the clearance gap region.

The current particle tracking algorithm assumes a dilute flow with low platelet loading in blood. It neglects the effect of platelet loading on blood flow as well as interaction of platelets with other platelets and red blood cells. Capturing the full dynamics of platelets, in the presence of red blood cells of biconcave shape in appropriate physiological concentrations, requires multi-scale modeling that will incorporate the effects of the particulate nature of blood, including cell-cell and cell-surface interactions. Efforts are underway to incorporate multi-scale models into the macro-scale simulations.

REFERENCES

1. Adalsteinsson, D. and Sethian, J. A., A Fast Level Set Method for Propagating Interfaces. Journal of Computational Physics. 1995. 118(2): 269-277.

2. Aftosmis, M. J., Berger, M. and Adomavicius, G., A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries. AIAA, Reno. 2000.

3. Agresar, G., Linderman, J. J., Tryggvason, G. and Powell, K. G., An Adaptive, Cartesian, Front-Tracking Method for the Motion, Deformation and Adhesion of Circulating Cells. Journal of Computational Physics. 1998. 143(2): 346-380.

4. Al-Rawahi, N. and Tryggvason, G., Numerical Simulation of Dendritic Solidification with Convection: Two-Dimensional Geometry. Journal of Computational Physics. 2002. 180(2): 471-496.

5. Anderson, D. M., McFadden, G. B. and Wheeler, A. A., Diffuse-Interface Methods in Fluid Mechanics. Annual Review of Fluid Mechanics. 1998. 30(139-165.

6. Anderson, D. M., McFadden, G. B. and Wheeler, A. A., A Phase-Field Model of Solidification with Convection. Physica D. 2000. 135(1-2): 175-194.

7. Anderson, R. W., Elliott, N. S. and Pember, R. B., An Arbitrary Lagrangian-Eulerian Method with Adaptive Mesh Refinement for the Solution of the Euler Equations. Journal of Computational Physics. 2004. 199(2): 598-617.

8. Avrahami, I., Rosenfeld, M., Einav, S., Eichler, M. and Reul, H., Can Vortices in the Flow across Mechanical Heart Valves Contribute to Cavitation? Medical & Biological Engineering & Computing. 2000. 38(1): 93-97.

9. Balaras, E., Modeling Complex Boundaries Using an External Force Field on Fixed Cartesian Grids in Large-Eddy Simulations. Computers & Fluids. 2004. 33(3): 375-404.

10. Bell, J., Berger, M., Saltzman, J. and Welcome, M., 3-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation-Laws. Siam Journal on Scientific Computing. 1994. 15(1): 127-138.

11. Berger, M. J., Stability of Interfaces with Mesh Refinement. Mathematics of Computation. 1985. 45(172): 301-318.

12. Berger, M. J. and Colella, P., Local Adaptive Mesh Refinement for Shock Hydrodynamics. Journal of Computational Physics. 1989. 82(1): 64-84.

13. Berger, M. J. and Leveque, R. J., Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems. Siam Journal on Numerical Analysis. 1998. 35(6): 2298-2316.

14. Berger, M. J. and Oliger, J., Adaptive Mesh Refinement for Hyperbolic Partial-Differential Equations. Journal of Computational Physics. 1984. 53(3): 484-512.

15. Bluestein, D., Einav, S. and Hwang, N. H. C., A Squeeze Flow Phenomenon at the Closing of a Bileaflet Mechanical Heart-Valve Prosthesis. Journal of Biomechanics. 1994. 27(11): 1369-1378.

16. Bluestein, D., Li, Y. M. and Krukenkamp, I. B., Free Emboli Formation in the Wake of Bi-Leaflet Mechanical Heart Valves and the Effects of Implantation Techniques. Journal of Biomechanics. 2002. 35(12): 1533-1540.

17. Bluestein, D., Rambod, E. and Gharib, M., Vortex Shedding as a Mechanism for Free Emboli Formation in Mechanical Heart Valves. Journal of Biomechanical Engineering-Transactions of the Asme. 2000. 122(2): 125-134.

18. Bluestein, D., Yin, W., Affeld, K. and Jesty, J., Flow-Induced Platelet Activation in Mechanical Heart Valves. Journal of Heart Valve Disease. 2004. 13(3): 501-508.

19. Bodnar, E., Grunkemeier, G. L. and Gabbay, S., Heart Valve Replacement: A Statistical Review of 35 Years' Results - Discussion. Journal of Heart Valve Disease. 1999. 8(5): 470-471.

