ABSTRACT

Title of Thesis: NUMERICAL SIMULATION OF LIQUID - VAPOR INTERFACES IN THE SHARP INTERFACE LIMIT

Vamsee K. Yerramilli, Master of Science, 2005

Thesis directed by: Dr. Elias Balaras
Department of Mechanical Engineering

A sharp interface method to capture the exact dynamics of liquid vapor interfaces, has been developed. The two-dimensional, unsteady Navier-Stokes equations are solved using a standard finite-difference formulation on a fixed Cartesian grid, while the interface is tracked dynamically using a linear front tracking method that utilizes marker particles. To capture the exact physics of the interface, a sharp interface technique is used, where all interfacial and material properties like surface tension, density and viscosity are directly evaluated based on the exact location of the interface. In addition, to enforce the appropriate boundary conditions for the pressure, a special variable coefficient pressure Poisson solver with modified coefficients has been developed. Single and multiple vapor bubbles rising through a quiescent liquid, are simulated. Bubble merger and fracture processes are also investigated. The proposed method was not only found to be more accurate, but also more efficient when compared diffused-interface methods.

NUMERICAL SIMULATION OF LIQUID - VAPOR INTERFACES IN THE SHARP INTERFACE LIMIT

by

Vamsee K. Yerramilli

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2005

Advisory Committee:

Dr. Jungho Kim, Advisor
Dr. Elias Balaras, Co-Advisor
Professor Arnaud Trouve

This work is dedicated to my father and mother.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisors Dr. Jungho Kim and Dr. Elias Balaras. Without their guidance and expert advice, this work would have been impossible. I would also like to thank my lab member, Jianming Yang, without whose help and advice, this work would not have been complete within time.

I would also like to thank my parents, Visweswara Rao Yerramilli and Ratna S. Yerramilli for their support and encouragement. I would also like to thank my sister Srividya Chaganti and her family for keeping me motivated throughout my graduate study.

Also, I would like to acknowledge the support of my roommates Arun K. Kota and Arun I. Jose. Also working along with my other lab members Nikolaous Beratlis, Antonio Cristallo and Vincenzo Pezza has made my work more enjoyable.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Understanding Boiling Phenomena

Boiling is a very complex phenomena with many mechanisms, which are yet to be fully understood. For example nucleation and bubble growth under subcooling or superheating conditions could not be studied completely due to lack of well resolved time and space information of a growing bubble as recently pointed out by Myers et al. [1]. There has been a great deal of interest in the research community to understand these mechanisms fully, to better the use of boiling in practical applications like cooling of electronics through spray cooling or flow boiling. The boiling process can result in tremendous heat transfer since it is a phase change process. A better understanding of this process and the different boiling regimes could result in advanced cooling technologies which can increase the processing capacities of future high power electronic devices.

In the beginning large scale experiments were used to acquire data from which empirical correlations were developed through the use of dimensionless groups [2, 3, 4]. However these models have been limited in their application because the data were generated for specific cases of heating conditions, fluid-surface combinations and fluid properties. Since the mechanisms of boiling, change with these combinations, it has caused development of several, often conflicting, theories to

explain these mechanisms.

Alternatively, analytical models were used to develop heat and mass transfer models of the boiling process. Forster [5] developed a model to understand the heat transfer due to microlayer evaporation and transient conduction near nucleation sites, while Zuber [6] developed a model to evaluate the maximum heat flux on infinite flat plates. These models were based on exclusive analysis of specific mechanisms and do not consider the multiple mechanisms which often occur simultaneously during boiling. These models have been typically used to approximate the amount of heat transfer in heterogeneous nucleate boiling.

It is generally acknowledged in the literature that the understanding of boiling process is still incomplete due to the lack of adequate tools to investigate the physics behind the process [8]. Thus it has become necessary to develop numerical tools to study the boiling process using mass, momentum and energy conservation equations. It is necessary for such a model to include condensation or evaporation. A numerical model for boiling flows is incomplete unless it includes surface tension, latent heat, interface mass transfer, and discontinuous material properties, thus making development of numerical methods for boiling flows very challenging. Juric and Tryggvason [7] developed a 2D model to capture the boiling process through surface tension and latent heat terms. Although such models capture the boiling process, they have later been shown to cause parasitic currents due to numerical smearing [9]. Parasitic currents are artificial velocities around the interface often found in the diffused interface methods. They are produced due to inaccurate representation of the surface tension or the location of the interface. In this present work, a numeri-

cal method has been developed to simulate two-phase flows and which can then be extended to solve for boiling flows with phase change.

1.2 Numerical Methods in Two Phase Flows

There have been several approaches towards numerical simulation of two phase flows. They can be broadly categorized as direct or indirect methods, based on the mathematical model to solve the problem. In the direct method, the exact interface dynamics are captured with sharp discontinuity in material properties and pressure. In the indirect approach, some model is used to represent the interface with finite thickness. The material properties are smeared onto the grid and the surface tension force is distributed over a few grid cells. This type of formulation often leads to parasitic currents around the interface and causes area losses in representing the interface.

Numerical methods in multiphase flows can also be classified as Eulerian, Lagrangian, or mixed scheme based on the numerical scheme used to solve the governing equations. In the Lagrangian method the grid follows the fluid, whereas in Eulerian method the grid is stationary. In the mixed method, the fluids are considered on a fixed stationary grid, but the interface is represented using a front which moves along with the local fluid velocity. A comprehensive literature survey on the different numerical methods for two phase flows, can be found in the article by Tryggvason et al. [9].

Ryskin and Leal [10] used a direct approach with boundary fitted grids for

each phase. They simulated buoyancy driven two dimensional axisymmetric inviscid bubbles rising through a quiescent liquid. They presented results for various ranges of Reynolds and Weber numbers. Later Dandy and Leal [11] developed numerical solutions for buoyancy driven deformable viscous drop through a quiescent liquid at intermediate Reynolds numbers. These results were later used to validate more modern approaches towards numerical simulations of vapor bubbles.

In both these cases, a Lagrangian method was used to solve this problem directly. They used separate boundary fitted grids, one for each fluid, and solved the governing equations for both fluids independently. The boundary conditions near the interface were then matched by an iterative technique.

Boundary fitted grids cannot, however, be used for more general cases. The main challenge in such an approach is the implementation of the interface boundary condition. Since the boundary conditions were implemented iteratively, as a continuous shear stress condition at the interface, these methods have been limited to very simple applications. Although they provide accurate results for the cases which can be solved, they cannot be used to simulate complex three dimensional bubbles, owing to the computational cost and complexity of the simulation.

Ye et al. [8] adopted an immersed boundary method to capture the interface in a cut cell approach. They also approached the problem directly, but instead of using boundary fitted grid, they used a mixed Eulerian-Lagrangian method by tracking a moving interface on a fixed grid. However their approach is also computationally very expensive because the governing equations for the two fluids are solved separately. They use iterative methods to match the boundary conditions at

the interface between the two fluids.

Kang et al. [12] developed a Ghost Fluid Method (GFM) to simulate multiphase flows while treating the interface as sharp. They developed a modified pressure Poisson equation, in the context of a fractional step method, to account for the discontinuity arising due to surface tension and viscous forces [13]. They used level set to represent the interface and compared their results with a δ-function approach. In their simulation however, there was significant area loss, causing loss of bubble/droplet mass and parasitic currents in the flow field. With the sharp interface method their area loss was 17.23%.

Among the more popular approaches towards this problem, is the indirect method. In the indirect method, a *model* is used to represent the interface and to smoothen the material properties. In such models, the first step is to represent the interface using a front or by a level set. Once the two phases are identified, they are dynamically tracked or their location is indirectly captured as they evolve during the flow. Brackbill et al. [14] developed a Lagragian inviscid model with color function to represent the interface. This CSF (continuum surface force) model distributes the interface over a few grid cells. This smoothing of the interface creates unphysical currents as discussed by other researchers later.

Sussman et al. [15] developed a level set method to simulate incompressible two phase flow. They have been able to represent a sharp interface using level set and solved the Navier-Stokes equations using the standard projection method, while density and surface tension were smoothed around the interface. They have been able to simulate vapor bubbles and liquid droplets for large density ratio and surface

tension. Later they [16] presented an improved level set method for two phase flows, where they were able to capture merger or break up of bubbles in flows with high Reynolds number.

Tryggvason et al. [9] published a landmark paper proposing a hybrid of front tracking and front capturing method for computations of multiphase flows. They solved a single set of governing equations for the entire flow field consisting of both the phases by representing surface tension as an external force field in the momentum equations and obtaining the pressure jump condition implicitly. However their simulation was limited to solution of mass and momentum equations only. It was pointed out that high density ratios caused convergence problems in the pressure Poisson solver, while high surface tension caused parasitic currents or unphysical velocities.

Francois and Shyy [17] developed a front tracking method using marker points on the interface. They used an indirect approach to solve the problem, by distributing the interface over few grid cells. They also solved the problem with energy transport. They studied the effect of property ratio on the evolution of a buoyancy driven axisymmetric bubble. They were able to simulate bubbles under various test conditions, and validated their results with many cases from the literature. They used a multigrid method to reduce the computational cost, while tracking the interface in a Lagrangian way. In their two part article, first they presented their model and validated with several standard test cases, then they extended their study to much more advanced cases of two phase flows. Their study was limited to mass, momentum and energy transport without any interfacial mass transfer.

To conclude the numerical simulations in two phase flows, there are two approaches towards solving this problem. First, the direct method, which has been used with boundary fitted grids, is computationally expensive and quite difficult to extended to complex applications. This limitation is primarily due to the iterative techniques used in matching the interfacial boundary condition and trying to solve two different sets of governing equations for the two fluids. Some modern applications of the direct method have caused area losses in representing the bubble/droplet. In the indirect methods, the density, viscosity and surface tension are distributed over a few grid cells. This treatment of interface with finite thickness has been shown to produce unphysical velocities or parasitic currents in the flow field.

1.3 Treatment of the Interface

Numerical simulations of two phase flows have a characteristic and fundamental problem of representing the interface. Any inaccuracy in representing the interface can cause severe problems with surface tension distribution, mass conservation or numerical convergence. Especially in the distributed interface method, any anisotropy in representing surface tension of the interface can cause high parasitic currents around the interface. There are primarily two approaches to represent the interface, front tracking and front capturing method.

In the front capturing scheme, the location of the interface is captured indirectly through solution of the density field or some other indicator function. This

method results in errors in interface representation, often leading to parasitic currents around the interface. It also requires very well refined grids thus limiting the simulation to simple applications, especially for unsteady flows. Front capturing schemes have been used by Tryggvason et al. [9] in some combination with front tracking scheme. Sussman et al. [15] have used level set to represent the interface in the context of a front capturing scheme. A good discussion on this has been presented by Unverdi and Tryggvason [18], when they presented a front tracking method to simulate viscous, incompressible multiphase flows.

In tracking schemes, the location of the interface or volume of each fluid is explicitly tracked using marker points or computational elements. This scheme is very cost effective and accurate, since numerical diffusion is completely eliminated and there is no need for highly resolved grid. Within the tracking methods, there are volume tracking and interface (front) tracking methods. Marker-And-Cell (MAC) and Volume-Of-Fluid (VOF) methods come under volume tracking schemes while the current work is a front tracking method. The main disadvantage of this scheme is that when the interface deforms significantly, the error in curvature can be high. This is observed in cases when there is bubble merger, oscillating bubble or physical instabilities like Rayleigh-Taylor instability.

Another method of representing the interface is through level sets. Level set is a sign function which represents the distance from the interface - it is positive outside the interface, zero at the interface, and negative inside. Using level set to represent interface is by far the most popular method. It has been found to be quite easy to be used even when the shape of the interface is deformed. However, level

set function and its representation may often be computationally expensive.

Once the interface location is identified, in the indirect approach, the interface is distributed over a few cells, thus making the interface of finite thickness and enforcing the shear stress balance over a few cells.

At the interface, the tangential components of the shear stress must be continuous along the thickness of the interface. The normal component of the shear stress near the interface ensures that the pressure jump across the interface is proportional to the surface tension and the viscous forces. Thus, the accurate representation of interface is necessary to ensure continuous shear stress across the interface.

1.4 Objectives of the present work

It has been shown in the previous sections that numerical simulations of two phase flows using direct methods have been limited by their complexity and computational cost, since two separate sets of governing equations are used for the two fluids. There have been several studies with indirect methods, but such studies have been shown to result in unphysical velocities around the interface. Also in the indirect methods, there is a need for a highly resolved grid around the interface since the material properties are distributed around the interface.

In the present work, a direct method to simulate two phase flows is presented. A single set of governing equations are used for the entire flow field consisting of two fluids. A single grid is used for the entire flow field while the interface is represented as a front with linearly connected marker particles.

In order to validate the Navier-Stokes solver with immersed boundary method, first the indirect approach is adopted. The surface tension force and the material properties are distributed on to the fixed grid over a few grid cells around the interface. The results are validated with several analytical/numerical results.

Then a direct approach model is developed to represent a sharp interface with jump discontinuity in material properties. This representation is not only a more accurate representation of the actual physics but also computationally very efficient.

The two approaches are compared for their computational cost and efficiency. Order of accuracy studies are also performed to study the dependence of the results on the grid resolution. Several test cases are simulated with the sharp interface method with different Reynolds number, Bond (or Weber) number, density ratio and viscosity ratio. Bubble merger and fracture processes are also studied.

1.5 Outline of Thesis

In Chapter 2, the mathematical model and numerical scheme for the distributed interface method (DIM) are presented. Various schemes are studied for distribution of material properties onto the fixed grid. Continuity and momentum equations are analyzed in the presence of surface tension. A fractional step method is used in a third order Runge-Kutta scheme to solve the momentum equations numerically.

In Chapter 3, the sharp interface method (SIM) is presented. Surface tension is introduced as a jump discontinuity in the pressure Poisson equation. The devel-

opment and validation of the variable coefficient Poisson solver is then discussed. The modified Poisson equation is solved using standard iterative techniques. The validated pressure solver is then integrated into the Navier-Stokes solver in the fractional step method. Implementation of the immersed boundary method is described and mass conservation for the vapor bubble is enforced as an additional constraint to satisfy global and local continuity.

In Chapter 4, the results using both methods (DIM and SIM) are presented and discussed in detail. The code is validated with several benchmark problems to incorporate special conditions like variable viscosity, bubble evolution, etc. Order of accuracy, grid resolution, and convergence tests are presented. The effect of Reynolds number, Bond number, density ratio, and viscosity ratio is studied by comparing the current simulations with standard benchmark analytical and numerical solutions. Bubble merger and fracture are then studied.

In Chapter 5, conclusions are presented with a discussion on possible extension of this work.

Chapter 2

Distributed Interface Method

In this chapter the mathematical model and numerical scheme for the distributed interface method (DIM) are presented. Various schemes are studied for distribution of material properties onto the fixed grid. Continuity and momentum equations are analyzed in the presence of surface tension. A fractional step method is used in a third order Runge-Kutta scheme to solve the momentum equations numerically.

Consider a single vapor bubble rising through a liquid medium due to gravity. We use a single fluid formulation to represent the entire domain with one set of governing equations. The material properties, namely density and viscosity, are discontinuous across the interface, but this discontinuity is smoothed out by distributing the interface over a few cells. Also, the surface tension is also smoothed out and distributed onto the grid over a few cells. It is assumed that both fluids are Newtonian fluids and that the material properties are constant within each fluid phase.

The domain consisting of two fluids is illustrated in Fig. 2.1. The domain considered here is a 2D square on a Cartesian grid. The interface is represented by a linear front while the grid is stationary. The front is constructed from marker points which are distributed along the arc length of the interface. This distribution

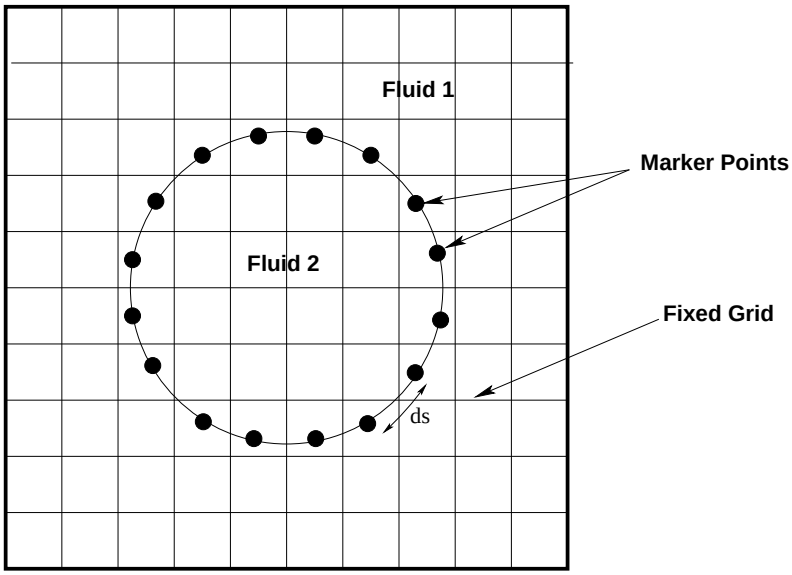


Figure 2.1: Computational domain with the two fluids.

is done carefully such that the arc length between the markers is comparable to the grid size. The two fluids are represented as fluid 1 and fluid 2, with the interface separating them.

In a distributed interface *model*, some strategy must be adopted to distribute the surface tension around the interfacial grid cells. The surface tension body force which is added as an external force in the momentum equations, must be concentrated around the interface and non-existent far from it. It is evident that some form of a δ-function must be used to represent this distribution.

We use a dimensional approach to represent the interface, that is, the interface is represented as a δ-function along each dimension. Thus, we arrive at a two dimensional δ-function to represent the interface. It must be noted here that this is a weak solution approach towards solving the differential governing equations and it is satisfied only in an integral sense. Since the material properties are discontinuous in

the actual case, we can consider them in the differential equations only as generalized functions and not as discontinuous functions.

The Heaviside function H is used to represent the gradual variation in the material properties. This function has a value of 1 inside the closed contour represented by the interface and 0 outside liquid. Along the thickness of the interface it varies smoothly. This Heaviside function can be represented using the dimensional approach as follows.

$$H(x, y, h) = \int_{A(h)} \delta(x - x')\delta(y - y')da' \tag{2.1}$$

Thus, the Heaviside function is an integral over an area around the contour defined by the interface location. The location (x, y) represents the grid point while (x', y') is the coordinate of the point on the interface closest to (x, y). Using this Heaviside function, the material properties can be represented as:

$$\rho(x, y, t) = \rho_i H(x, y, h) + \rho_o(1 - H(x, y, h)) \tag{2.2}$$

and,

$$\mu(x, y, t) = \mu_i H(x, y, h) + \mu_o(1 - H(x, y, h)) \tag{2.3}$$

Using the Heaviside formulation, the density distribution can be obtained as a contour plot, as shown in Figure 2.2.

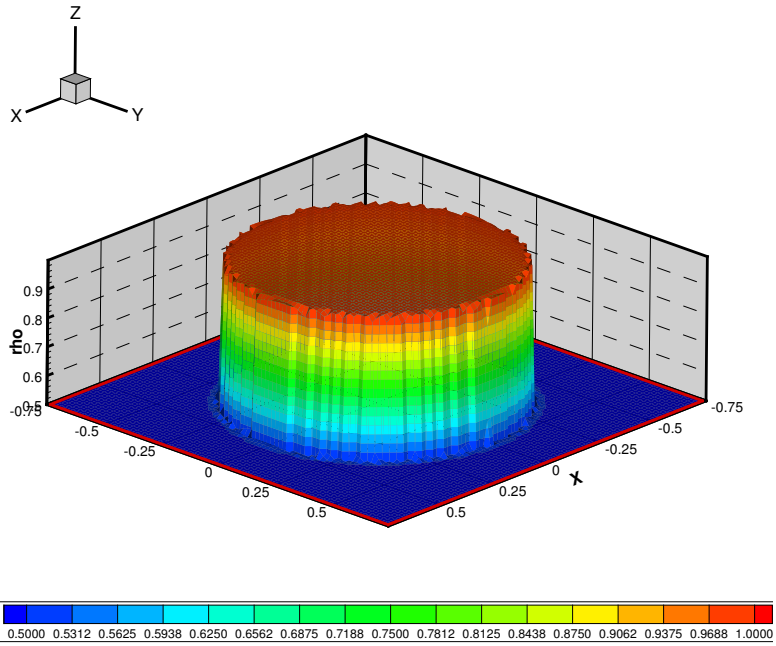


Figure 2.2: Density distribution using the Heaviside formulation for a liquid droplet.

2.1 Mathematical model and governing equations

2.1.1 Conservation of Mass

Within the domain considered, the conservation of mass is given by:

$$\frac{\partial \rho}{\partial t} + \nabla . \rho \mathbf{u} = 0. \tag{2.4}$$

Since each fluid is incompressible, with constant density we have:

$$\frac{D\rho}{Dt} = 0. \tag{2.5}$$

Thus the conservation of mass simplifies to the standard continuity equation as:

$$\nabla . \mathbf{u} = 0. \tag{2.6}$$

However it may be noted that when the interface is moving, the above continuity equation is not satisfied at the interface points. This is a direct consequence of the weak formulation, in which the differential equations are satisfied only in an integral sense. This necessitates that the mass of the bubble be conserved in addition to the global continuity. If the mass of the bubble is not conserved within the specified residual, then the shape of the bubble must be corrected. This procedure is explained in detail in a later section.

The viscosity is also constant, thereby yielding,

$$\frac{D\mu}{Dt} = 0. \tag{2.7}$$

Tryggvason et al. [9] have demonstrate that this equation for viscosity implicitly represents the interfacial jump condition for pressure as follows.

$$[-P + \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}]\mathbf{n} = \sigma\kappa\mathbf{n} \tag{2.8}$$

Where, [.] represents the jump across the interface. The physical interpretation of this last equation is that the normal components of the shear stress at the interface are balanced by the surface tension causing a continuous shear stress distribution. It must be noted here that the normal components of the velocity across the interface are continuous. In the case where there is mass transfer across the interface, this condition is violated and special treatment must be used to deal with such cases.

2.1.2 Conservation of Momentum

For conservation of momentum, we use the conservative form of the Navier-Stokes equations with variable density and viscosity. The external forces on the

system are the surface tension and the gravity. The general form of the Navier-Stokes equation for an incompressible Newtonian fluid can be written as:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla . \rho \mathbf{u}\mathbf{u} = -\nabla p + \nabla . \tau + \rho \mathbf{g} + \mathbf{F}_{St,v} \tag{2.9}$$

the shear stress tensor being

$$\tau = \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}) \tag{2.10}$$

and the surface tension body force represented as:

$$\mathbf{F}_{St,v} = \int \sigma \kappa' \mathbf{n}' \delta^{\beta}(x - x') ds' \tag{2.11}$$

Thus the set of governing equations can be summarized as:

$$\begin{aligned} \nabla . \mathbf{u} &= 0. \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla . \rho \mathbf{u}\mathbf{u} &= -\nabla p + \nabla . \tau + \rho \mathbf{g} + \mathbf{F}_{St,v} \end{aligned} \tag{2.12}$$

This single set of Navier-Stokes equations represents two fluids where the density and viscosity are discontinuous spatial variables which have been smoothed out. In this equation $\mathbf{u}$, represents the velocity, p the pressure, and $\mathbf{g}$ the acceleration due to gravity. The surface tension is represented using the dimensional representation as a product of δ functions, β=2 or 3 for two or three dimensional problem. Surface tension volume force is represented by $F_{St,v}$.

Using the following non-dimensional parameters the above equation can be normalized.

$$\mathbf{x}^* = \frac{\mathbf{x}}{D} \tag{2.13}$$

$$\mathbf{u}^* = \frac{\mathbf{u}}{U} \tag{2.14}$$

$$\mathbf{g}^* = \frac{\mathbf{g}}{g_{ref}} \tag{2.15}$$

$$\rho^* = \frac{\rho}{\rho_{ref}} \tag{2.16}$$

$$\mu^* = \frac{\mu}{\mu_{ref}} \tag{2.17}$$

$$\sigma^* = \frac{\sigma}{\sigma_{ref}} \tag{2.18}$$

And the normalized equations can be written as follows. Here the superscript has been dropped for convenience.

$$\begin{aligned} \nabla.\mathbf{u} &= 0. \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla.\rho\mathbf{u}\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla.\tau + \frac{1}{Fr}\rho\mathbf{g} + \frac{1}{We}\mathbf{F}_{St,v} \end{aligned} \tag{2.19}$$

The non-dimensional numbers governing the behavior of these equations are:

$$\begin{aligned} Re &= \frac{\rho_{ref} U D}{\mu_{ref}} \\ We &= \frac{\rho_{ref} U^2 D}{\sigma_{ref}} \\ Fr &= \frac{U^2}{g_{ref} D} \end{aligned} \tag{2.20}$$

2.1.3 Initial and boundary conditions

The above set of governing equations are solved in a domain which is at least $1.5D \times 1.5D$, where D is the diameter of the bubble. Higher domain sizes have also been used to avoid the boundary condition effects seeping into the interface. Initially, all the velocity components and pressure, are assumed zero. The density and viscosity are assumed to be that of the heavier (ambient) fluid. The interface is assumed to appear instantaneously at this moment and then two fluids exist from

the next time step. The boundaries of the domain are assumed to be homogeneous Neumann boundaries, meaning that the fluids are enclosed inside a 2D chamber, with the walls at rest.

Apart from this global boundary conditions, it is necessary to match the interface boundary conditions. Near the interface, the velocities must be continuous to ensure the continuity of shear stress. However there must be a jump discontinuity in the pressure, which must be accounted through surface tension and viscosity. In DIM, this interfacial boundary condition is obtained implicitly since the surface tension is modeled as an external body force in the momentum equations. The actual interfacial boundary condition that must be imposed is:

$$[-P + \mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}]\mathbf{n} = \sigma\kappa\mathbf{n} \tag{2.21}$$

In a discrete sense this can be expanded as follows:

$$\begin{aligned} p_l - p_v + \frac{1}{We}\kappa &= \frac{1}{Re}\left\{\left(\frac{\partial u_n}{\partial n}\right)_l - \left(\frac{\mu_v}{\mu_l}\right)\left(\frac{\partial u_n}{\partial n}\right)_v\right\} \\ &- \left[(u_n)_l - (u_n)_{int}\right](u_n)_t \\ &+ \left(\frac{\rho_v}{\rho_l}\right)\left[(u_n)_v - (u_n)_{int}\right](u_n)_v \end{aligned} \tag{2.22}$$

2.1.4 Treatment of density, viscosity and surface tension

In the distributed interface model, there are several methods to smoothen the material properties and the surface tension. The distribution of density and viscosity, using the Heaviside function, has been discussed earlier in this chapter. The surface tension can be smoothened using Peskin's method [19] or Tryggvason's method [9]. In Peskin's method the distribution stencil is larger thereby causing the

interface to be thicker than Tryggvason's method. Peskin's scheme may be written as follows.

$$d_r = \begin{cases} (1/4h)(1 + cos(\pi r/2h)) & \text{if } |r| < 2h \\ 0 & \text{if } |r| \geq 2h \end{cases} \tag{2.23}$$

Tryggvason's [9] scheme can be expressed as:

$$d_r = \begin{cases} (h - |r|)/h & \text{if } |r| < h \\ 0 & \text{if } |r| \geq h \end{cases} \tag{2.24}$$

In the above equations, $d(r)$ is the weighting function, which depends on the distance, r, to the closest grid point from the marker point. The grid size is represented by h. This is a dimensional representation, that is a two dimensional weighting function must be constructed by multiplying the functions evaluated along each dimension. Thus while evaluating the smoothing from a marker point (x_{ib}, y_{ib}), the weighting function for a grid point (i, j), can be represented as:

$$w_{ij} = d(x_{ib} - i\Delta x)d(y_{ib} - j\Delta y) \tag{2.25}$$

Once the weighting function is evaluated, the surface tension must be converted into a volume force and weighted at the grid points. This process can be summarized as:

$$F(i,j)_{St,v,x} = \sum_{ib} w_{ij}\sigma\kappa_{ib}\frac{\Delta s}{\Delta x^2} \tag{2.26}$$

Here $F(i,j)_{St,v,x}$ is the surface tension volume force component evaluated and located at the u-velocity nodes. w_{ij} is the weighting function associated with the grid point, which is evaluated based on the Peskin's scheme or Tryggvason's scheme. σ and κ_{ib} are the surface tension and curvature at the marker point. Δs and Δx are the arc length and the grid size parameters respectively. The surface tension is an

additive force and all the contributions of neighboring marker points are added up. Hence ib is the summation index of the marker point which falls within the distribution stencil (Tryggvason's or Peskin's) from the grid point (i, j). In the current work, Tryggvason's scheme has been used, since it is a very compact stencil. Figure 2.3 illustrates this distribution scheme.

The surface tension volume force may be plotted as a vector plot to indicate that this force is responsible for holding the vapor bubble together. Figure 2.4 is such a plot which shows the compactness of the distribution stencil when Tryggvason's approach is used.

2.2 Interface Tracking Scheme

In the present work, front tracking is done using an immersed boundary method. The domain is 2D cartesian with staggered grid, that is, the pressure and material properties (density and viscosity) are located at the cell center, while the velocities are located at the cell face. The front is made up of linear elements (marker points) which are connected by a quadratic polynomial. At the initial moment the bubble is assumed to be circular, this corresponds to the cylindrical bubble tests done by Sussman et al. [15].

Ye et al. [8] have developed an accurate Cartesian grid method to simulate two-dimensional unsteady viscous, incompressible flows with complex immersed boundaries. They have analzed the imposition of boundary conditions on the immersed boundary by accurate discretization of the governing equations in the cell which are

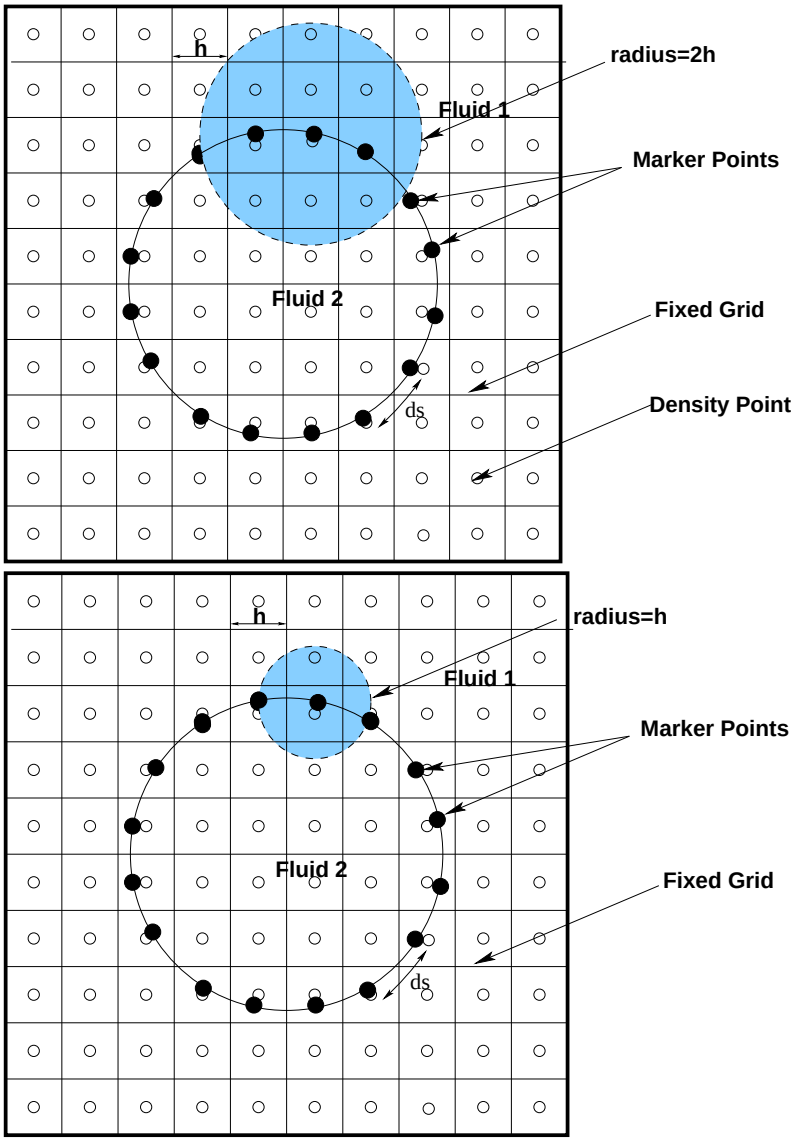


Figure 2.3: Surface tension distribution using a) Peskin's scheme, b) Tryggvason's scheme

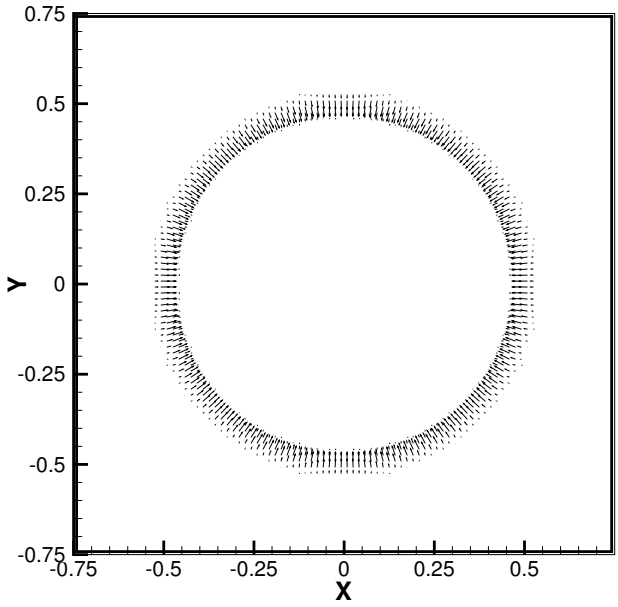


Figure 2.4: Surface tension volume force plotted as a vector plot, using Tryggvason's distribution.

cut by the immersed boundary. They have presented a new interpolation scheme which preserves the second order spatial accuracy of the solver. In this present work, this method is used and extended to incorporate a moving interface between two fluids.

The interface for front tracking is a curve since the domain is 2D. A finite number of marker points are distributed uniformly along this curve. Care is taken such that the distance between marker points is comparable to the grid size. The interface can be thus parameterized as a function of the arc length s by fitting quadratic polynomials through three consecutive marker points [17].

$$\mathbf{x}_{ib}(s) = \mathbf{a}_{ib}s^2 + \mathbf{b}_{ib}s + \mathbf{c}_{ib} \tag{2.27}$$

It may be shown that the normal vector at any given point along this interface may be evaluated from the equation of the polynomial as:

$$\begin{aligned} n_x &= -\frac{y_s}{\sqrt{x_s^2 + y_s^2}} \\ n_y &= \frac{x_s}{\sqrt{x_s^2 + y_s^2}} \end{aligned} \tag{2.28}$$

At any given point on this interface (polynomial), the curvature may be obtained from the normal vector as:

$$\kappa = -\nabla.\mathbf{n} \tag{2.29}$$

For a 2D case, the curvature can be simplified as follows:

$$\kappa = \frac{y_s x_{ss} - y_{ss} x_s}{(x_s^2 + y_s^2)^{3/2}} \tag{2.30}$$

Since, x and y are polynomials of s, the subscript s indicates the first derivative of the polynomial w.r.t s while ss represents the second derivative.

Once the location of the interface is identified, information must be communicated between the linear front and the 2D grid. For example, the information of surface tension weighting must be transferred from the front to the grid, while the information about velocity must be transferred from the grid to the front. This can be done using flags. Once a markerpoint's location is known w.r.t. the fixed grid, the cluster of the nodes surrounding the marker point can be evaluated. These grid points may be flagged as being 0, -1, or 1 depending on whether the fixed grid point is outside, near, or inside the interface. A more detailed description about the flagging scheme has been given by Balaras [20].

2.3 Numerical method for solving the governing equations

2.3.1 Discretization and fractional step method

Using a finite volume method on a staggered mesh, the governing equations can be discretized. This is done using second order central differencing for spatial derivatives and time advancement is done using a low storage third-order Runge-Kutta scheme in a fractional step method. The fractional step method, also called projection method, originally developed by Chorin [21] may be thought of as a predictor-corrector scheme. In the first step it is assumed that the pressure gradient does not affect the momentum equations. This affects the velocities and results in unphysical velocity, called predicted velocity which does not satisfy the continuity equation. The pressure is now solved separately by using the predicted velocity. Using this pressure, the actual physical velocity is obtained by the correction scheme.

This method may be illustrated as follows.

When advancing from time step n to $n+1$, the material properties are updated first. This is a simple process in the case of front tracking method, since the location of the interface is explicitly known. Based on the flags, the material properties are explicitly assigned. Then, the Navier-Stokes equation can be solved in two steps. First the effect of pressure gradient is neglected:

$$\frac{\rho^{p+1}\mathbf{u}^* - \rho^p\mathbf{u}^p}{\Delta t} = -\nabla.\rho^p\mathbf{u}^p\mathbf{u}^p + \nabla.\mu^p(\nabla\mathbf{u}^p + \nabla^T\mathbf{u}^p) + \mathbf{F}^p_{St,v} \tag{2.31}$$

It may be noted here that the interface is treated explicitly in this step. This treatment may impose stability constraint in terms of limiting the time step. Once the predicted velocity $\mathbf{u}^*$ is evaluated, a correction step is invoked and the physical velocity is recovered as follows:

$$\frac{\rho^{p+1}\mathbf{u}^{p+1} - \rho^p\mathbf{u}^*}{\Delta t} = -\nabla p \tag{2.32}$$

In order to invoke this correction step, the actual pressure field must be evaluated. This can be done such that the velocity at the next time step is divergence free (satisfies continuity equation). By taking the divergence of the above equation, the pressure equation may be obtained as:

$$\nabla\frac{1}{\rho^{p+1}}.\nabla p = \frac{1}{\Delta t}\nabla.\mathbf{u}^* \tag{2.33}$$

This is a variable coefficient Poisson (elliptic) equation. Using a finite volume method, this elliptic equation may be discretized into a 5 point stencil. This results in a matrix inversion problem as follows:

$$[a]\mathbf{p} = \mathbf{b} \tag{2.34}$$

Here, $[a]$ is a non-symmetric penta-diagonal matrix, $\mathbf{p}$ is the solution matrix, and $\mathbf{b}$ is the source vector. The non-symmetric matrix, $[a]$, must be inverted using iterative schemes, thus making this step the rate determining step for the entire simulation. Classical matrix inversion schemes cannot be used to solve this equation, since the coefficients (specific volume) are spatial functions and the matrix is non-symmetric.