20. Borthwick, A. G. L., Leon, S. C. and Jozsa, J., Adaptive Quadtree Model of Shallow-Flow Hydrodynamics. Journal of Hydraulic Research. 2001. 39(4): 413-424.

21. Borthwick, A. G. L., Leon, S. C. and Jozsa, J., The Shallow Flow Equations Solved on Adaptive Quadtree Grids. International Journal for Numerical Methods in Fluids. 2001. 37(6): 691-719.

22. Borthwick, A. G. L., Marchant, R. D. and Copeland, G. J. M., Adaptive Hierarchical Grid Model of Water-Borne Pollutant Dispersion. Advances in Water Resources. 2000. 23(8): 849-865.

23. Brackbill, J. U., Kothe, D. B. and Zemach, C., A Continuum Method for Modeling Surface-Tension. Journal of Computational Physics. 1992. 100(2): 335-354.

24. Brezina, M., Cleary, A. J., Falgout, R. D., Henson, V. E., Jones, J. E., Manteuffel, T. A., McCormick, S. F. and Ruge, J. W., Algebraic Multigrid Based on Element Interpolation (Amge). Siam Journal on Scientific Computing. 2001. 22(5): 1570-1592.

25. Butchart, E. G., Ionescu, A., Payne, N., Giddings, J., Grunkemeier, G. L. and Fraser, A. G., A New Scoring System to Determine Thromboembolic Risk after Heart Valve Replacement. Circulation. 2003. 108(10): 68-74.

26. Chandran, K. B., Cardiovascular Biomechanics. New York University Press, 1992.

27. Chen, H. and Marshall, J. S., A Lagrangian Vorticity Method for Two-Phase Particulate Flows with Two-Way Phase Coupling. Journal of Computational Physics. 1999. 148(1): 169-198.

28. Cheng, R., Lai, Y. G. and Chandran, K. B., Two-Dimensional Fluid-Structure Interaction Simulation of Bileaflet Mechanical Heart Valve Flow Dynamics. Journal of Heart Valve Disease. 2003. 12(6): 772-780.

29. Cheng, R., Lai, Y. G. and Chandran, K. B., Three-Dimensional Fluid-Structure Interaction Simulation of Bileaflet Mechanical Heart Valve Flow Dynamics. Annals of Biomedical Engineering. 2004. 32(11): 1471-1483.

30. Cheng, T. and Peskin, C. S., Stability and Instability in the Computation of Flows with Moving Immersed Boundaries - a Comparison of 3 Methods. Siam Journal on Scientific and Statistical Computing. 1992. 13(6): 1361-1376.

31. Chessa, J. and Belytschko, T., An Extended Finite Element Method for Two-Phase Fluids. Journal of Applied Mechanics-Transactions of the Asme. 2003. 70(1): 10-17.

32. Chessa, J., Smolinski, P. and Belytschko, T., The Extended Finite Element Method (Xfem) for Solidification Problems. International Journal for Numerical Methods in Engineering. 2002. 53(8): 1959-1977.

33. Choi, C. R., Kim, C. N., Kwon, Y. J. and Lee, J. W., Pulsatile Blood Flows through a Bileaflet Mechanical Heart Valve with Different Approach Methods of Numerical Analysis; Pulsatile Flows with Fixed Leaflets and Interacted with Moving Leaflets. Ksme International Journal. 2003. 17(7): 1073-1082.

34. Chopp, D. L., Some Improvements of the Fast Marching Method. Siam Journal on Scientific Computing. 2001. 23(1): 230-244.

35. Cleary, A. J., Falgout, R. D., Henson, V. E., Jones, J. E., Manteuffel, T. A., McCormick, S. F., Miranda, G. N. and Ruge, J. W., Robustness and Scalability of Algebraic Multigrid. Siam Journal on Scientific Computing. 2000. 21(5): 1886-1908.

36. Clift, R., Gracs, J. R. and Weber, M. E., Bubbles, Drops and Particles. Academic Press, London. 1978.

37. Coirier, W. J. and Powell, K. G., An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations. Journal of Computational Physics. 1995. 117(1): 121-131.

38. Coirier, W. J. and Powell, K. G., Solution-Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows. Aiaa Journal. 1996. 34(5): 938-945.

39. Colagrossi, A. and Landrini, M., Numerical Simulation of Interfacial Flows by Smoothed Particle Hydrodynamics. Journal of Computational Physics. 2003. 191(2): 448-475.

40. Crowe, C., Sommerfeld, M. and Tsuji, Y., Multiphase Flow with Droplets and Particles. CRC Press, Boca Raton, Boston, New York, Washington, London. 1998.