Since the matrix is not symmetric, typical fast solvers like conjugate gradient solvers cannot be used. However, to be able to still use such solvers, special strategy may be adopted like using the bi-conjugate gradient solvers. This is accomplished by decomposing the original system into two subsystems. The first subsystem is the original system while the second involves a transpose matrix. A more detailed discussion on this may be found in the book by Ferziger and Peric [22].

In this fractional step method, all the terms on the right hand side, namely convection, diffusion, pressure, and external force are treated explicitly in a Runge-Kutta scheme. Second order central differencing is used for spatial discretization, while the time advancement is done with a third order explicit low storage Runge-Kutta scheme. In the discretization, a simple average is taken for the density and viscosity at the points where they are not defined. For example, the flux terms are evaluated at the cell faces - this requires the value of viscosity at the cell face. Since in a staggered grid, viscosity is located at the cell center, a simple average is taken to get viscosity at the cell face. There are several other ways to take a representative viscosity, especially in the context of a staggered mesh. Patankar [23] proposed a geometric average for the viscosity. It has been found that a simple average is sufficient for the present case, as also pointed out by Tryggvason [9].

The third order low storage Runge-Kutta scheme may be illustrated as follows. Here a more convenient tensor notation is adopted. C_i, D_i, and F_i represent the convective, diffusive, and external force (including gravity and surface tension) terms respectively along the directions $i = x$ and y. Pressure is represented by P, but pressure gradient along any given direction is represented by P_i. Runge-Kutta scheme may be visualized as time advancement in terms of three sub-steps, the first from t to $t + \Delta t/3$, the second from $t + \Delta t/3$ to $t + 3\Delta t/4$ and the last sub-step from $t + 3\Delta t/4$ to $t + \Delta t$, which is in fact the next time step.

Step 1: time advancement from t to $t + \Delta t/3$, in a projection (predictor-corrector) method.

$$\begin{aligned} G_i &= C_i^p + D_i^p + F_i^p \\ v_i &= u_i^p + \frac{1}{3}\Delta t G_i \\ \nabla^2 P^* &= \frac{1}{\Delta t/3}\frac{\partial v_i}{\partial x_i} \\ u_i^* &= v_i + \frac{\Delta t}{3}P_i^* \end{aligned} \tag{2.35}$$

Step 2: time advancement from $t + \Delta t/3$ to $t + 3\Delta t/4$.

$$\begin{aligned} G_i &= -\frac{5}{9}G_i + C_i^* + D_i^* + F_i^* \\ v_i &= u_i^* + \frac{15}{16}\Delta t G_i \\ \nabla^2 P^{**} &= \frac{1}{5\Delta t/12}\frac{\partial v_i}{\partial x_i} \\ u_i^{**} &= v_i + \frac{5\Delta t}{12}P_i^{**} \end{aligned} \tag{2.36}$$

Step 3: time advancement from $t + 3\Delta t/4$ to $t + \Delta t$.

$$\begin{aligned} G_i &= -\frac{153}{128} G_i + C_i^{**} + D_i^{**} + F_i^{**} \\ v_i &= u_i^{**} + \frac{8}{15} \Delta t G_i \\ \nabla^2 P^{p+1} &= \frac{1}{\Delta t/4} \frac{\partial v_i}{\partial x_i} \\ u_i^* &= v_i + \frac{\Delta t}{4} P_i^{p+1} \end{aligned} \tag{2.37}$$

There are several other ways of approaching this simulation. For example, the spatial discretization may be done using a high order upwind method or QUICK scheme as proposed by Leonard [24] etc. The time advancement may be done using explicit schemes (Euler, Adams-Bashforth, Runge-Kutta, etc.), semi-implicit schemes (SIMPLE, Explicit Euler: Implicit Euler in fractional step method, Runge-Kutta: Crank-Nicholson in fractional step method etc) and fully implicit schemes (Implicit Euler, Crank-Nicholson, etc.). Stability of the time advancement plays the key role in the choice of the scheme. This is often a choice between computational time and computational power. Fractional step method, with fully explicit Runge-Kutta scheme was used in the current simulation. With this method, the results were obtained within reasonable amount of time with the available computational resources.

The stability of the time advancement scheme is often the limiting criteria for the unsteady fluid flow problems. As the flow advances in time and space, the time step has to be dynamically changed to be within the stability limit of the time advancement scheme. All the terms in the Navier-Stokes equation which have been

dealt explicitly, bring a stability constraint into the time step. For a uniform grid in x and y directions the stability limitations may be summarized as follows [14], [15], [25]. The time step limitations are given as:

$$\begin{aligned}
\Delta t_{conv} &\equiv min\left(\frac{\Delta x}{|\mathbf{u}|}\right) \\
\Delta t_{diff} &\equiv min\left(\frac{3}{14}\rho Re(\Delta x)^2/\mu\right) \\
\Delta t_{surf} &\equiv \sqrt{(\rho_v+\rho_l)We/8\pi}(\Delta x)^{3/2} \\
\Delta t^{p+1} &= \frac{1}{2}min(\Delta t_{conv}+\Delta t_{diff}+\Delta t_{surf})
\end{aligned} \tag{2.38}$$

2.3.2 Interface advection and mass conservation

Once the entire flow field has been solved for, the interface has to be advected based on the velocity of the marker points. For each marker point, the local velocity is computed by taking a weighted average of the velocity nodes surrounding the marker point within a radius of Δx. Thus the velocity of each marker point may be computed as follows:

$$\mathbf{V}_{ib} = \sum_{ij} \mathbf{u}_{ij} w_{ij} h^2 \tag{2.39}$$

Here, (i, j) is the index of the grid point corresponding to the u or v velocity point, which lies within a radius of $h = \Delta x$ around the marker point with an index ib. w_{ij} is the weighting function, which is in fact a δ-function around the interface, as computed for density distribution earlier. After the velocity of the marker points is evaluated, the interface can be advected as follows:

$$\mathbf{x}_{ib}^{p+1} = \mathbf{x}_{ib}^{p} + \Delta t(\mathbf{V}_{ib}) \tag{2.40}$$

This interface advection is done within the Runge-Kutta scheme, so the Δt in the above equation corresponds to the length of the three sub steps within the RK3 method.

This solution is a weak formulation of the actual governing equations, that is, it satisfies the transport equations only in an integral sense. In the context of an immersed interface method, it is necessary to ensure that the mass of the vapor bubble is also conserved apart from the global continuity. In a 2D case, this is a simple conservation of area enclosed by the interface. After the interface is advected, the change in area is evaluated and if the normalized area change is not within the specified tolerance limit, a shape correction method is invoked. The change in the area can be evaluated as:

$$\Delta A = \sum_{ib} [(x_{ib}^{n+1} - x_{ib}^{n})^2 + (y_{ib}^{n+1} - y_{ib}^{n})^2]^{1/2} \Delta s_{ib} \tag{2.41}$$

And the constraint is specified as:

$$\frac{\Delta A}{A_0} \leq 1e-5 \tag{2.42}$$

where A_0 is the area of the bubble at the initial time, ib is the index of the marker points, (x_{ib}, y_{ib}) is the location of marker point at the time step of n or $n+1$ as seen in the superscript, and Δs is the arc length of the interface at the marker point.

If the above constraint is not satisfied, then the shape has to be corrected. This can be done by a simple bisection scheme. The new location of the interface is guessed between s^{n+1} and s^n and the residual of the area conservation is checked. If it satisfies the constraint, it is considered as the new location of the interface. If the constraint is not satisfied, the shape correction method is iterated until convergence

is reached. In most cases where there is a single bubble without shape deformation, this shape correction method was found to be insignificant as also noted by Tryggvason [9]. However, in cases of high bubble deformation, bubble merger of fracture, this criteria becomes very critical. In certain cases of bubble merger and fracture, this criteria had to be relaxed since the results are heavily dependant on the numerical resolution. The physical models governing the merger or split up of the bubbles are still incomplete and require formulation which are beyond the continuum level as pointed out by Ye et al. [8].

2.4 Summary of the numerical scheme

A brief description of the numerical algorithm is given in this section.

1. At the initial time, all the velocities and pressure are initialized to zero. The location of the interface is prescribed by distributing the marker points on a linear front.

2. Within the Runge-Kutta scheme the following steps are implemented.

   (a) The material properties are updated and distributed onto the grid based on the location of the interface. Flags are evaluated to mark the location of the interface on the fixed grid.

   (b) Based on the location of the interface, the surface tension volume force components are evaluated and distributed on to the grid points surrounding the interface.

(c) Predictor: The problem is decoupled (pressure and velocity are separated) and the effect of pressure is ignored in this step. The convective, diffusive, gravitational and surface tension force terms on the right hand side of the Navier-Stokes equation are evaluated explicitly. The predicted velocity is evaluated and the velocity boundary conditions are imposed on the predicted velocities.

(d) The coefficients of the pressure Poisson solver are evaluated as functions of specific volume. The divergence of the predicted velocity is the source term for this equation since the predicted velocity does not satisfy the continuity equation.

(e) The pressure equation is solved as a matrix inversion problem. This is an iterative method solved using bi-conjugate gradient solver with SIP preconditioner.

(f) Corrector: Based on the pressure gradient, the predicted velocities are corrected and the actual velocity at the next time step is recovered.

(g) Interface advection: Once the velocity is known, the marker points are advected by evaluating the local velocity of each marker point. This is done similar to the density and surface tension distribution. The mass of the bubble is conserved separately and a shape correction scheme is invoked if the mass is not conserved.

3. In most simulations, the time advancement is stopped once the velocity reaches a steady state value. In other cases, the simulation is stopped at a specific

known time to compare the results.

Chapter 3

Sharp Interface Method

In this chapter the sharp interface method (SIM) is presented. Surface tension is introduced as a jump discontinuity in the pressure Poisson equation. The development and validation of variable coefficient Poisson solver is discussed. The validated pressure solver is then integrated into the Navier-Stokes solver in the fractional step method. Implementation of immersed boundary method is described and mass conservation for the vapor bubble is enforced as an additional constraint to satisfy global and local continuity.

SIM is a direct approach, where the material properties and the interface itself are treated sharply. The pressure jump across the interface is obtained quite accurately, since the jump discontinuity is imposed as a Dirichlet boundary condition in the method. Since the equations are similar to those with DIM, the focus is mostly on the distinctive part.

3.1 Mathematical model and governing equations

3.1.1 Conservation of mass

The basic continuity equation is similar to the distributed interface method. The set of equations representing the mass conservation and incompressibility of the

fluids can be written as:

$$\frac{D\rho}{Dt} = 0 \tag{3.1}$$

$$\frac{D\mu}{Dt} = 0 \tag{3.2}$$

$$\frac{\partial \rho}{\partial t} + \nabla.\rho\mathbf{u} = 0 \tag{3.3}$$

Thus the continuity equation can be derived as:

$$\nabla.\mathbf{u} = 0 \tag{3.4}$$

3.1.2 Conservation of momentum

Unlike DIM, the surface tension is incorporated directly into the pressure solver as a Dirichlet boundary condition at the interface. The surface tension is not treated as an external force in the momentum equations. Thus the momentum equations will be the standard Navier-Stokes equations with external forces like gravity.

$$\frac{\partial \rho\mathbf{u}}{\partial t} + \nabla.\rho\mathbf{u}\mathbf{u} = -\nabla p + \nabla.\tau + \rho\mathbf{g} \tag{3.5}$$

where, the shear stress tensor can be expressed as

$$\tau = \mu(\nabla\mathbf{u} + \nabla^T\mathbf{u}) \tag{3.6}$$

Using the same non dimensional numbers as with DIM, the normalized set of governing equations is thus

$$\begin{aligned} \nabla.\mathbf{u} &= 0. \\ \frac{\partial \rho\mathbf{u}}{\partial t} + \nabla.\rho\mathbf{u}\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla.\tau + \frac{1}{Fr}\rho\mathbf{g} \end{aligned} \tag{3.7}$$

Here the superscript * is dropped for convenience, but it must be noted that all the variables are non dimensional.

3.1.3 Initial and boundary conditions

All the initial and boundary conditions remain the same from DIM. That is, at the initial time, all the velocity components and pressure, are assumed to be zero and the interface appears instantaneously. The boundary conditions are assumed to be homogeneous Neumann boundary conditions for velocity and pressure.

In SIM, the pressure jump is explicitly specified across the interface through the Young-Laplace equation. Ye et al. [8] have shown that for incompressible Newtonian fluids with known viscosity and density ratios, the pressure jump is obtained in the non-dimensional form as follows.

$$\begin{aligned} p_l - p_v + \frac{1}{We}\kappa &= \frac{1}{Re}\left\{\left(\frac{\partial u_n}{\partial n}\right)_l - \left(\frac{\mu_v}{\mu_l}\right)\left(\frac{\partial u_n}{\partial n}\right)_v\right\} \\ &- \left[(u_n)_l - (u_n)_{int}\right](u_n)_t \\ &+ \left(\frac{\rho_v}{\rho_l}\right)\left[(u_n)_v - (u_n)_{int}\right](u_n)_v \end{aligned} \tag{3.8}$$

This is directly obtained from the Young-Laplace equation for viscous fluids through the momentum balance across the interface. When the pressure Poisson solver is incorporated into the Navier-Stokes solver, the above equation is used to obtain the pressure jump $[p]$, all other discontinuities like $[p_n], [p_{nn}], [\beta p_n], [(\beta p_n)_n]$ are assumed to be zero. This condition is true for a typical multiphase flow consisting of incompressible fluids.

3.2 Numerical method for solving the governing equations

3.2.1 Discretization and fractional step method

The governing equations are discretized in space using the standard central differencing scheme, on a staggered uniform mesh in a 2D domain in a Cartesian system. Time advancement is done by a low-storage third order Runge-Kutta scheme in a fractional step method. The overall scheme is identical to the distributed interface method as illustrated in detail in chapter 2. However the surface tension term is included in the pressure Poisson solver instead of adding it as an external force in the actual momentum equations.

One of the fundamental difficulties in the direct methods for solving multi phase flows is the representation of the interface and capturing the pressure discontinuity due to surface tension. Typical direct methods used separate grids to represent the two fluids and used an iterative scheme to match the interface boundary conditions between the two grids. This imposes several limitations since the simulation can be quite expensive in terms of computational cost and it is very difficult to extend it to more complex cases. In the present simulation, a new approach is presented, where the pressure jump is obtained by modifying the pressure Poisson equation in the fractional step method.

It is also important to maintain consistency in the sharp representation of both density and surface tension. The density discontinuity is resolved in SIM to a sub-cell accuracy. For example if the interface lies between x_i and x_{i+1} then the

sub-cell parameter is defined as:

$$a = \frac{x_{int} - x_i}{x_{i+1} - xi} \tag{3.9}$$

and the specific volume is evaluated based on the sub cell resolution parameter as suggested by Liu et al [13]:

$$\beta_{avg} = \frac{\beta^+ \beta^-}{\beta^+ a + \beta^-(1-a)} \tag{3.10}$$

When advancing from time step n to $n+1$, the material properties are updated first. This is a simple process in the case of front tracking method, since the location of the interface is explicitly known. Based on the flags, the material properties are explicitly assigned. Then, the Navier-Stokes equation can be solved in two steps. First the effect of pressure gradient is neglected:

$$\frac{\rho^{p+1}\mathbf{u}^* - \rho^p \mathbf{u}^p}{\Delta t} = -\nabla.\rho^p \mathbf{u}^p \mathbf{u}^p + \nabla.\mu^p(\nabla \mathbf{u}^p + \nabla^T \mathbf{u}^p) \tag{3.11}$$

Once the predicted velocity $\mathbf{u}^*$ is evaluated, a correction step is invoked and the physical velocity is recovered as follows:

$$\frac{\rho^{p+1}\mathbf{u}^{p+1} - \rho^p \mathbf{u}^*}{\Delta t} = -\nabla p \tag{3.12}$$

In order to invoke this correction step, the actual pressure field must be evaluated. This can be done such that the velocity at the next time step is divergence free (satisfies the continuity equation). By taking the divergence of the above equation, the pressure equation may be obtained as:

$$\nabla \frac{1}{\rho^{p+1}}.\nabla p = \frac{1}{\Delta t}\nabla.\mathbf{u}^* \tag{3.13}$$

This equation is now modified to impose discontinuity in the pressure across the interface as follows:

$$\nabla \frac{1}{\rho^{p+1}} . \nabla p = \frac{1}{\Delta t} \nabla . \mathbf{u}^* + C_\Gamma \tag{3.14}$$

Hence a new variable coefficient Poisson (elliptic) solver must be developed which can impose discontinuity across the interface, as represented by C_Γ in the above equation. Such a Poisson solver can be used as a standard black box iterative solver like the bi-conjugate gradient solver. Once the actual velocities are recovered, the interface is advected and the mass of the bubble is tested for its conservation. If the change in mass of the bubble with respect to the initial mass is not within the acceptable tolerance limits, the shape correction scheme is invoked. The shape of the interface is corrected using the bisection method as illustrated in chapter 2. In the next section the development of the new Poisson solver is described.