41. DeZeeuw, D. L., A Quadtree-Based Adaptively Refined Cartesian Grid Algorithm for Solution of the Euler Equations. Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI. 1993.

42. DeZeeuw, D. L. and Powell, K. G., A Quadtree-Based Adaptively Refined Cartesian Mesh Solver for the Euler Equations. AIAA. 1991.

43. Dolbow, J., Moes, N. and Belytschko, T., An Extended Finite Element Method for Modeling Crack Growth with Frictional Contact. Computer Methods in Applied Mechanics and Engineering. 2001. 190(51-52): 6825-6846.

44. Durbin, P. A. and Iaccarino, G., An Approach to Local Refinement of Structured Grids. Journal of Computational Physics. 2002. 181(2): 639-653.

45. Ellis, J. T., Healy, T. M., Fontaine, A. A., Saxena, R. and Yoganathan, A. P., Velocity Measurements and Flow Patterns within the Hinge Region of a Medtronic Parallel(Tm) Bileaflet Mechanical Valve with Clear Housing. Journal of Heart Valve Disease. 1996. 5(6): 591-599.

46. Ellis, J. T., Healy, T. M., Fontaine, A. A., Weston, M. W., Jarret, C. A., Saxena, R. and Yoganathan, A. P., An in Vitro Investigation of the Retrograde Flow Fields of Two Bileaflet Mechanical Heart Valves. Journal of Heart Valve Disease. 1996. 5(6): 600-606.

47. Ellis, J. T., Travis, B. R. and Yoganathan, A. P., An in Vitro Study of the Hinge and near-Field Forward Flow Dynamics of the St. Jude Medical (R) Regent (Tm) Bileaflet Mechanical Heart Valve. Annals of Biomedical Engineering. 2000. 28(5): 524-532.

48. Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J., Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. Journal of Computational Physics. 2000. 161(1): 35-60.

49. Fedkiw, R. P., Aslam, T., Merriman, B. and Osher, S., A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). Journal of Computational Physics. 1999. 152(2): 457-492.

50. Flaherty, J. E., Loy, R. M., Ozturan, C., Shephard, M. S., Szymanski, B. K., Teresco, J. D. and Ziantz, L. H., Parallel Structures and Dynamic Load Balancing for Adaptive Finite Element Computation. Applied Numerical Mathematics. 1998. 26(1-2): 241-263.

51. Flaherty, J. E., Loy, R. M., Shephard, M. S., Szymanski, B. K., Teresco, J. D. and Ziantz, L. H., Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws. Journal of Parallel and Distributed Computing. 1997. 47(2): 139-152.

52. Gao, Z. B., Hosein, N., Dai, F. E. and Hwang, N. H. C., Pressure and Flow Fields in the Hinge Region of Bileaflet Mechanical Heart Valves. Journal of Heart Valve Disease. 1999. 8(2): 197-205.

53. Ge, L., Leo, H. L., Sotiropoulos, F. and Yoganathan, A. P., Flow. In a Mechanical Bileaflet Heart Valve at Laminar and near-Peak Systole Flow Rates: Cfd Simulations and Experiments. Journal of Biomechanical Engineering-Transactions of the Asme. 2005. 127(5): 782-797.

54. Ghia, U., Ghia, K. N. and Shin, C. T., High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. Journal of Computational Physics. 1982. 48(387-411.

55. Giddens, D. P., Yoganathan, A. P. and Schoen, F. J., Prosthetic Cardiac Valves. Cardiovascular Pathology. 1993. 2(3): S167-S177.

56. Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D. and Periaux, J., A Distributed Lagrange Multiplier/Fictitious Domain Method for Flows around Moving Rigid Bodies: Application to Particulate Flow. International Journal for Numerical Methods in Fluids. 1999. 30(8): 1043-1066.

57. Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D. and Periaux, J., A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow Past Moving Rigid Bodies: Application to Particulate Flow. Journal of Computational Physics. 2001. 169(2): 363-426.

58. Greaves, D., A Quadtree Adaptive Method for Simulating Fluid Flows with Moving Interfaces. Journal of Computational Physics. 2004. 194(1): 35-56.

59. Greaves, D., Simulation of Interface and Free Surface Flows in a Viscous Fluid Using Adapting Quadtree Grids. International Journal for Numerical Methods in Fluids. 2004. 44(10): 1093-1117.

60. Greaves, D. M. and Borthwick, A. G. L., On the Use of Adaptive Hierarchical Meshes for Numerical Simulation of Separated Flows. International Journal for Numerical Methods in Fluids. 1998. 26(3): 303-322.

61. Greaves, D. M. and Borthwick, A. G. L., Hierarchical Tree-Based Finite Element Mesh Generation. International Journal for Numerical Methods in Engineering. 1999. 45(4): 447-471.

62. Greaves, D. M., Borthwick, A. G. L., Wu, G. X. and Taylor, R. E., A Moving Boundary Finite Element Method for Fully Nonlinear Wave Simulations. Journal of Ship Research. 1997. 41(3): 181-194.

63. Griebel, M., Neunhoeffer, T. and Regler, H., Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries. International Journal for Numerical Methods in Fluids. 1998. 26(3): 281-301.

64. Gross, J. M., Shu, M. C. S., Dai, F. F., Ellis, J. and Yoganathan, A. P., Microstructural Flow Analysis within a Bileaflet Mechanical Heart Valve Hinge. Journal of Heart Valve Disease. 1996. 5(6): 581-590.

65. Haase, G., Kuhn, M. and Reitzinger, S., Parallel Algebraic Multigrid Methods on Distributed Memory Computers. Siam Journal on Scientific Computing. 2002. 24(2): 410-427.

66. Healy, T. M., Fontaine, A. A., Ellis, J. T., Walton, S. P. and Yoganathan, A. P., Visualization of the Hinge Flow in a 5: 1 Scaled Model of the Medtronic Parallel Bileaflet Heart Valve Prosthesis. Experiments in Fluids. 1998. 25(5-6): 512-518.

67. Hellums, J. D., 1993 Whitaker Lecture - Biorheology in Thrombosis Research. Annals of Biomedical Engineering. 1994. 22(5): 445-455.

68. Hellums, J. D., Peterson, D. M., Stathopoulos, N. A., Moake, J. L. and Giorgio, T. D., Studies of Mechanisms of Shear-Induced Platelet Activation. Cerebral Ischemia and Hemorheology. 1987. 80-89.

69. Henson, V. E. and Yang, U. M., Boomeramg: A Parallel Algebraic Multigrid Solver and Preconditioner. Applied Numerical Mathematics. 2002. 41(1): 155-177.

70. Huang, S. C. and Glicksman, M. E., Fundamentals of Dendritic Solidification.1. Steady-State Tip Growth. Acta Metallurgica. 1981. 29(5): 701-715.

71. Huang, S. C. and Glicksman, M. E., Fundamentals of Dendritic Solidification.2. Development of Sidebranch Structure. Acta Metallurgica. 1981. 29(5): 717-734.

72. Ivanstov, G. P., Temperature Filed around Spherical, Cylindrical and Needle Shaped Crystals Which Grow in Supercooled Melt. Doklady Akademii Nauk. 1947. 58(567-569.

73. Jesty, J., Yin, W., Perrotta, P. and Bluestein, D., Platelet Activation in a Circulating Flow Loop: Combined Effects of Shear Stress and Exposure Time (Vol 14, Pg 143, 2003). Platelets. 2003. 14(6): 399-399.

74. Kang, M., Fedkiw R.P, A Boundary Condition Capturing Method for Multiphase Incompressible Flow. J. Sci. Comput. 2002. 15(323-360.

75. Kelly, S. G. D., Computational Fluid Dynamics Insights in the Design of Mechanical Heart Valves. Artificial Organs. 2002. 26(7): 608-613.

76. Kelly, S. G. D., Verdonck, P. R., Vierendeels, J. A. M., Riemslagh, K., Dick, E. and Van Nooten, G. G., A Three-Dimensional Analysis of Flow in the Pivot Regions of an Ats Bileaflet Valve. International Journal of Artificial Organs. 1999. 22(11): 754-763.

77. Kessler, D. A., Koplik, J. and Levine, H., Pattern Selection in Fingered Growth Phenomena. Advances in Physics. 1988. 37(3): 255-339.

78. Kim, J., Kim, D. and Choi, H., An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. Journal of Computational Physics. 2001. 171(1): 132-150.

79. Kim, Y. T., Goldenfeld, N. and Dantzig, J., Computation of Dendritic Microstructures Using a Level Set Method. Physical Review E. 2000. 62(2): 2471-2474.

80. King, M. J., Corden, J., David, T. and Fisher, J., A Three-Dimensional, Time-Dependent Analysis of Flow through a Bileaflet Mechanical Heart Valve: Comparison of Experimental and Numerical Results. Journal of Biomechanics. 1996. 29(5): 609-618.