3.2.2 Variable coefficient Poisson equation for discontinuous solutions

To match the interface boundary condition for pressure a new pressure Poisson equation must be developed similar to Liu et al. [13] and Berthelsen [26]. Using this new solver, the momentum equations can be solver similar to the ghost fluid method by Kang et al [12]. The actual pressure Poisson equation can be written as:

$$\nabla . \left(\frac{1}{\rho} \nabla p \right) = \frac{1}{\Delta t} \nabla . \mathbf{u}^* \tag{3.15}$$

In a 2D Cartesian system, this can be written as:

$$(\beta p_x)_x + (\beta p_y)_y = f(x, y) \tag{3.16}$$

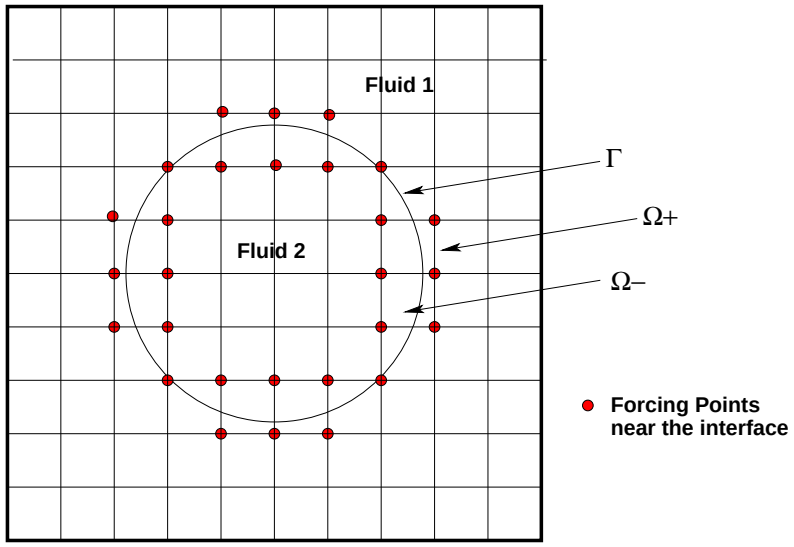


Figure 3.1: Domain, location of interface and the forcing points

This equation must be solved inside a domain Ω which is divided by the closed interface into exterior, Ω^+, and interior Ω^- regions. The interface itself is represented by a contour Γ. Fig 3.1 illustrates such an interface in a ghost fluid method.

The pressure discontinuity is only across the interface, hence an extra source term is introduced on the right hand side of the above pressure equation as follows:

$$(\beta p_x)_x + (\beta p_y)_y = f(x,y) - C(x,y) \tag{3.17}$$

The extra term $C(x,y)$ accounts for the pressure discontinuity at the pressure nodes on the fixed grid near the interface. At all other points, far from the interface, this function vanishes. In a discrete sense, with central differencing for spatial coordinates, the above equation can be expressed in the standard five point discrete stencil formulation as follows:

$$\frac{\beta_{i+1/2,j}(p_{i+1,j} - p_{i,j}) - \beta_{i-1/2,j}(p_{i,j} - p_{i-1,j})}{(\Delta x)^2} \qquad +$$

$$\frac{\beta_{i,j+1/2}(p_{i,j+1} - p_{i,j}) - \beta_{i,j-1/2}(p_{i,j} - p_{i,j-1})}{(\Delta y)^2} = f_{i,j} - C_{i,j} \tag{3.18}$$

Now the main objective is to evaluate the new source term $C(x, y)$ which can bring about the discontinuity. It may be noted here that by using the above formulation, standard 'black box' solvers may be used to solve the Poisson equation. This preserves the order of accuracy of the discretization and the convergence is not affected since only the source term is being modified. It has been shown by Berthelsen [26] that the discontinuity can be decomposed into each dimension separately. Hence the source term may be represented as:

$$C_{i,j} = C^x_{i,j} + C^y_{i,j} \tag{3.19}$$

Without loss of generality, the discontinuity may be just evaluated for the x direction and then a similar analysis can be done for y. To prove the validity of the formulation, the discretization can be illustrated for x. Consider the one dimensional variable coefficient Poisson equation:

$$(\beta p_x)_x = f(x) \tag{3.20}$$

Using second order central differencing this equation may be discretized as:

$$\frac{1}{\Delta x}\left[\beta_{i+1/2}\left(\frac{p_{i+1} - p_i}{\Delta x}\right) - \beta_{i-1/2}\left(\frac{p_i - p_{i-1}}{\Delta x}\right)\right] = f_i \tag{3.21}$$

Suppose the interface lies within the interval (x_i, x_{i+1}), then the above discretization has to be modified to correct for the pressure discontinuity and the density difference. Hence the equation has to be modified as:

$$\frac{1}{\Delta x}\left[\beta^*_{i+1/2}\left(\frac{p^-_{i+1} - p^+_i}{\Delta x}\right) - \beta_{i-1/2}\left(\frac{p^+_i - p^+_{i-1}}{\Delta x}\right)\right] = f_i \tag{3.22}$$

Here, p^+ stands for the pressure outside the interface while p^- is the pressure inside. Since a cut-cell approach is adopted, the interface must be resolved within a single cell to evaluate the density correction β^*, thus a sub-cell resolution is necessary. With the interface located between the grid points i and $i+1$, the subcell parameter can be defined as:

$$a = \frac{x_{int} - x_i}{x_{i+1} - x_i} \tag{3.23}$$

Since the pressure jump across the interface is explicitly known, the above equation can be written as:

$$\frac{1}{\Delta x}\left[\beta^*_{i+1/2}\left(\frac{p^+_{i+1} + C_\Gamma - p^+_i}{\Delta x}\right) - \beta_{i-1/2}\left(\frac{p^+_i - p^+_{i-1}}{\Delta x}\right)\right] = f_i \tag{3.24}$$

Re-arranging this into standard three point stencil form:

$$\frac{1}{\Delta x}\left[\beta^*_{i+1/2}\left(\frac{p_{i+1} - p_i}{\Delta x}\right) - \beta_{i-1/2}\left(\frac{p_i - p_{i-1}}{\Delta x}\right)\right] = f_i - \frac{\beta^*_{i+1/2} C_\Gamma}{\Delta x^2} \tag{3.25}$$

The discontinuous term C_Γ can be determined based on the location of interface, surface tension, and viscous forces. If the second derivative is also discontinuous, then it must be also corrected for $[(\beta u_x)_x]$. In the following section, a detailed derivation of the discontinuous terms is illustrated.

3.2.3 Evaluating the discontinuous terms in the discrete form

In this section, a more rigorous derivation of the discontinuous terms is illustrated. The main objective is to derive the discrete form of the three (or five point) stencil with discontinuous terms in a more generic case. Taylor's expansion is written for the pressure term near the interface. The discontinuity is identified and added

as source terms to the right hand side. These additional terms can be quantified from constitutive relationships like the inviscid fluid pressure jump (Young-Laplace) equation:

$$\Delta p = \sigma\kappa \tag{3.26}$$

Since central differencing is used for spatial derivatives, the flux term is evaluated at $x_{i+1/2}$. It becomes necessary to identify which side of the cell face the interface lies, i.e., if $x_{int} \in (x_i, x_{i+1/2})$ then $0 < a \leq 1/2$ and the coefficients for the index i and $i+1$ must be modified. The discrete equations for the pressure may be derived directly from Taylor's expansion as [26]:

$$p(x_{i+1}) = p(x_{i+1/2}) + p_x(x_{i+1/2})\frac{\Delta x}{2} + \frac{1}{2}p_{xx}(x_{i+1/2})\left(\frac{\Delta x}{2}\right)^2 + \bigcirc(\Delta x^3) \tag{3.27}$$

and

$$p(x_i) = p(x_{i+1/2}) - p_x(x_{i+1/2})\frac{\Delta x}{2} + \frac{1}{2}p_{xx}(x_{i+1/2})\left(\frac{\Delta x}{2}\right)^2 - C_1(x,a) + \bigcirc(\Delta x^3) \tag{3.28}$$

The correction term represents:

$$C_1(x,a) = [p] - [p_x]a\Delta x + \frac{1}{2}[p_{xx}]a^2\Delta x^2 \tag{3.29}$$

Thus the flux term can be evaluated as:

$$p_x(x_{i+1/2}) = \frac{p(x_{i+1}) - p(x_i)}{\Delta x} - \frac{C_1(x,a)}{\Delta x} + \bigcirc(\Delta x^2) \tag{3.30}$$

Similarly with a variable coefficient, in this case specific volume β, the flux term can be written in the discrete form as:

$$\beta p_x(x_{i+1/2}) = \beta p(x_i) + (\beta p_x)_x(x_i)\frac{\Delta x}{2} + C_2(x,a) + \bigcirc(\Delta x^2) \tag{3.31}$$

$$\beta p_x(x_{i-1/2}) = \beta p(x_i) - (\beta p_x)_x(x_i)\frac{\Delta x}{2} + \bigcirc(\Delta x^2) \tag{3.32}$$

where the correction term is:

$$C_2(x,a) = [\beta p_x] + \frac{1}{2}[(\beta p_x)_x](1-2a)\Delta x \tag{3.33}$$

The divergence term can be thus written in the discrete form as:

$$(\beta p_x)_x(x_i) = \frac{\beta p_x(x_{i+1/2}) - \beta p_x(x_{i-1/2})}{\Delta x} - \frac{C_2}{\Delta x} + \bigcirc(\Delta x) \tag{3.34}$$

Simplifying this further we obtain the standard three-point stencil discrete form of the Poisson's equation in one dimension, with an additional term representing the discontinuity:

$$(\beta p_x)_x(x_i) = \frac{\beta_{i+1/2}(p(x_{i+1}) - p(x_i)) - \beta_{i-1/2}(p(x_i) - p(x_{i-1}))}{\Delta x^2} + C_i + \bigcirc(\Delta x) \tag{3.35}$$

The additional term, representing the discontinuity, can be obtained for a general case for interface lying anywhere between any given two grid points [26].

$$C_i = S_i \beta^* \frac{C_1(x,a)}{h^2} + S_i \frac{C_2(x,a)}{h} \tag{3.36}$$

where,

$$S_i = \begin{cases} \text{-1} & \text{in } \Omega^- \\ 1 & \text{in } \Omega^+ \end{cases} \tag{3.37}$$

The discontinuous terms are:

$$C_1(x,a) = \begin{cases} [p] - \lambda[p_x]a\Delta x + \frac{1}{2}[p_{xx}]a^2\Delta x^2 & \text{if } (\Omega^+ \textit{ and } 0 \le a < 1/2) \\ & \text{or } (\Omega^- \textit{ and } 0 < a \le 1/2) \\ [p] + \lambda[p_x](1-a)\Delta x + & \text{if } (\Omega^+ \textit{ and } 1/2 \le a < 1) \\ \frac{1}{2}[p_{xx}](1-a)^2\Delta x^2 & \text{or } (\Omega^- \textit{ and } 1/2 < a \le 1) \end{cases} \tag{3.38}$$

and

$$C_2(x,a) = \begin{cases} \lambda[\beta p_x] + \frac{1}{2}[(\beta p_x)_x](1-2a)\Delta x & \text{if } (\Omega^+ \; and \; 0 \leq a < 1/2) \\ & \text{or } (\Omega^- \; and \; 0 < a \leq 1/2) \\ 0 & \text{otherwise} \end{cases} \tag{3.39}$$

Also the terms λ, β^* are defined as follows:

1. If the interface lies in the interval (x_i, x_{i+1}) then $\lambda = 1, \beta^* = \beta_{i+1/2}$

2. If the interface lies in the interval (x_{i-1}, x_i) then $\lambda = -1, \beta^* = \beta_{i-1/2}$

It may be noted here that, although the local truncation error around the interface is first order, the global accuracy is still preserved since the interface is always one dimension lower than the fixed grid. This is the major advantage of the front tracking scheme as highlighted earlier. While preserving the order of accuracy, the scheme is computationally much less expensive. Also, in the typical case of two phase flows, the higher order derivatives of pressure are continuous thereby causing the terms $[p_x], [p_{xx}], [\beta p_x], [(\beta p_x)_x]$ to vanish. A similar correction term is obtained in the y-direction for the points around the interface.

3.2.4 Validation of the variable coefficient Poisson equation with discontinuity

A two dimensional variable coefficient Poisson solver has been developed, which can impose a discontinuity across interface as a Dirichlet boundary condition on the immersed boundary. In this section, several test cases are undertaken when the actual analytical solution is known and the Poisson solver is validated. In

all these cases, the domain boundary conditions, the location of the interface, and the discontinuity at the interface are all known. The analytical solution is obtained through solution of the partial differential equations using analytical methods, e.g., the variable separable method, etc. The error norms and the order of accuracy are evaluated and compared with analytical and numerical solutions of Berthelsen [26] and Liu et al. [13].

- Example 1: In this example, a simple 2D Laplace equation, $\nabla^2 p = 0$, is solved in the domain $[0, 2] \times [0, 2]$. The interface is described by the equation $x^2 + y^2 = 1/4$ and the discontinuities are imposed as: $[p] = 0$ and $[p_n] = 2$. The analytical (exact) solution is given by:

$$p(x, y) = \begin{cases} 1 & \text{in } \Omega^- \\ 1 + log(2\sqrt{x^2 + y^2}) & \text{in } \Omega^+ \end{cases} \tag{3.40}$$

  The domain boundary conditions are imposed from the exact solution. The numerical solution and the error distribution are plotted in Fig 3.2.

- Example 2: This case is again a Laplace equation with the governing equation:

$$p_{xx} + p_{yy} = 0 \tag{3.41}$$

  The discontinuities are given by:

$$\begin{aligned} [p] &= (y^2 - x^2) \\ [p_n] &= 4(y^2 - x^2) \end{aligned} \tag{3.42}$$

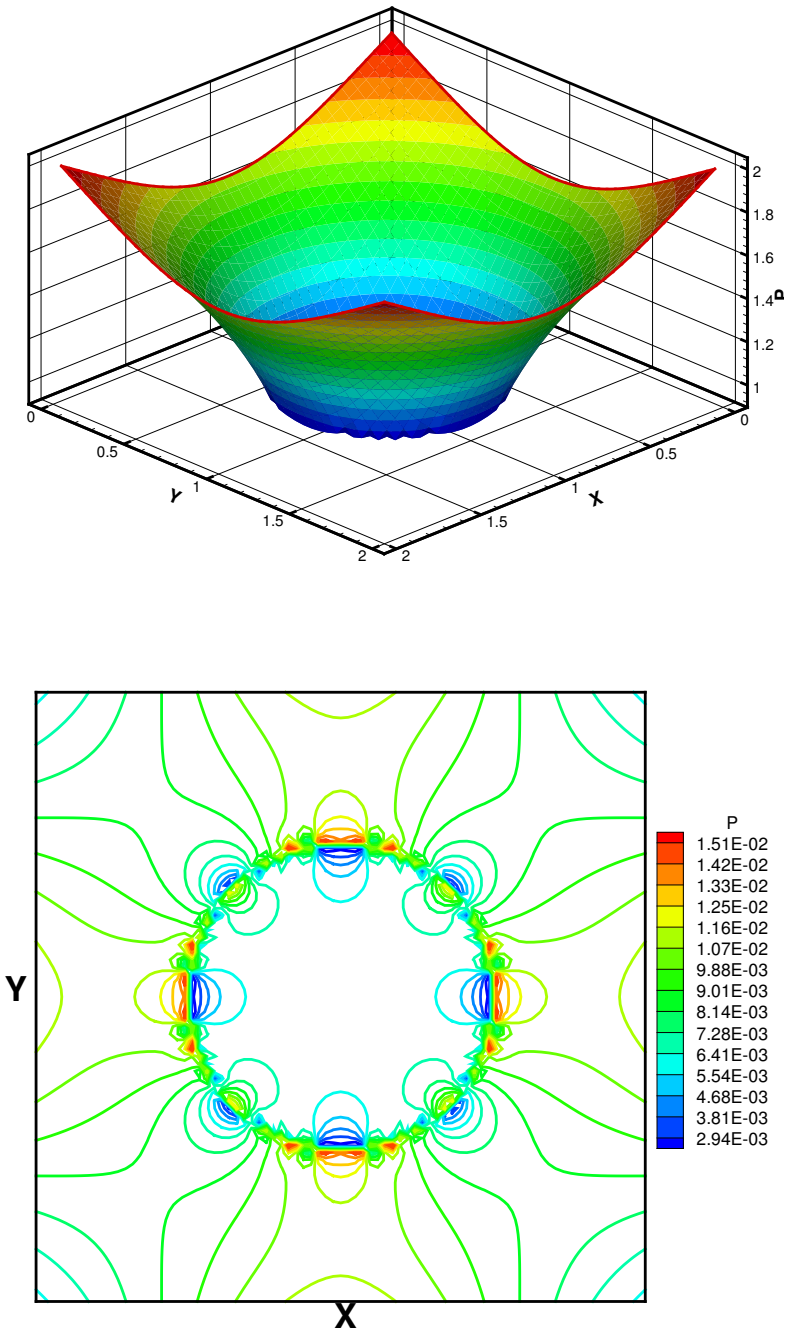


Figure 3.2: Numerical solution and error distribution for Example 1

The exact solution is given by:

$$p(x,y) = \begin{cases} -x^2 - y^2 & \text{in } \Omega^- \\ 0 & \text{in } \Omega^+ \end{cases} \tag{3.43}$$

The domain boundary condition is set to be the Dirichlet boundary condition $p = 0$. The numerical solution and the error distribution are plotted in Fig 3.3.

- Example 3: In this example a discontinuity is imposed on the pressure and its first derivative. The domain and the interface location are the same as in the previous examples. The governing equation is:

$$(\beta p_x)_x + (\beta p_y)_y = f(x,y) \tag{3.44}$$

With the coefficients given by:

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1 & \text{if } r < 1/2 \\ 1 & \text{if } r \geq 1/2 \end{cases} \tag{3.45}$$

The right hand side given by:

$$f(x,y) = \begin{cases} 2e^x(ysiny - xcosy) & \text{if } r < 1/2 \\ 0 & \text{if } r \geq 1/2 \end{cases} \tag{3.46}$$

The discontinuities are given by:

$$\begin{aligned} [p] &= -e^x cosy \\ [\beta p_n] &= 2e^x(x^2 + y^2 + 1)(ysiny - xcosy) \end{aligned} \tag{3.47}$$

The domain boundary condition is set to be $p = 0$. The numerical solution and the error distribution are plotted in Fig 3.4.

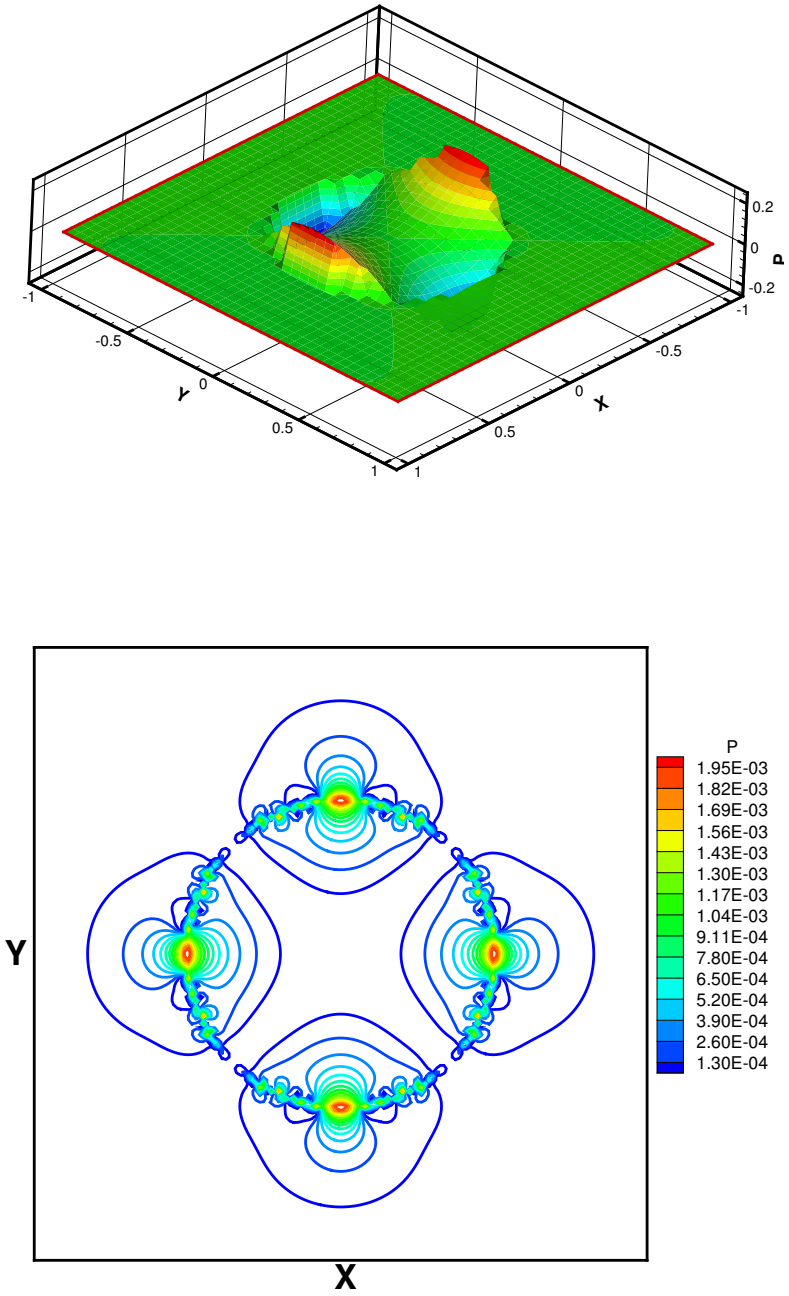


Figure 3.3: Numerical solution and error distribution for Example 2

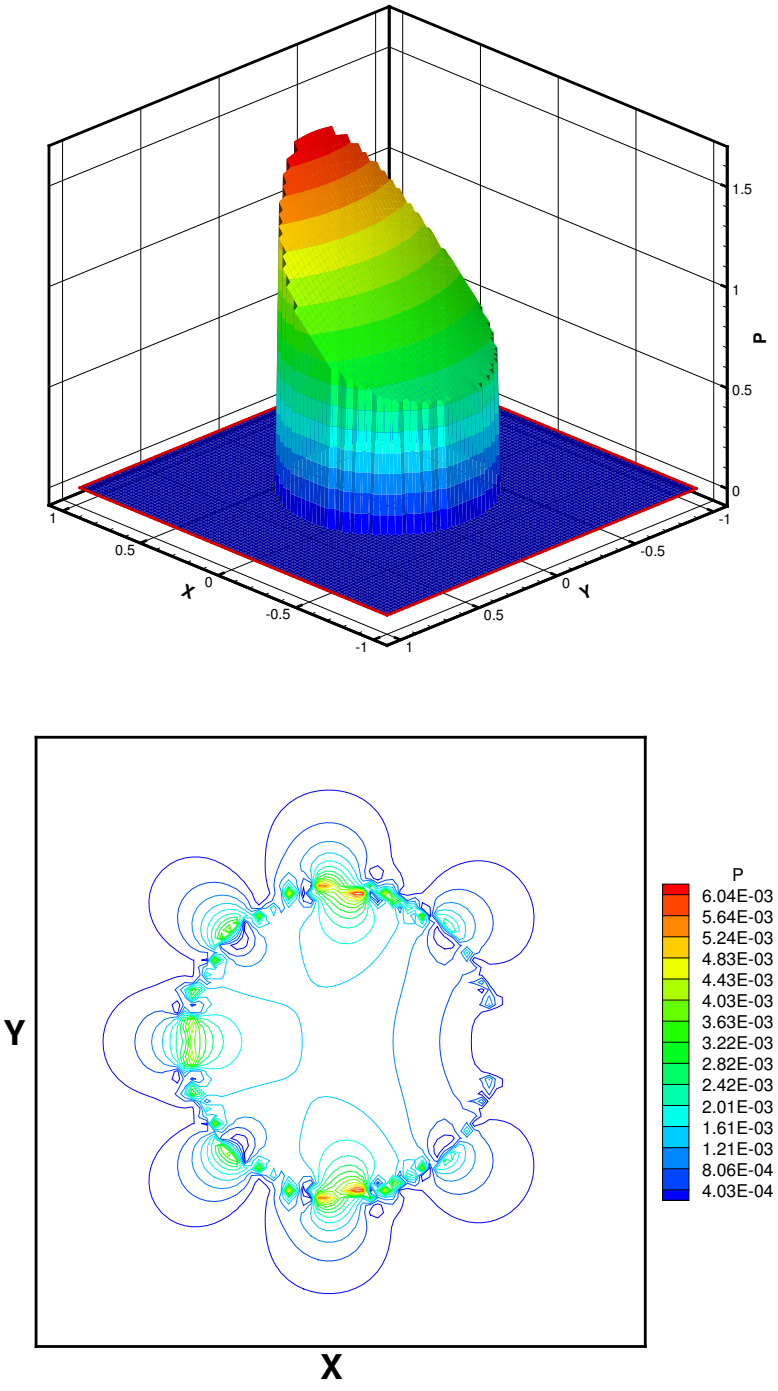


Figure 3.4: Numerical solution and error distribution for Example 3

- Example 4: In this case the Poisson equation is solved in a domain defined by $[0,1] \times [0,1]$ when the interface is described by the equation: $(x-0.5)^2 + (y-0.5)^2 = 0.25^2$. The governing equation is:

$$(\beta p_x)_x + (\beta p_y)_y = f(x,y) \tag{3.48}$$

with the coefficients given by:

$$\beta(x,y) = \begin{cases} 2 & \text{in } \Omega^- \\ 1 & \text{in } \Omega^+ \end{cases} \tag{3.49}$$

The right hand side given by:

$$f(x,y) = \begin{cases} 8(x^2+y^2-1)e^{-x^2-y^2} & \text{in } \Omega^- \\ 0 & \text{in } \Omega^+ \end{cases} \tag{3.50}$$

The discontinuities are given by:

$$\begin{aligned} [p] &= -e^{-x^2-y^2} \\ [\beta p_n] &= 8(2x^2+2y^2-x-y)e^{-x^2-y^2} \end{aligned} \tag{3.51}$$

The exact solution is given by:

$$p(x,y) = \begin{cases} e^{-x^2-y^2} & \text{in } \Omega^- \\ 0 & \text{in } \Omega^+ \end{cases} \tag{3.52}$$

The domain boundary condition is set to be the Dirichlet boundary condition $p = 0$. The numerical solution and the error distribution are plotted in Fig 3.5.

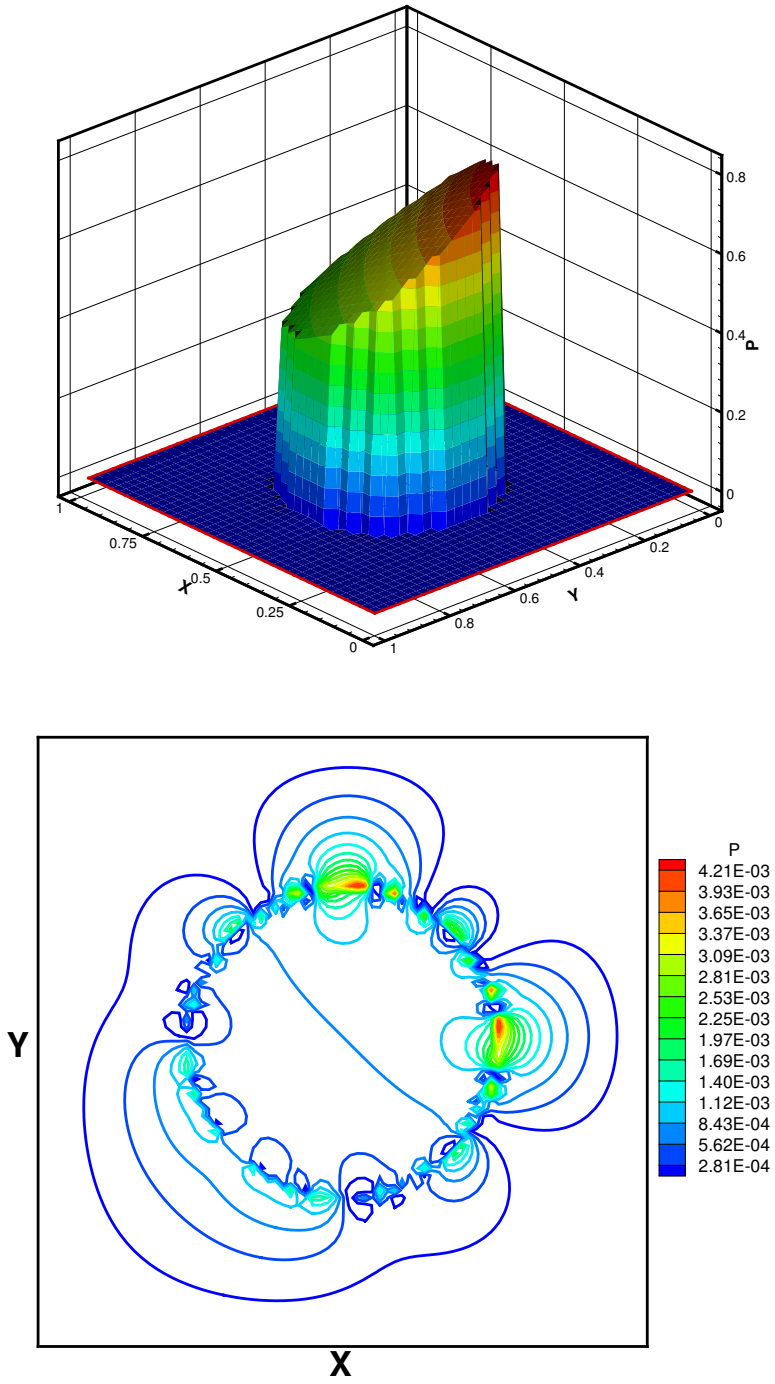


Figure 3.5: Numerical solution and error distribution for Example 4

3.3 Summary of the numerical scheme

A brief description of the numerical algorithm is given in this section.

1. At the initial time, all the velocities and pressure are initialized to zero. The location of the interface is prescribed by distributing the marker points on a linear front.

2. Within the Runge-Kutta scheme the following steps are implemented.

   (a) The material properties are updated and distributed onto the grid based on the location of the interface. Flags are evaluated to mark the location of the interface on the fixed grid.

   (b) Predictor: The problem is decoupled (pressure and velocity are separated) and the effect of pressure is ignored in this step. The convective, diffusive, gravitational and surface tension force terms on the right hand side of the Navier-Stokes equation are evaluated explicitly. The predicted velocity is evaluated and the velocity boundary conditions are imposed on the predicted velocities.

   (c) Using the Young-Laplace equation the pressure jump across the interface is evaluated and imposed as a Dirichlet boundary condition on the immersed interface. The source term representing the discontinuity is added to the right hand side of the pressure equation.

   (d) The coefficients of the pressure Poisson solver are evaluated as functions of specific volume. The divergence of the predicted velocity is the source

term for this equation, since the predicted velocity does not satisfy the continuity equation.

(e) The pressure equation is solved as a matrix inversion problem. This is an iterative method solved using a bi-conjugate gradient solver with SIP preconditioner.

(f) Corrector: Based on the pressure gradient, the predicted velocities are corrected and the actual velocity at the next time step is recovered.

(g) Interface advection: Once the velocity is known, the marker points are advected by evaluating the local velocity of each marker point. This is done similar to the density and surface tension distribution. The mass of the bubble is conserved separately and a shape correction scheme is invoked if the mass is not conserved.

3. In most simulations, the time advancement is stopped once the velocity reaches a steady state value. In other cases, the simulation is stopped at a specific known time to compare the results.

Chapter 4

Results and discussion

Distributed and sharp interface techniques have been developed as detailed in Chapter 2 and 3 respectively. In this chapter, these two methods are tested on standard analytical solutions and for specific cases in literature where the numerical results have been validated. The order of accuracy is tested for each of these methods and convergence tests are taken up to study the cost-efficiency of the simulation. The effect of density ratio on the convergence and the accuracy is studied. Then the evolution of the liquid droplet/vapor bubble is studied with various Reynolds and Bond numbers. Finally, special cases like bubble merger and fracture are simulated.

4.1 Stationary inviscid liquid droplet

Consider a 2D liquid droplet in equilibrium. If the viscous, gravitational and other external forces are absent, the surface tension of the liquid causes it to become spherical and the droplet continues to be spherical. This is the simplest case of multiphase flows and corresponds to the equilibrium rod example of Brackbill et al. [14]. This case is simulated using both DIM and SIM and the results are compared with the analytical results and numerical results of Brackbill et al [14].

The main parameters which govern this flow are: the density ratio ρ_l/ρ_v, the viscosity ratio μ_l/μ_v, the Reynolds number UD/ν and the Bond number $\rho_l g U^2/\sigma$.

The Reynolds number is based on the terminal velocity of the bubble which is given by: $U = \sqrt{gD}$, where D is the initial diameter of the bubble. Since both the fluids are assumed to be inviscid, $Re = \infty$, and the viscosity ratio consequently drops out of the simulation. All parameters being non-dimensional, suppose the density of the ambient fluid is $\rho_\infty = 0.5$, that of the droplet is $\rho_l = 1.0$, and the surface tension is $\sigma = 0.02361$. For a liquid droplet of initial diameter $D = 1$, the pressure jump across the interface can be evaluated from the Young-Laplace formula as:

$$\Delta p = \frac{\sigma}{R} = 0.04722 \tag{4.1}$$

This analytical solution is used to compare the results of the current simulation and the numerical results of Brackbill et al. [14]. In the present simulation, the domain was $1.5D \times 1.5D$ with a grid size ranging from 30×30 to 120×120. Fig 4.1 shows the pressure contours and the surface tension body force vectors. As compared to similar result of Brackbill et al., this simulation uses a more compact stencil for the density and surface tension distribution. Hence the pressure varies smoothly from the liquid droplet to the ambient fluid.

The RMS error in the pressure can be defined as the norm:

$$L_2 = \left[\frac{\sum_{i,j=1}^{N_d} (p_{i,j} - p_{ex})^2}{N_d \; p_{ex}^2} \right]^{1/2} \tag{4.2}$$

Where, N_d is the total number of grid points inside the liquid droplet and p_{ex} is the exact pressure obtained from the analytical solution. Another indicator for accuracy is the error in curvature. This can be defined as:

$$\epsilon_\kappa = MAX_{ib=1}^{Ns} \left(\frac{\kappa_{ib} - \kappa_{ex}}{\kappa_{ex}} \right) \tag{4.3}$$

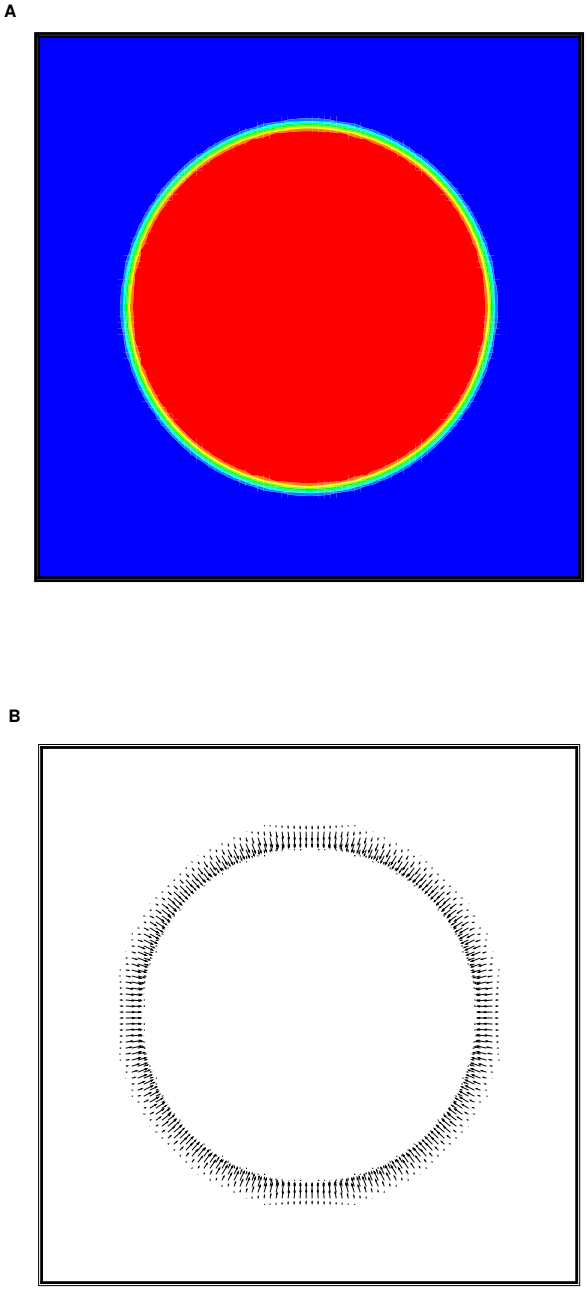


Figure 4.1: Compact distribution stencil: density contours and surface tension body force vectors for DIM

where, κ_{ib} is the local curvature evaluated at every marker point and κ_{ex} is the exact curvature which is simply $1/R = 2.0$ in this particular case. It must be noted here that the L_2 norm in pressure is the direct indicator of the actual interface dynamics being captured. In the case of DIM, this is a very strict criteria since the pressure discontinuity is implemented indirectly. For SIM, the L_2 norm of the pressure is expected to be very accurate since the pressure discontinuity is directly implemented as a Dirichlet boundary condition.

Fig 4.2 shows the order of accuracy based on the L_2 norm of pressure as the grid size is changed from a coarse to fine grid. In this study, the time step is kept constant at $\Delta t = 10^{-4}$ and the simulation is run until time=0.01. The average global order of accuracy of DIM is 1.92 while that of SIM is 2.17. The results indicate that SIM is more accurate than DIM by at least 2 orders of magnitude. In order to have the same accuracy, DIM requires many more points than SIM. For example to get an accuracy of L_2 norm of pressure to within 10^{-3}, DIM requires a grid size of at least 0.001 while SIM requires a grid size of just 0.7. The coarse grid result with 30×30 grid points is not very realistic in both the simulations, since typical multiphase flow simulations require at least 40 grid points along the diameter in each direction. Hence it may be concluded that SIM is more accurate and has higher accuracy than DIM. This in turn translates into better computational efficiency and less computational cost.