81. King, M. J., David, T. and Fisher, J., Three-Dimensional Study of the Effect of Two Leaflet Opening Angles on the Time-Dependent Flow through a Bileaflet Mechanical Heart Valve. Medical Engineering & Physics. 1997. 19(3): 235-241.

82. Koumotsakos, P. and Leonard, A., High-Resolution Simulations of the Flow around an Impulsively Started Cylinder Using Vortex Method. Journal of Fluid Mechanics. 1995. 196(1-38):

83. Lai, M. C. and Li, Z. L., A Remark on Jump Conditions for the Three-Dimensional Navier-Stokes Equations Involving an Immersed Moving Membrane. Applied Mathematics Letters. 2001. 14(2): 149-154.

84. Lai, M. C. and Peskin, C. S., An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. Journal of Computational Physics. 2000. 160(2): 705-719.

85. Lai, Y. G., Chandran, K. B. and Lemmon, J., A Numerical Simulation of Mechanical Heart Valve Closure Fluid Dynamics. Journal of Biomechanics. 2002. 35(7): 881-892.

86. Lan, Z. L., Taylor, V. E. and Bryan, G., A Novel Dynamic Load Balancing Scheme for Parallel Systems. Journal of Parallel and Distributed Computing. 2002. 62(12): 1763-1781.

87. Lee, L. and Leveque, R. J., An Immersed Interface Method for Incompressible Navier-Stokes Equations. Siam Journal on Scientific Computing. 2003. 25(3): 832-856.

88. Leo, H. L., He, Z. M., Ellis, J. T. and Yoganathan, A. P., Microflow Fields in the Hinge Region of the Carbomedics Bileaflet Mechanical Heart Valve Design. Journal of Thoracic and Cardiovascular Surgery. 2002. 124(3): 561-574.

89. Leveque, R. J. and Li, Z. L., The Immersed Interface Method for Elliptic-Equations with Discontinuous Coefficients and Singular Sources. Siam Journal on Numerical Analysis. 1994. 31(4): 1019-1044.

90. Leveque, R. J. and Li, Z. L., The Immersed Interface Method for Elliptic-Equations with Discontinuous Coefficients and Singular Sources (Vol 31, Pg 1019, 1994). Siam Journal on Numerical Analysis. 1995. 32(5): 1704-1704.

91. Li, Z. L. and Lai, M. C., The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces. Journal of Computational Physics. 2001. 171(2): 822-842.

92. Liu, H., Krishnan, S., Marella, S. and Udaykumar, H. S., Sharp Interface Cartesian Grid Method Ii: A Technique for Simulating Droplet Interactions with Surfaces of Arbitrary Shape. Journal of Computational Physics. 2005. 210(1): 32-54.

93. Liu, X. D., Fedkiw, R. P. and Kang, M. J., A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains. Journal of Computational Physics. 2000. 160(1): 151-178.

94. Losasso, F., Gibou, F. and Fedkiw, R., Simulating Water and Smoke with an Octree Data Structure. Acm Transactions on Graphics. 2004. 23(3): 457-462.

95. Manning, K. B., Kini, V., Fontaine, A. A., Deutsch, S. and Tarbell, J. M., Regurgitant Flow Field Characteristics of the St. Jude Bileaflet Mechanical Heart Valve under Physiologic Pulsatile Flow Using Particle Image Velocimetry. Artificial Organs. 2003. 27(9): 840-846.

96. Marella, S., Krishnan, S., Liu, H. and Udaykumar, H. S., Sharp Interface Cartesian Grid Method I: An Easily Implemented Technique for 3d Moving Boundary Computations. Journal of Computational Physics. 2005. 210(1): 1-31.

97. Marella, S. V. and Udaykumar, H. S., Computational Analysis of the Deformability of Leukocytes Modeled with Viscous and Elastic Structural Components. Physics of Fluids. 2004. 16(2): 244-264.

98. Marshall, J. S., A Model of Heavy Particle Dispersion by Organized Vortex Structures Wrapped around a Columnar Vortex Core. Physics of Fluids. 1998. 10(12): 3236-3238.

99. Minyard, T. and Kallinderis, Y., Octree Partitioning of Hybrid Grids for Parallel Adaptive Viscous Flow Simulations. International Journal for Numerical Methods in Fluids. 1998. 26(1): 57-78.

100. Minyard, T. and Kallinderis, Y., Parallel Load Balancing for Dynamic Execution Environments. Computer Methods in Applied Mechanics and Engineering. 2000. 189(4): 1295-1309.