Fig 4.3 compares the convergence of the Poisson solver for both DIM and SIM. In this figure, the residual is plotted after every iteration of the Poisson solver. In both the cases, a grid size of 60×60 was used with homogeneous Neumann

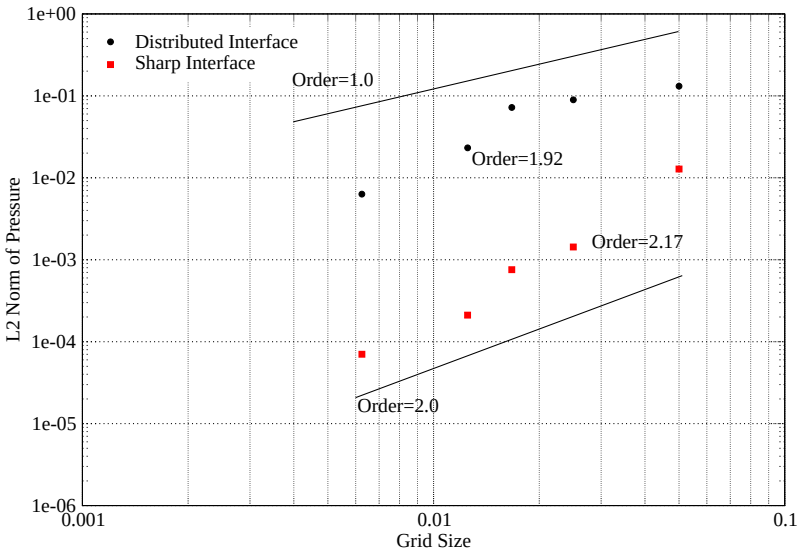
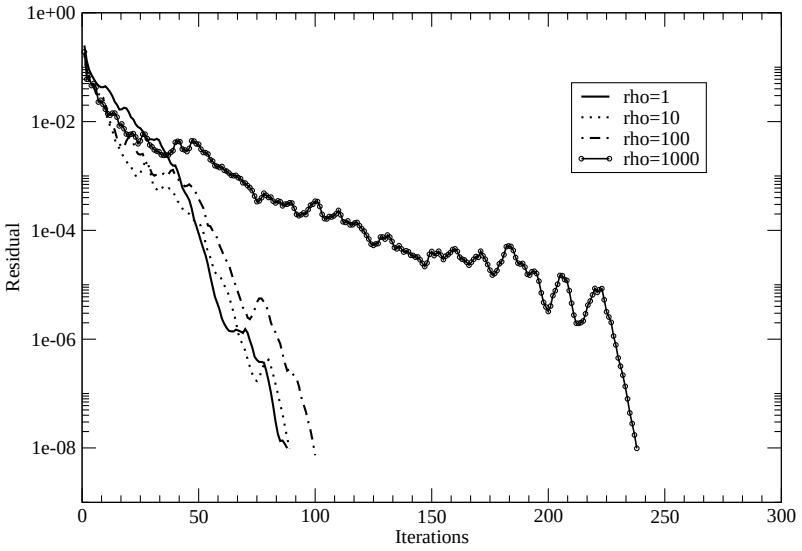


Figure 4.2: Order of accuracy: for SIM and DIM, $\Delta t = 10^{-4}$, time=0.01

boundary conditions on the flow domain. The density ratio affects the convergence of the Poisson solver since it appears as a coefficient of the matrix which must be inverted. In the case of DIM there is a strong dependence on the number of iterations, especially at high density ratios. This is consistent with the results of Tryggvason et al [9], where it was identified that parasitic currents and high density ratio can directly affect the convergence. However for SIM, there is little influence of density ratio on the convergence of the Poisson solver. This may be attributed to the fact that in SIM, there are no parasitic currents around the interface. Since the momentum equations are not modified, unphysical velocities are not generated in SIM. Surface tension is accounted for in the Poisson equation by modifying the right hand side term, and this modification does not affect the convergence.

Fig 4.4 shows the effect of density ratio on the the L_2 norm of pressure. As before, both SIM and DIM cases are carried until time=0.01 with constant time



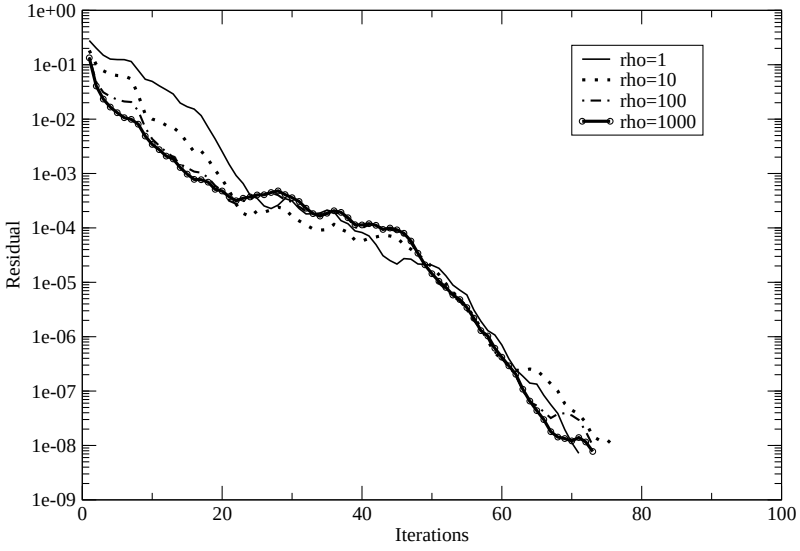


Figure 4.3: Convergence of the pressure Poisson equation on a 60×60 grid with different density ratios, for (a) Distributed Interface Method and (b) Sharp Interface Method.

step of $\Delta t = 10^{-4}$, with a fixed grid size of 60×60. The results are also compared with those of Francois and Shyy [17]. It can be observed that for DIM, the error in pressure is independent of the density ratio. This conforms with the results of similar method used by Francois and Shyy. However, for SIM, the error in pressure is dependant on the density ratio. So in the case of SIM, although the density ratio does not affect the convergence of the Poisson solver, it does affect the error in pressure. This may be attributed to the discretization scheme for the advecting terms in the Navier-Stokes equations. Density must be resolved to within a cell using the sub-cell parameter and then a weighted average of density is taken at the points where density is not defined. It may however, be observed that SIM is more accurate than DIM and the results of Francois and Shyy. This proves the major advantage of SIM - it is faster, more accurate, and does not need highly resolved grids when compared with DIM.

Fig 4.5 shows the error in curvature for both DIM and SIM. It can be observed that the error in curvature is higher for SIM than DIM. This may be attributed again to the error in the evaluation of density to sub-cell resolution. Curvature is greatly affected by the bubble mass conservation and shape correction scheme. The error in density resolution can affect the mass of the bubble and thereby cause greater error in the interface curvature.

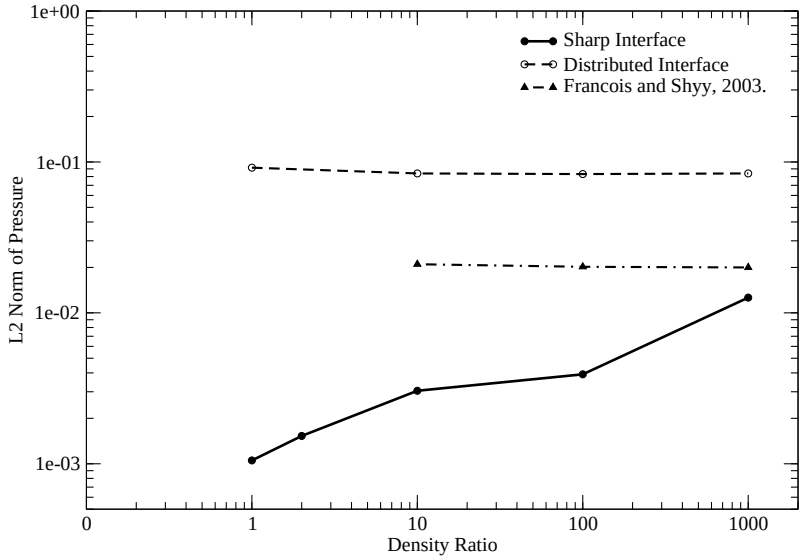


Figure 4.4: L_2 norm of pressure as a function of the density ratio, $\Delta t = 10^{-4}$, time=0.01, grid 60×60.

4.2 Buoyancy driven vapor bubble in a viscous quiescent fluid

Reynolds number and Bond number (Weber number) are the critical parameters which determine the jump discontinuity to be imposed explicitly in SIM. In the previous example, the Reynolds numbers was set as $Re = \infty$ and the code was validated. In the next example, an actual physical case is studied with a vapor bubble rising steadily through a quiescent liquid due to buoyancy, with finite values for Re and Bo. Consider a bubble of initial diameter $D = 1$ which is trapped inside a fluid with a density ratio $\rho_v/\rho_l = 0.01$ and viscosity ratio of $\mu_v/\mu_l = 1.0$. The Reynolds number is $Re = 100$, Bond number is $Bo = 200$ and Froude number is $Fr = 1$. The bubble shape evolution with time is tracked dynamically and compared with the results of Sussman et al [15]. The results are presented for SIM simulation with

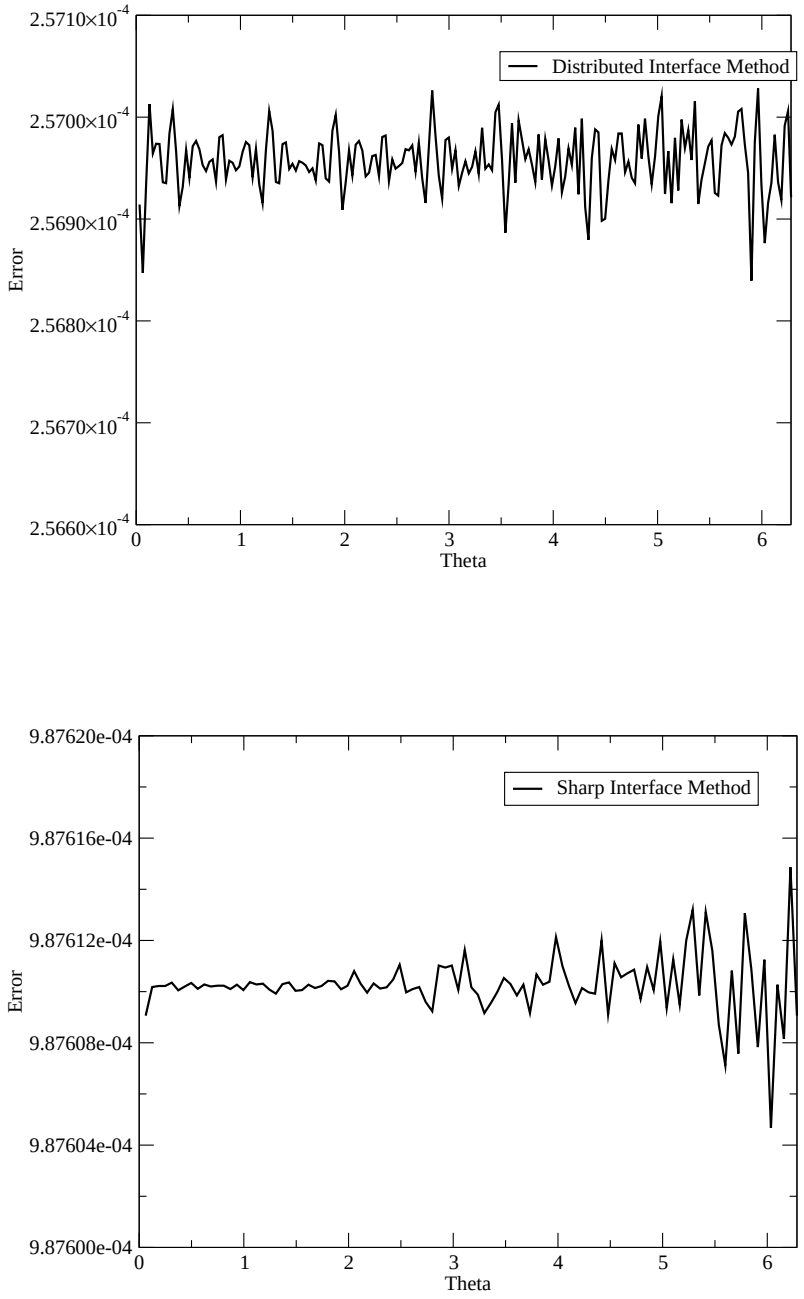


Figure 4.5: Error in curvature on a 60×60 grid, with density ratio=100, for (a) Distributed Interface Method and (b) Sharp Interface Method.

a grid size of 144×144 and the simulation was performed until time $t = 4.40$. The bubble shape and evolution characteristics agree well with the results published by Sussman et al. [15]. The aspect ratio (ratio between the bubble neck width and depth) at steady state was found to be 1.67 by Sussman et al. In the present study it is 1.56, with an error of about 6.6%

In the next case, the effect of surface tension is investigated by changing the Bond number to $Bo = \infty$. This is the case when the surface tension between the liquid and vapor phases becomes negligible. In order to compare the numerical results with those of Sussman et al, the density ratio in this case is $\rho_v/\rho_v = 0.2$ and the Reynolds number is $Re = 100$. The simulation is done on a grid size of 72×72, to match the results of Sussman et al. [15]. Fig 4.7 shows the evolution of bubble with time. Using the bubble neck aspect ratio, the error was found to be 5.4% when compared with the results of Sussman et al.

4.3 Merger of two bubbles

Bubble merger and fracture processes are known to be quite complicated, since the mechanisms which govern these processes are not completely understood. Apart from being highly dependent on the grid resolution, these processes also require some explicitly stated criteria to determine merger or break up of the bubble. In many cases, the length and time scales associated with such criteria are assumed to be of the order of a few grid cells, however, this may be beyond the continuum level in certain cases [8]. With front tracking methods, the treatment of merger or

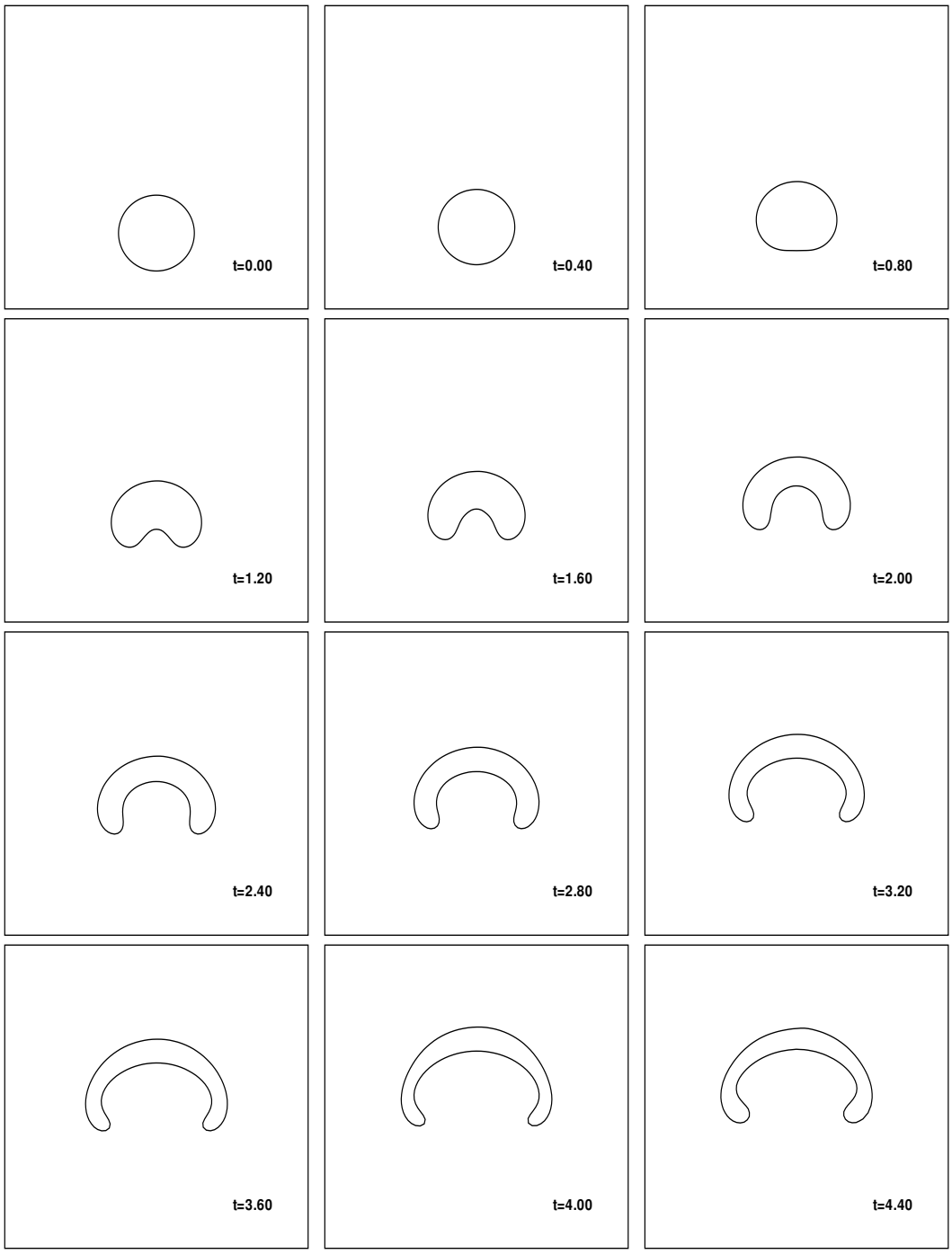


Figure 4.6: Evolution of the interface shape with time for Re=100, Bo=200, Fr=1, grid 144×144.

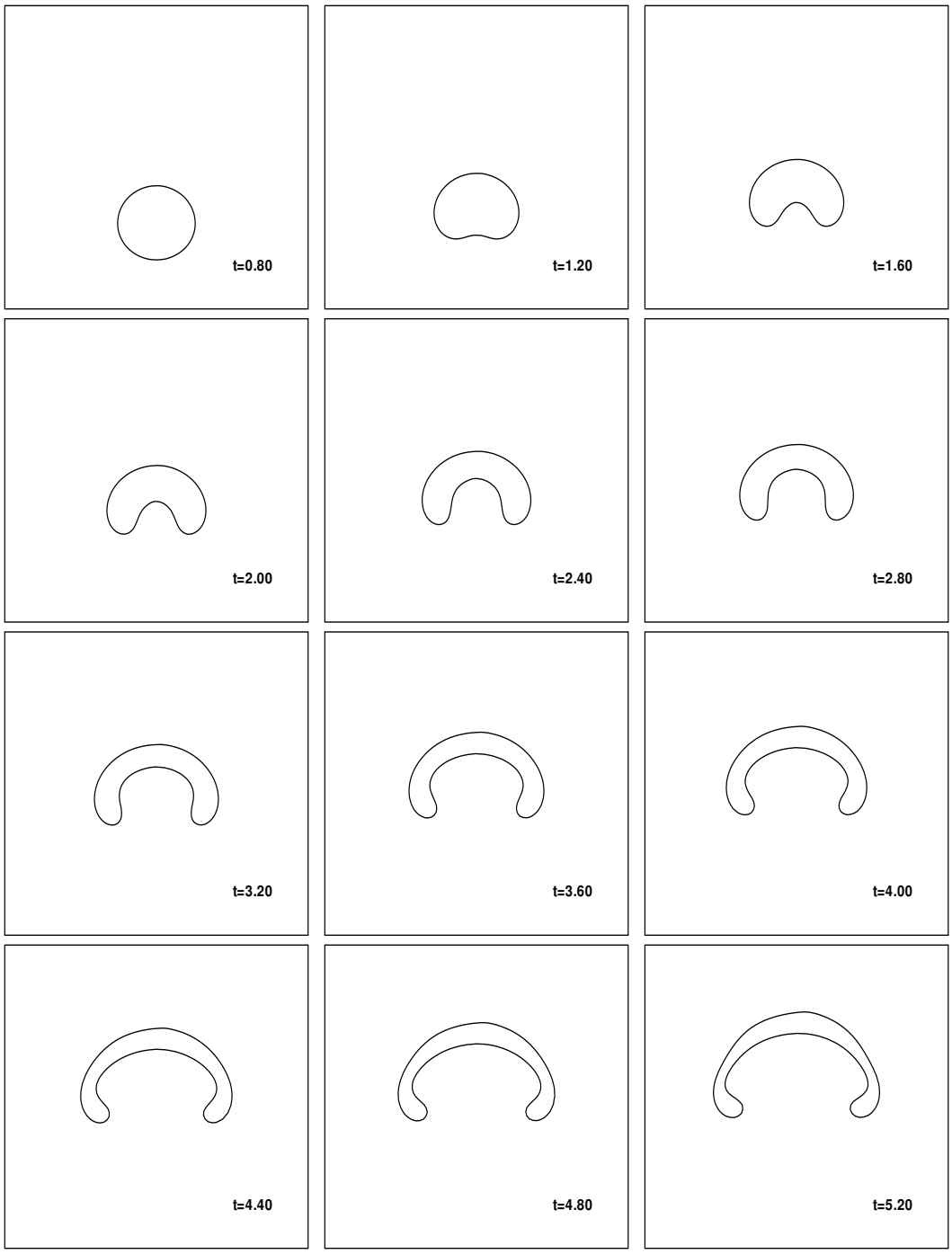


Figure 4.7: Evolution of the interface shape with time for Re=100, Bo=∞, Fr=1, grid 72×72.

fracture is much more difficult. In front tracking methods, the marker points are connected linearly and any changes in the front structure must be accomplished by appropriately changing the way the marker points are connected. The complexity of this operation is often cited as a major disadvantage of the front tracking methods [9].

Bubble merger must be differentiated from bubble collision. Consider two distinct bubbles in the flow field, under the influence of gravity and buoyancy. The presence of the leading bubble causes the second bubble to drift into the wake left behind the first bubble due to its motion upwards. If this wake drift is strong enough, it causes the two bubbles to collide. However, not all the bubbles that collide can merge together into a single bigger bubble. If the bubble overlap in time and space is longer than a critical fraction, then merger occurs. This critical fraction for minimum overlap in time and space is dependent on several parameters including the Reynolds number, surface tension (Weber number or Bond number), viscosity, and density ratios.

Bubble fracture has been found to be much more complicated than merger. This is because the length scales involved in the fracture process are often in the range of a few hundred angstroms [9]. The continuum laws are not applicable at such scales [28] and computational limitations do not permit such highly resolved simulations. Past simulations of bubble fracture were entirely based on continuum level assumptions that the fracture occurs if the interface deforms closer than two grid cells. However with most simulations using DIM and density distribution around a radius of two grid sizes, the validity of such a fracture model is questionable. Several

studies are currently focused on this issue and some recent publications in this field have been primarily by Esmaeeli and Tryggvason [29]

In the present simulation, bubble merger is accomplished similar to the one proposed by Tryggvason et al. [9]. The first stage is to identify the location on the interface where the merger happens. All the marker points, which are within a certain tolerance limit, are identified. Then the shortest distance between the two bubbles is evaluated based on the location of these marker points. The next stage is to declare the merger at this point by re-defining the density distribution and integrating the marker points into a single interface. Numerical implementation of this process is not only expensive in terms of computational power, but also quite difficult because of the interface smoothing function. It may be recalled that the immersed interface method requires explicit monitoring of the mass of the bubble to ensure that apart from global mass conservation, the mass of the bubble is also conserved. This is accomplished by the interface smoothing function. However, in the context of bubble merger, such a smoothing process causes convergence problems. This can be avoided by addition or removal of marker points near the interface where merger happens. The interface is smoothed out at these points using cubic spline re-distribution of the marker points.

Three different cases are identified here:

	Re	We	ρ_l/ρ_v	Initial Separation
Case 1	100	4	2	$2\Delta x$
Case 2	100	4	100	Δx
Case 3	100	∞	100	Δx

- Case 1: In this case, the location of the two bubbles is initialized such that the shortest distance between the two interfaces is $2\Delta x$. The density ratio is very small and the surface tension is very large (small Weber number). The density distribution is plotted with time in Fig 4.8. In this case, it is observed that the wake drift produced by the leading bubble is not very large and the two bubbles diverge from each other. This can be attributed to two reasons. First, since the surface tension is high, the deformation in the bubble shape is not significant. Second, since the density ratio is not very large, the wake drift is not strong enough to pull the second bubble into the wake of the leading bubble. This divergence in bubble motion prevents bubble collision, hence no bubble merger is expected in this case

- Case 2: At the initial time, the two bubbles are located such that the shortest distance between the two bubbles is Δx, while the density ratio and the surface tension are relatively large. The density distribution is plotted at various times in Fig 4.9. It is observed that the wake drift produced by the leading bubble is strong enough to pull the second bubble into the wake, however the collision overlap time is not sufficient to cause a merger. Since the surface tension in this case is quite large, the two bubbles behave like elastic bodies which deform upon collision, but retain their individual identity. During the collision process, momentum is transferred from one bubble to another and the second bubble rebounds upon impact with the leading bubble. In some cases, this collision is so strong that it produces a motion in the downward

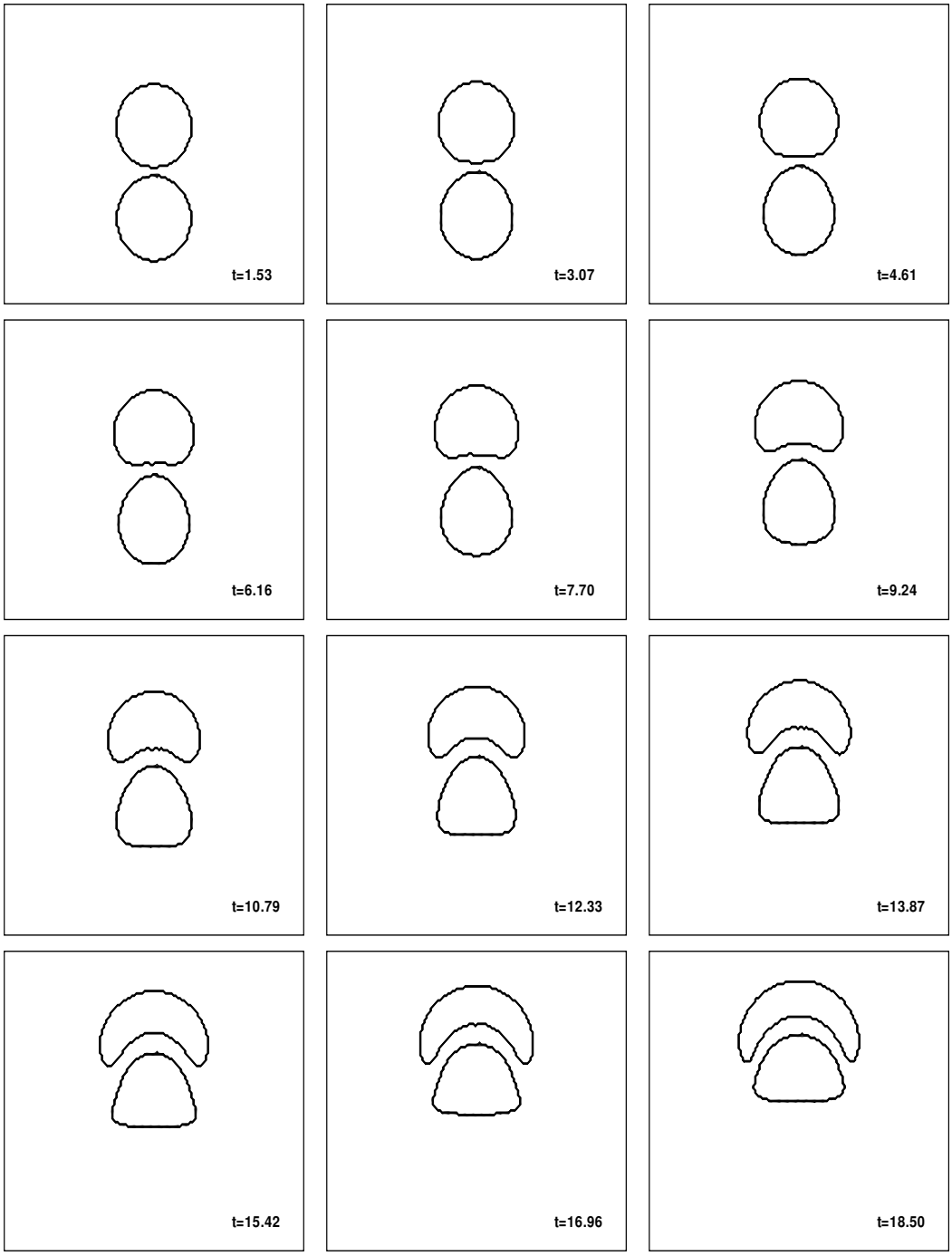


Figure 4.8: Interface shape obtained through density distribution for Re=100, We=4, Density ratio=2, grid 72×72.

direction, against the buoyant force. This case highlights the importance of surface tension and collision overlap time in the merger process.

- Case 3: In this case, all parameters are kept constant like in Case 2, but the surface tension is changed to a very low value, the Weber number being $We = \infty$. The density distribution is shown in Fig 4.10. Just like the previous case, the wake drift produced by the leading bubble is strong enough to pull the second bubble close to the first bubble. Once collision occurs, merger occurs since the surface tension is very small. During this process of merger, the interface ruptures and a leak develops across the interface causing mass exchange between the two bubbles as may be observed through the density distribution. The interface rupture is accomplished by identifying the closest marker points of each of the bubble and vanishing the surface tension to zero at, and around, these marker points. The second bubble now merges completely into the leading bubble and the single big bubble now evolves in such a way to conserve the total momentum of the two bubbles separately. The evolution of the bubble shape is quite interesting and very similar to the interface shape for a Rayleigh-Taylor instability shown by Tryggvason et al [9]

The above cases highlight the effect of density ratio (which governs the wake drift) and the surface tension (which governs the shape deformation and interface rupture during collision) on the bubble merger process. It may be noted here that in all the above cases, density distribution is used to represent the two bubbles. Interface location does not necessarily give the right idea about merger process as

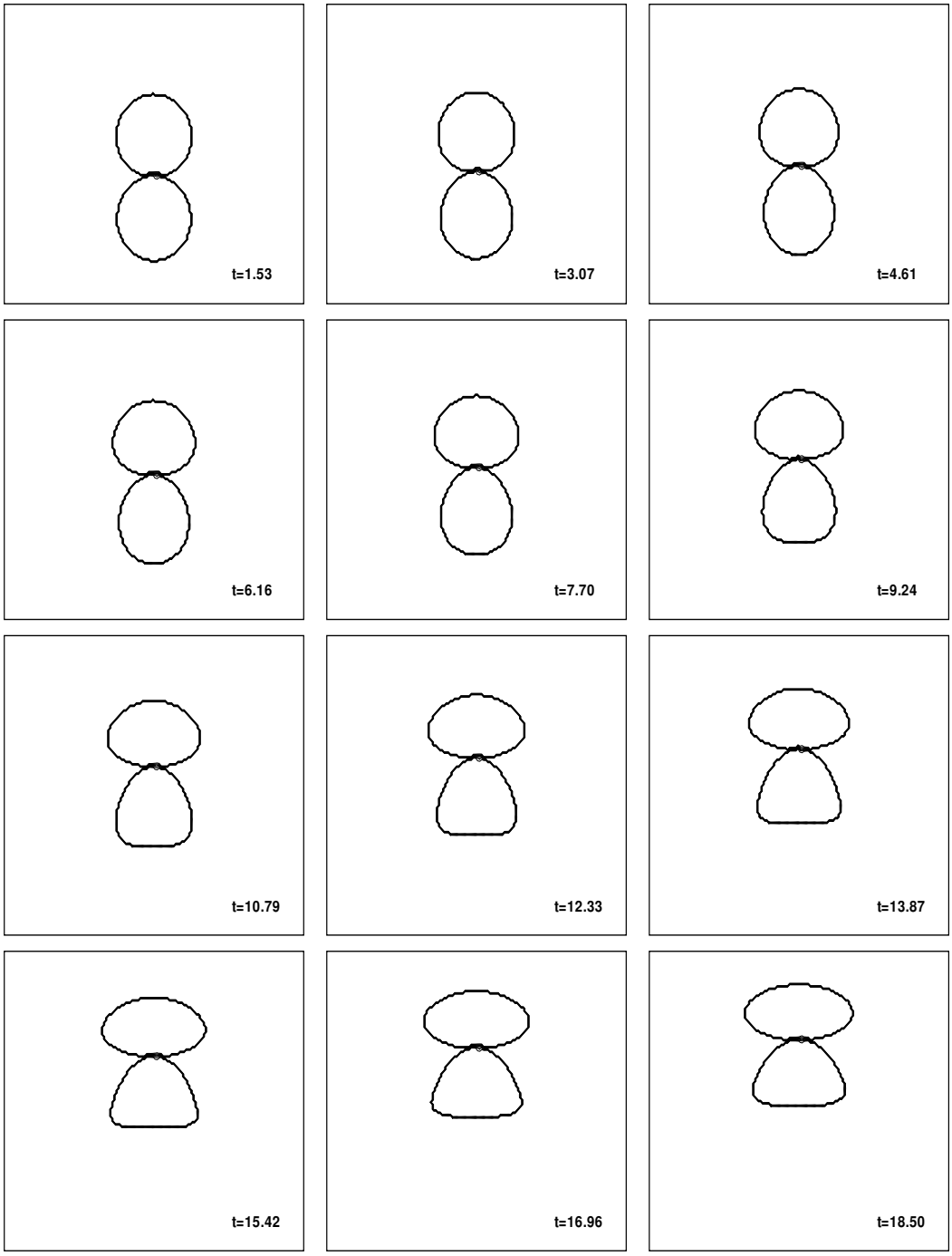


Figure 4.9: Interface shape obtained through density distribution for Re=100, We=4, Density ratio=100, grid 72×72.

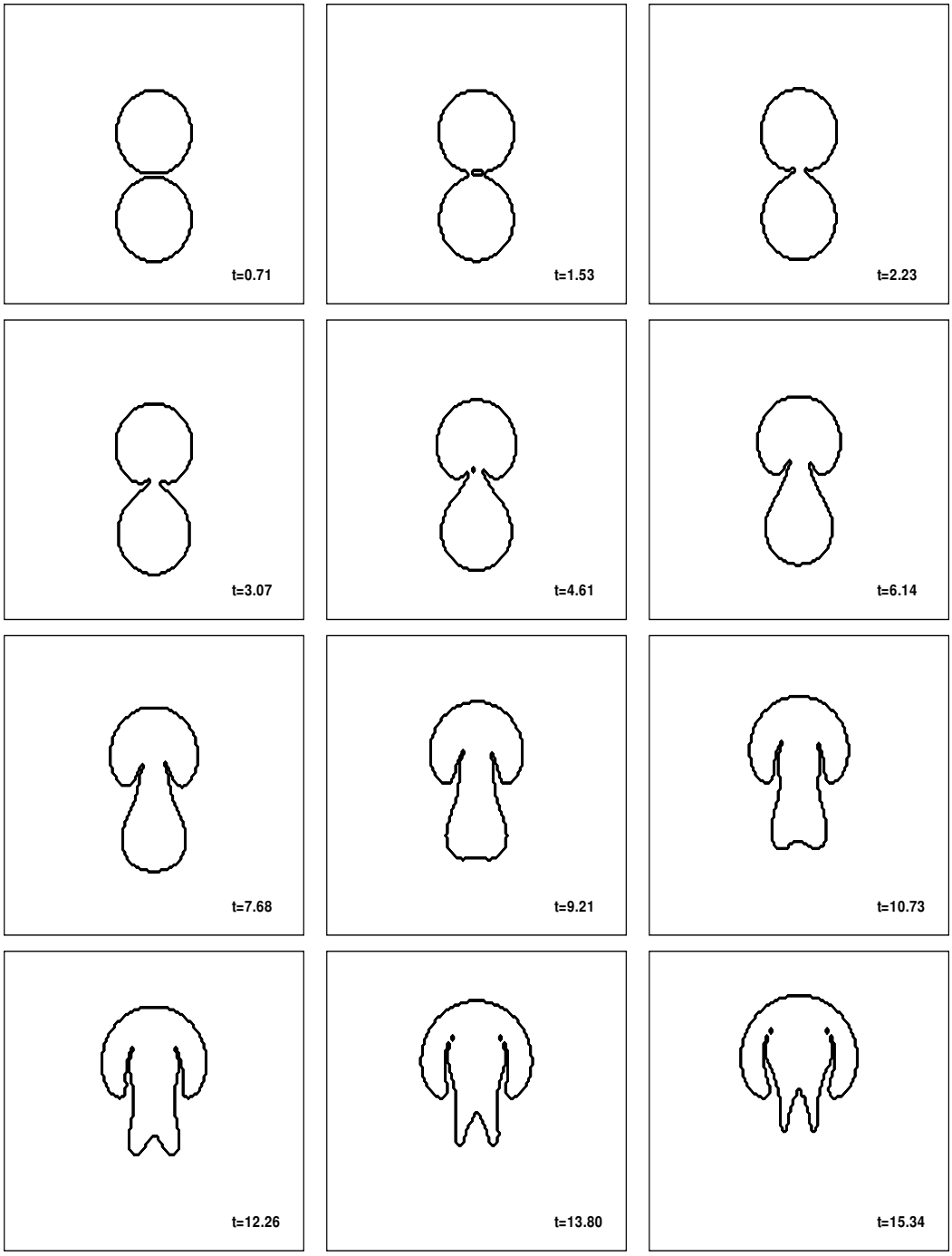


Figure 4.10: Interface shape obtained through density distribution for Re=100, We=∞, Density ratio=100, grid 72×72.

also pointed out by Unverdi and Tryggvason [18]. The interface location as indicated by the marker point distribution is shown in Fig 4.11. It may be observed that this does not indicate merger since surface tension disappears near the overlap region and the marker points do not capture it.