101. Monaghan, J. J., Simulating Free-Surface Flows with Sph. Journal of Computational Physics. 1994. 110(2): 399-406.

102. Monaghan, J. J., Sph and Riemann Solvers. Journal of Computational Physics. 1997. 136(2): 298-307.

103. Mullins, W. W. and Sekerka, R. F., Stability of a Planar Interface During Solidification of a Dilute Binary Alloy. Journal of Appliued Physics. 1964. 35(2): 444-451.

104. Nguyen, D. Q., Fedkiw, R. P. and Kang, M., A Boundary Condition Capturing Method for Incompressible Flame Discontinuities. Journal of Computational Physics. 2001. 172(1): 71-98.

105. Osher, S. and Sethian, J. A., Fronts Propagating with Curvature-Dependent Speed - Algorithms Based on Hamilton-Jacobi Formulations. Journal of Computational Physics. 1988. 79(1): 12-49.

106. Patankar, N. A., Singh, P., Joseph, D. D., Glowinski, R. and Pan, T. W., A New Formulation of the Distributed Lagrange Multiplier/Fictitious Domain Method for Particulate Flows. International Journal of Multiphase Flow. 2000. 26(9): 1509-1524.

107. Peskin, C. S., Numerical Analysis of Blood Flow in the Heart. J. Comp. Phys. 1977. 25(220-243):

108. Peskin, C. S. and Mcqueen, D. M., Cardiac Fluid-Dynamics. Critical Reviews in Biomedical Engineering. 1992. 20(5-6): 451-&.

109. Peskin, C. S. and Printz, B. F., Improved Volume Conservation in the Computation of Flows with Immersed Elastic Boundaries. Journal of Computational Physics. 1993. 105(1): 33-46.

110. Popinet, S., Gerris: A Tree-Based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries. Journal of Computational Physics. 2003. 190(2): 572-600.

111. Provatas, N., Goldenfeld, N. and Dantzig, J., Efficient Computation of Dendritic Microstructures Using Adaptive Mesh Refinement. Physical Review Letters. 1998. 80(15): 3308-3311.

112. Roma, A. M., Peskin, C. S. and Berger, M. J., An Adaptive Version of the Immersed Boundary Method. Journal of Computational Physics. 1999. 153(2): 509-534.

113. Rosenfeld, M., Avrahami, I. and Einav, S., Unsteady Effects on the Flow across Tilting Disk Valves. Journal of Biomechanical Engineering-Transactions of the Asme. 2002. 124(1): 21-29.

114. Saffman, P. G., The Lift on a Small Sphere in a Slow Shear Flow. Journal of Fluid Mechanics. 1965. 22(385.

115. Saffman, P. G., Corrigendum to "the Lift Force on a Small Sphere in a Slow Shear Flow. Journal of Fluid Mechanics. 1968. 31(624.

116. Sethian, J. A., Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. Journal of Computational Physics. 2001. 169(2): 503-555.

117. Sethian, J. A. and Smereka, P., Level Set Methods for Fluid Interfaces. Annual Review of Fluid Mechanics. 2003. 35(341-372.

118. Shi, Y. B., Zhao, Y., Yeo, T. J. H. and Hwang, N. H. C., Numerical Simulation of Opening Process in a Bileaflet Mechanical Heart Valve under Pulsatile Flow Condition. Journal of Heart Valve Disease. 2003. 12(2): 245-255.

119. Shipkowitz, T., Ambrus, J., Kurk, J. and Wickramasinghe, K., Evaluation Technique for Bileaflet Mechanical Valves. Journal of Heart Valve Disease. 2002. 11(2): 275-282.

120. Simon, H. A., Leo, H. L., Carberry, J. and Yoganathan, A. P., Comparison of the Hinge Flow Fields of Two Bileaflet Mechanical Heart Valves under Aortic and Mitral Conditions. Annals of Biomedical Engineering. 2004. 32(12): 1607-1617.

121. Sochnikov, V. and Efrima, S., Level Set Calculations of the Evolution of Boundaries on a Dynamically Adaptive Grid. International Journal for Numerical Methods in Engineering. 2003. 56(13): 1913-1929.

122. Sorensen, E. N., Burgreen, G. W., Wagner, W. R. and Antaki, J. F., Computational Simulation of Platelet Deposition and Activation: I. Model Development and Properties. Annals of Biomedical Engineering. 1999. 27(4): 436-448.

123. Sorensen, E. N., Burgreen, G. W., Wagner, W. R. and Antaki, J. F., Computational Simulation of Platelet Deposition and Activation: Ii. Results for Poiseuille Flow over Collagen. Annals of Biomedical Engineering. 1999. 27(4): 449-458.