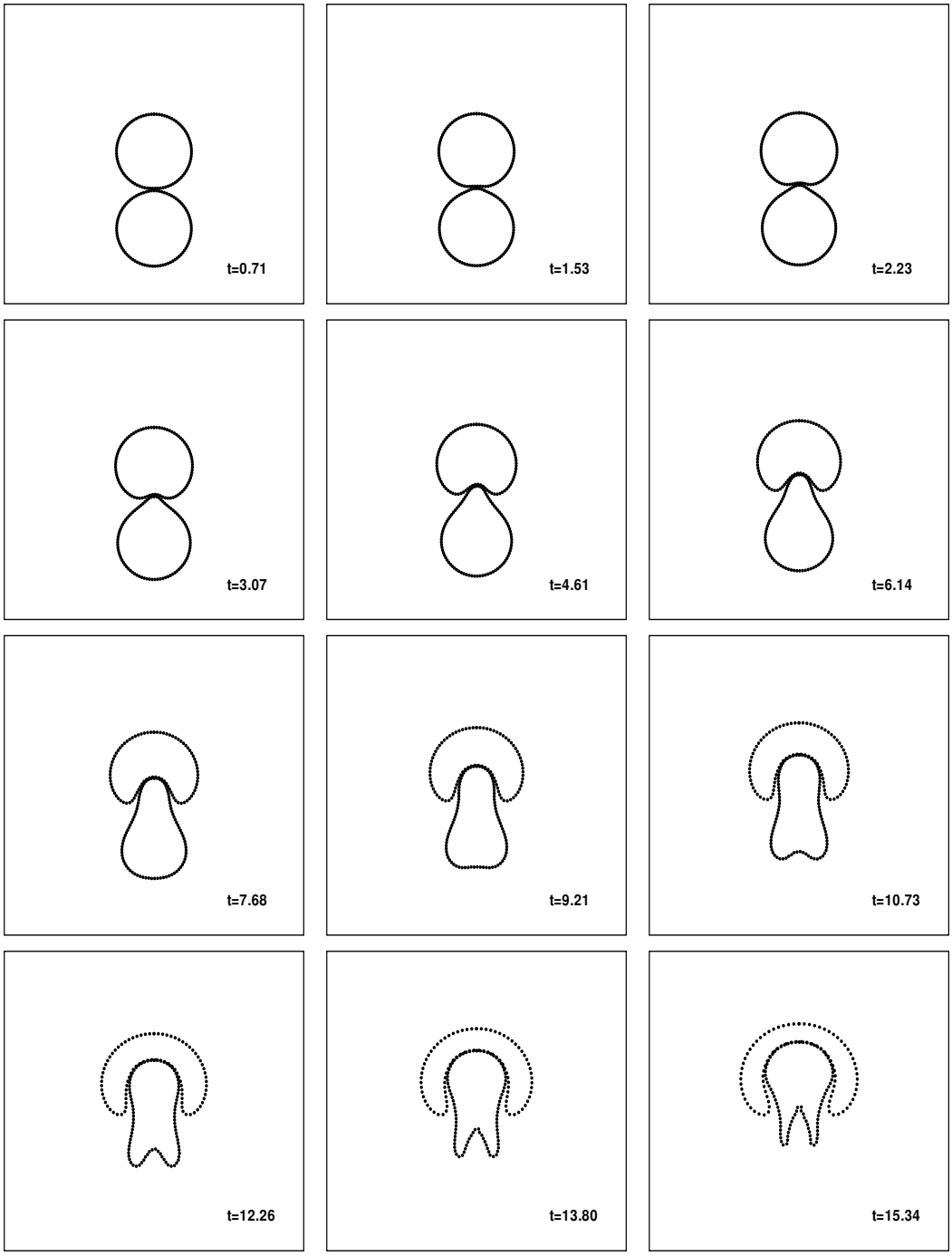


Figure 4.11: Interface shape obtained through marker point location for Re=100, We=∞, Density ratio=100, grid 72×72.

Chapter 5

Conclusions and future work

5.1 Conclusions

The major contribution of this work is that a new sharp interface method is developed, which can capture the actual physics of multiphase flows more accurately. It is also shown that this method is much faster and more accurate than any distributed interface method or other sharp interface methods developed earlier. The main conclusions from the current work are:

1. Using a single set of governing equations for the entire flow field, a new sharp interface method is developed to simulate multiphase flows. The governing equations are discretized in a 2D Cartesian system using a finite volume method. The discrete equations are solved using a fractional step method with a Runge-Kutta time advancement scheme and central differencing for spatial coordinates. This method is found to be computationally less expensive than other schemes developed earlier by other researchers.

2. Using a distributed interface method, the code is validated with analytical results and other numerical studies published in the literature. DIM is implemented by representing the interface with marker points interconnected to form the moving front. The material properties like density, viscosity and sur-

face tension are distributed around the interface, to give a smooth variation from one fluid to another. This model is an indirect approach and it has been shown to produce parasitic currents around the interface. It is also shown that this model is sensitive to convergence problems and computationally expensive at high density ratios.

3. The problem is also solved using a direct approach with the sharp interface method. An advanced Poisson solver is developed with the ability to solve multiphase flow problems with variable coefficients and discontinuities across the interface. It is shown that this Poisson solver is accurate and very fast compared to similar solvers for a distributed interface method. This variable coefficient Poisson solver is used as a black box solver in the Navier-Stokes fractional step method.

4. A single bubble rising in a quiescent fluid under the influence of buoyancy is studied and the results are compared with those published in literature. Simulations are done with DIM, SIM and compared with the analytical results and numerical results. It is observed that the results of SIM are not only more accurate, but also that the method itself is very efficient since there is no need for highly refined grids.

5. The bubble merger process is simulated and the effect of wake drift and surface tension on the bubble collision process is studied. Wake drift, bubble collision and bubble merger processes are identified separately. The merger process is implemented in two stages by using the density distribution as an indicator

function. In the first stage, the points where merger occurs is identified and in the second stage, actual merger is accomplished by removing the surface tension and modifying the density distribution.

5.2 Future Work

1. Extension of the current 2D simulations to 3D is perhaps the most important advancement for this work. With a 3D model, it would be possible to study the effect of multiple bubbles in collision, merger, or fracture. One of the most practical applications of a such 3D model would be simulations of bubble nucleation and departure from the surface of a heater array under various conditions of subcooling and superheating similar to the studies done by Yerramilli et al [30] and Myers et al [31, 1]. Development of such a 3D model is quite challenging, especially when using a front tracking scheme. Tryggvason et al. [9] have described the 3D front tracking scheme is great detail. Apart from the implementation difficulties, the execution of such advanced codes is also challenging, since 3D numerical codes are time consuming.

2. Breaking waves, break up of laminar jets or thin films and effect of surfactants can also be taken up without any significant change in the present code. When the surface tension is a variable, it is a simple case to implement as the surface tension can be prescribed as a simple function along the bubble surface. Breaking waves are currently being investigated experimentally at the University of Maryland. It would be interesting to compare the numerical

simulations with the experiments.

3. Another important and necessary extension of current work is inclusion of energy equation and mass transfer into the model. Mass transfer is a very challenging in terms of implementation since the fractional step method has to be modified to include mass transfer across the interface. Several researchers approached this problem by implementing the mass transfer iteratively. Since most of the solutions to multiphase flows are weak formulations, iterative techniques are the most common approaches towards simulating mass transfer. A more detailed discussion on implementation of such a technique can be found in an article by Udaykumar et al [28].

4. Physical instabilities like Rayleigh-Taylor and Kevin-Helmholtz instability can also be studied with little modification in the present code. Oscillating bubbles damped out due to viscous forces is another classic example which can be easily studied.

Appendix A

User guide for the two-phase flow solver

In this appendix, a short description of the two-phase flow solver is given. First DIM solver is discussed followed by SIM. For a better understanding of the solver, it is recommended that the user use this guide in conjunction with the FORTRAN codes in the accompanying CD-ROM. The codes were compiled using PGF-77 compiler on a LINUX machine running SuSE 9.1 Pro.

A.1 Distributed Interface Method

The description of the mathematical model, numerical scheme and algorithm is given in the thesis. In this section, the general flow of information through each subroutine of the program is discussed.

The MAKEFILE shows the compiler arguments and the source code files used in the program. The main program in "NSus.f" which stands for Navier-Stokes unsteady solver. Each sub-routine is written in a separate file and linked to the main project to enable easy access. Each file and its contents are discussed below.

- NSus.f: This is the main parent program based on 2D Navier-Stokes solver. It was developed using finite volume method on a staggered uniform grid using second-order central differencing for spatial coordinates and third-order Runge-Kutta scheme for time advancement in a fractional step scheme. This parent program begins with declaration of all variables used in the program. Care has been taken to represent variables with names which can signify their usage. For example, u, v and p stand for the two velocities and pressure at current time step while variables like $u0$ stand for values at previous step. Density and viscosity are represented as rho and mu. Distributed surface tension force is represented by $fexty$ and $fextx$ meaning external forces alone y and x directions. All the program parameters are read from an external file called "input.dat" and the numbers in the file are self explanatory. Once the parameters are known, grid is generated using "gridgen" subroutine and the grid is tested for the maximum and minimum size. This is followed by initialization

of the variables and the interface. The interface location is specified using the "ObjectGrdWann" subroutine which distributes the marker points around a circle given the location of the center of the circle and the radius. The next step is to do time advancement inside an R-K scheme. Within the R-K scheme, the first step is to evaluate the flags. If the location of the interface is known with respect to the fixed grid, then the flags can be evaluated using the immersed boundary method developed by Balaras [20]. Using the marker points, surface tension is distributed as a delta function around the interface. This is done in the subroutine "Heaviside". Then, density and viscosity are also distributed on to the fixed grid using the subroutine "property". The matrix coefficients required for inversion of the Poisson equation are evaluated next using the subroutine "matrixcoeff". This is followed by predictor-corrector scheme which has been developed and validated separately. Once the corrected velocities are also evaluated, the interface is advected through the subroutine "movemarker". Depending on the output required, the postprocessing part has been modified to suit the application.

- ObjectGrdWann.f: This subroutine creates the interface by locating the marker points along the arc length using simple trigonometry. Then polynomial coefficients are evaluated using "polcoe" which is a standard function in numerical recipes.

- bpoints.f: This subroutine takes the location of the interface in terms of arc length parameter and the location of marker points. It then evaluates the

cluster of fixed grid points around each marker point. Using trial and error method, it evaluates the shortest normal passing through the interface from each of the 2×2 cluster. Then the flags are evaluated as: $+1$ if the grid point is inside the interface, 0 if grid point is a true boundary point and -1 is it is outside the interface. The curvature is also evaluated at each marker point. The flags and curvature are critical input parameters for the next subroutines.

- Heaviside.f: This subroutine evaluates the surface tension force and distributes it onto the fixed grid by constructing a local δ-function around each marker point. It takes in the information about location of interface, magnitude of surface tension and outputs the vector field of distributed surface tension body force. It may be noted that surface tension is an additive force, i.e., two marker points can add surface tension at the same grid point.

- property.f: Density and viscosity are distributed around the interface by using this subroutine. Here also, local δ-function is constructed around the interface for distribution. Tryggvason's and Peskin's scheme have both been used for this purpose. Material properties are not additive, hence density distribution overwrites the previous values at a grid point.

- matrixcoeff.f: Since material properties are spatial variables, the coefficients of the pressure Poisson equation will be dependent on the coordinates. In this subroutine, the matrix coefficients are evaluated.

- predictor.f: In this subroutine, the predicted velocities are evaluated. The

code is self explanatory and it may be noted that the surface tension vector field is added as an external force in this step.

- bcond.f: Domain boundary conditions (homogeneous Neumann BC) are imposed on the predicted velocities.

- divuv.f: Once the predicted velocities are evaluated, the divergence of the predicted velocity field is evaluated and this will be the RHS term of the pressure Poisson equation.

- pbound.f: The domain boundary conditions are implemented through modification of the coefficients of the Poisson solver near the boundary. The RHS for the Poisson equation is the divergence of the predicted velocity.

- cgs.f: This is the iterative pressure Poisson solver, very often the rate determining step in the entire N-S solver. In this current scheme, a simple conjugate gradient solver has been used. The source code has been developed and tested by Ferziger and Peric.

- corrector.f: Once the pressure has been obtained, the actual velocity field can be recovered by the correction step. This completes the time advancement from initial time step to the next.

- movemarker.f: Once the velocity field has been solved for, the interface must be advected. This is done by constructing local δ-function around the interface and evaluating the velocity of the marker points as a weighted sum of the velocity of surrounding grid points. Once the new location of the marker points

has been evaluated, the arc length parameter is evaluated by solving the two quadratic polynomials simultaneously. Typically this must be followed by the shape correction subroutine, but most simulations did not require this step, hence this has been omitted in the discussion. A simple way to accomplish shape correction would be to use a bisection method available in standard numerical recipes.

- calcCFL.f: In this subroutine, the time step is dynamically changed to maintain the stability of the simulation. For more detailed description please see the relevant part in the thesis.

A.2 Sharp Interface Method

SIM is quite similar to DIM in its structure. The only difference is in the way material properties and surface tension are dealt with. The primary differences in these two codes are discussed here.

- bpoints.f: Apart from evaluating the flags, the local information of the interface is also evaluated. The intersection of the interface along each grid direction, the local normals, and the curvature are stored in the variables 'simbx' and 'simby'.

- evalrhs.f: This is the most critical part in development of the variable coefficient Poisson solver. The formulation closely represents that of Berthelsen [26]. Only an inviscid formulation has been highlighted in the attached codes, to focus the attention on the development of the solver. A simple modification of the RHS would give the formulation for a viscous droplet.

- property.f: Based on the flags, the properties are evaluated. Since there is not distribution in this case, the material property evaluation is trivial.

A.3 Further information

The source codes are mostly self-explanatory. For a description of the algorithm and mathematical model, the reader is referred to the thesis and the literature cited in the relevant sections. In case of further questions, doubts or clarifications, the author may be reached by email at vamsee.ky@gmail.com.

BIBLIOGRAPHY

[1] J. G. Myers, V. K. Yerramilli, S. W. Hussey, G. F. Yee, J. Kim, "Time and space resolved temperature and heat flux measurements during nucleate boiling with constant heat flux boundary conditions," Int. J. of Heat and Mass Transfer, **48**, 2429 (2005).

[2] W. M. Rohsenow, "A method of correlating heat-transfer data for surface boiling of liquids," ASME J. of Heat Transfer, **74**, 969 (1952).

[3] B. B. Mikic, W. M. Rohsenow, "A new correlation of pool-boiling data including the effect of heating surface characteristics," ASME J. of Heat Transfer, **91**, 245 (1969).

[4] K. Stephan, M. Abdelsalem, "Heat transfer correlations for natural convection boiling," Int. J. of Heat Mass Transfer, **23**, 78 (1980).

[5] , D. E. Forster, R. Grief, "Heat transfer to a boiling liquid - mechanism and correlation," ASME J. Heat Transfer, **81**, 43 (1959).

[6] N. Zuber, "Hydrodynamic aspects of boiling heat transfer," PhD Thesis, University of California, Los Angeles, (1959).

[7] D. Juric, G. Tryggvason, "Computations of boiling flows," Int. J. Multiphase Flow, **24**, 387 (1998).

[8] T. Ye, W. Shyy, and J. N. Chung, "A fixed-grid sharp interface method for bubble dynamics and phase change," J. Comp. Phys. **174**, 781 (2001).

[9] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W.Tauber, J. Han, S. Nas, and Y. J. Jan, "A front tracking method for computations of Multiphase Flow," J. Comp. Phys. **169**, 708 (2001).

[10] G. Ryskin and L. G. Leal, "Numerical solution of free-boundary problems in fluid mechanics, Part 1, Part 2, Part 3," J. Fluid Mech., **148**, 1 (1984).

[11] D. S. Dandy, L. G. Leal, "Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate Reynolds number," J. Fluid Mech. **208**, 161 (1989).

[12] M. Kang, R. P. Fedkiw, and X. Liu, " A boundary condition capturing method for multiphase incompressible flow," J. Sci. Comp. **15**, 323 (2000).

[13] X. Liu, R. P. Fedkiw, and M. Kang, "A boundary condition capturing method for poisson's equation on irregular domains," J. Comp. Phys. **160**, 151 (2000).

[14] J. U. Brackbill, D. B. Kothe and C. Zemach, " A continuum method for modeling surface tension," J. Comp. Phys. **100**, 335 (1992).

[15] M. Sussman, P. Smereka, and S. Osher, "An level set approach for computing solutions to incompressible two-phase flow," J. Comp. Phys. **114**, 146 (1994).

[16] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows," Comp. and Fluids **27**, 663 (1998).

[17] M. Francois, and W. Shyy, "Computations of drop dynamics with the immersed boundary method part 1: numerical algorithm and buoyancy induced effect," Num. Heat Transfer, Part B. **44**, 101 (2003).

[18] S. O. Unverdi and G. Tryggvason, " A front-tracking method for viscous incompressible multi-fluid flows," J. Comp. Phys. **100**, 25 (1992).

[19] C. S. Peskin, "Numerical analysis of blood flow in the heart," J. Comp. Phys. **25**, 220 (1977).

[20] E. Balaras, "Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations," Computers and Fluids, **33**, 375 (2004).

[21] A. J. Chorin, "Numerical solution of the Navier-Stokes equations," Math. Comput. **22**, 745 (1968).

[22] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics* (Springer, New York 2002).

[23] S. V. Patankar, *Numerical heat transfer and fluid flow* (Taylor & Francis, Washington, DC, 1980).

[24] B. P. Leonard, "A stable and accurate convective modeling procedure based on quadratic ppstream interpolation," Computer Methods in Applied Mechanics and Engineering, **19**, 59 (1979).

[25] W. Mulder, S. Osher and J. A. Sethian, "Computing interface motion in compressible gas dynamics," J. Comp. Phys. **100**, 209 (1992).

[26] P. A. Berthelsen, "A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions," J. Comp. Phys. **197**, 364 (2004).

[27] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, "An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries," J. Comp. Phys. **156**, 209 (1999).

[28] H. S. Udaykumar, H. Kan, W. Shyy, and R. Tran-Son-Tay, "Multiphase dynamics in arbitrary geometries on fixed cartesian grids," J. Comp. Phys. **137**, 366 (1997).

[29] A. Esmaeeli and G. Tryggvason, "Computations of film boiling," Int. J. of Heat and Mass Transfer, **47**, 5451 (2004).

[30] V. K. Yerramilli, J. G. Myers, S. W. Hussey, G. F. Yee, J. Kim, "Time and space resolved heat flux measurements during nucleate boiling with constant heat flux boundary conditions," Int. Conference of Multiphase FLows, Yokohama, Japan (2004).

[31] J. G. Myers, V. K. Yerramilli, S. W. Hussey, G. F. Yee, J. Kim, "Time and space resolved temperature measurements during nucleate boiling with constant heat flux boundary conditions," Proceedings of the 2004 ASME Heat Transfer Fluids Engineering Summer Conference, Charlotte, North Carolina, USA, 2004.