124. Stockie, J. M. and Wetton, B. R., Analysis of Stiffness in the Immersed Boundary Method and Implications for Time-Stepping Schemes. Journal of Computational Physics. 1999. 154(1): 41-64.

125. Strain, J., Fast Tree-Based Redistancing for Level Set Computations. Journal of Computational Physics. 1999. 152(2): 664-686.

126. Strain, J., Semi-Lagrangian Methods for Level Set Equations. Journal of Computational Physics. 1999. 151(2): 498-533.

127. Strain, J., Tree Methods for Moving Interfaces. Journal of Computational Physics. 1999. 151(2): 616-648.

128. Strain, J., A Fast Modular Semi-Lagrangian Method for Moving Interfaces. Journal of Computational Physics. 2000. 161(2): 512-536.

129. Strain, J., A Fast Semi-Lagrangian Contouring Method for Moving Interfaces. Journal of Computational Physics. 2001. 170(1): 373-394.

130. Sukumar, N., Chopp, D. L., Moes, N. and Belytschko, T., Modeling Holes and Inclusions by Level Sets in the Extended Finite-Element Method. Computer Methods in Applied Mechanics and Engineering. 2001. 190(46-47): 6183-6200.

131. Sukumar, N., Moes, N., Moran, B. and Belytschko, T., Extended Finite Element Method for Three-Dimensional Crack Modelling. International Journal for Numerical Methods in Engineering. 2000. 48(11): 1549-1570.

132. Sussman, M. and Fatemi, E., An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow. Siam Journal on Scientific Computing. 1999. 20(4): 1165-1191.

133. Sussman, M., Fatemi, E., Smereka, P. and Osher, S., An Improved Level Set Method for Incompressible Two-Phase Flows. Computers & Fluids. 1998. 27(5-6): 663-680.

134. Tambasco, M. and Steinman, D. A., Path-Dependent Hemodynamics of the Stenosed Carotid Bifurcation. Annals of Biomedical Engineering. 2003. 31(9): 1054-1065.

135. Tong, X., Beckermann, C., Karma, A. and Li, Q., Phase-Field Simulations of Dendritic Crystal Growth in a Forced Flow. Physical Review E. 2001. 63(6): -.

136. Trompert, R., Local-Uniform-Grid Refinement and Transport in Heterogeneous Porous-Media. Advances in Water Resources. 1993. 16(5): 293-304.

137. Trompert, R. A. and Verwer, J. G., A Static-Regridding Method for 2-Dimensional Parabolic Partial-Differential Equations. Applied Numerical Mathematics. 1991. 8(1): 65-90.

138. Tryggvason, G., Bunner, B., Esmaeeli, A. and Al-Rawahi, N., Computations of Multiphase Flows. Advances in Applied Mechanics, Vol 39. 2003. 39(81-120.

139. Tseng, Y. H. and Ferziger, J. H., A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry. Journal of Computational Physics. 2003. 192(2): 593-623.

140. Udaykumar, H. S. and Mao, L., Sharp-Interface Simulation of Dendritic Solidification of Solutions. International Journal of Heat and Mass Transfer. 2002. 45(24): 4793-4808.

141. Udaykumar, H. S., Mao, L. and Mittal, R., A Finite-Volume Sharp Interface Scheme for Dendritic Growth Simulations: Comparison with Microscopic Solvability Theory. Numerical Heat Transfer Part B-Fundamentals. 2002. 42(5): 389-409.

142. Udaykumar, H. S., Marella, S. and Krishnan, S., Sharp-Interface Simulation of Dendritic Growth with Convection: Benchmarks. International Journal of Heat and Mass Transfer. 2003. 46(14): 2615-2627.

143. Udaykumar, H. S., Mittal, R. and Rampunggoon, P., Interface Tracking Finite Volume Method for Complex Solid-Fluid Interactions on Fixed Meshes. Communications in Numerical Methods in Engineering. 2002. 18(2): 89-97.

144. Udaykumar, H. S., Mittal, R., Rampunggoon, P. and Khanna, A., A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries. Journal of Computational Physics. 2001. 174(1): 345-380.

145. Udaykumar, H. S., Mittal, R. and Shyy, W., Computation of Solid-Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids. J. Comput. Phys. 1999. 153(2): 535-574.

146. Udaykumar, H. S., Mittal, R. and Shyy, W., Computation of Solid-Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids. Journal of Computational Physics. 1999. 153(2): 535-574.

147. Venuturumilli, R. and Chen, L. D., Numerical Simulation Using Adaptive Mesh Refinement for Laminar Jet Diffusion Flames. Numerical Heat Transfer Part B-Fundamentals. 2004. 46(2): 101-120.

148. Wagner, C., Introduction to Algebraic Multigrid. Manuscript. 1998.

149. Wang, J. H., Yao, H., Lim, C. J., Zhao, Y., Yeo, T. J. H. and Hwang, N. H. C., Computational Fluid Dynamics Study of a Protruded-Hinge Bileaflet Mechanical Heart Valve. Journal of Heart Valve Disease. 2001. 10(2): 254-262.

150. Wang, J. P., Borthwick, A. G. L. and Taylor, R. E., Finite-Volume-Type Vof Method on Dynamically Adaptive Quadtree Grids. International Journal for Numerical Methods in Fluids. 2004. 45(5): 485-508.

151. Wang, X. D. and Liu, W. K., Extended Immersed Boundary Method Using Fem and Rkpm. Computer Methods in Applied Mechanics and Engineering. 2004. 193(12-14): 1305-1321.

152. Wang, Z. J., A Quadtree-Based Adaptive Cartesian/Quad Grid Flow Solver for Navier-Stokes Equations. Computers & Fluids. 1998. 27(4): 529-549.

153. Wang, Z. J., A Fast Nested Multi-Grid Viscous Flow Solver for Adaptive Cartesian/Quad Grids. International Journal for Numerical Methods in Fluids. 2000. 33(5): 657-680.

154. Webster, R., An Algebraic Multigrid Solver for Navier-Stokes Problems. International Journal for Numerical Methods in Fluids. 1994. 18(8): 761-780.

155. Webster, R., An Algebraic Multigrid Solver for Navier-Stokes Problems in the Discrete Second-Order Approximation. International Journal for Numerical Methods in Fluids. 1996. 22(11): 1103-1123.

156. Woo, Y. R., Williams, F. P., Faughan, P. D. and Yoganathan, A. P., Pulsatile Flow Visualization Studies with Aortic and Mitral Mechanical Valve Prostheses. Chemical Engineering Communications. 1986. 47(1-3): 23-48.

157. Woo, Y. R. and Yoganathan, A. P., Pulsatile Flow Velocity and Shear-Stress Measurements on the St-Jude Bileaflet Valve Prosthesis. Scandinavian Journal of Thoracic and Cardiovascular Surgery. 1986. 20(1): 20-28.

158. Yang, Y., Garvin, J. W. and Udaykumar, H. S., Sharp Interface Simulation of Interaction of a Growing Dendrite with a Stationary Solid Particle. International Journal of Heat and Mass Transfer. 2005. 48(25-26): 5270-5283.

159. Yang, Y. and Udaykumar, H. S., Sharp Interface Cartesian Grid Method Iii: Solidification of Pure Materials and Binary Solutions. Journal of Computational Physics. 2005. 210(1): 55-74.

160. Ye, T., Mittal, R., Udaykumar, H. S. and Shyy, W., An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. Journal of Computational Physics. 1999. 156(2): 209-240.

161. Yin, W., Alemu, Y., Affeld, K., Jesty, J. and Bluestein, D., Flow-Induced Platelet Activation in Bileaflet and Monoleaflet Mechanical Heart Valves. Annals of Biomedical Engineering. 2004. 32(8): 1058-1066.

162. Yiu, K. F. C., Greaves, D. M., Cruz, S., Saalehi, A. and Borthwick, A. G. L., Quadtree Grid Generation: Information Handling, Boundary Fitting and Cfd Applications. Computers & Fluids. 1996. 25(8): 759-769.

163. Yoganathan, A. P., Chandran, K. B. and Sotiropoulos, F., Flow in Prosthetic Heart Valves: State-of-the-Art and Future Directions. Annals of Biomedical Engineering. 2005. 33(12): 1689-1694.

164. Yoganathan, A. P., He, Z. M. and Jones, S. C., Fluid Mechanics of Heart Valves. Annual Review of Biomedical Engineering. 2004. 6(331-362.

165. Zang, Y., Street, R. L. and Koseff, J. R., A Non-Staggered Grid, Fractional Step Method for Time-Dependent Incompressible Navier-Stokes Equations in Curvilinear Coordinates. Journal of Computational Physics. 1994. 114(1): 18-33.

166. Zhang, J., Sun, H. W. and Zhao, J. J., High Order Compact Scheme with Multigrid Local Mesh Refinement Procedure for Convection Diffusion Problems. Computer Methods in Applied Mechanics and Engineering. 2002. 191(41-42): 4661-4674.