ADAPTIVE STABILIZED FINITE ELEMENT ANALYSIS OF MULTI-PHASE FLOWS USING LEVEL SET APPROACH

By

Sunitha Nagrath

A Thesis Submitted to the Graduate Faculty of Rensselaer Polytechnic Institute in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major Subject: Mechanical Engineering

Approved by the Examining Committee:

Kenneth Jansen, Thesis Adviser

Richard T. Lahey, Jr., Thesis Adviser

Mark Shephard, Member

Donald Drew, Member

Rensselaer Polytechnic Institute
Troy, New York

March 2004 (For Graduation May 2004)
ADAPTIVE STABILIZED FINITE ELEMENT ANALYSIS OF MULTI-PHASE FLOWS USING LEVEL SET APPROACH

By

Sunitha Nagrath

An Abstract of a Thesis Submitted to the Graduate Faculty of Rensselaer Polytechnic Institute in Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY
Major Subject: Mechanical Engineering

The original of the complete thesis is on file in the Rensselaer Polytechnic Institute Library

Examining Committee:

Kenneth Jansen, Thesis Adviser
Richard T. Lahey, Jr., Thesis Adviser
Mark Shephard, Member
Donald Drew, Member

Rensselaer Polytechnic Institute
Troy, New York

March 2004
(For Graduation May 2004)
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>ix</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Sonoluminescence</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Existing Numerical Methods</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Resolving an interface</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Overview</td>
<td>6</td>
</tr>
<tr>
<td>2. The Level Set Method</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Level set description</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Interface thickness</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 Re-initialization or re-distancing of level sets</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Modeling of surface tension</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Chapter summary</td>
<td>12</td>
</tr>
<tr>
<td>3. The Finite Element Formulation of the Flow</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Compressible flow formulations</td>
<td>13</td>
</tr>
<tr>
<td>3.1.1 Compressible strong Form</td>
<td>13</td>
</tr>
<tr>
<td>3.1.2 Weak form-finite element discretization</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2.1 Discontinuity capturing operator</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Incompressible flow formulations</td>
<td>18</td>
</tr>
<tr>
<td>3.2.1 Incompressible flow</td>
<td>18</td>
</tr>
<tr>
<td>3.2.2 Weak form-Finite element discretization</td>
<td>19</td>
</tr>
<tr>
<td>3.3 Chapter summary</td>
<td>21</td>
</tr>
<tr>
<td>4. Finite Element Formulations for Level Sets</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Discretization</td>
<td>22</td>
</tr>
<tr>
<td>4.2 Redistancing - Volume constraint</td>
<td>24</td>
</tr>
<tr>
<td>4.3 Chapter Summary</td>
<td>26</td>
</tr>
</tbody>
</table>
5. Ghost Fluid Method ................................................................. 27
   5.1 Description ................................................................. 27
   5.2 Two-phase shock tube problem ........................................... 30
   5.3 Chapter summary ............................................................ 33

6. Explicit Solver ................................................................. 34
   6.1 General formulation of Runge-Kutta methods ......................... 34
   6.2 Implementation of fourth order explicit solver ....................... 36
   6.3 Chapter summary ............................................................ 37

7. Adaptive Mesh Refinement .................................................. 38
   7.1 Adaptive strategy .......................................................... 38
       7.1.1 Error estimation: A brief review ................................. 39
       7.1.2 Refinement strategy .................................................. 39
       7.1.3 Error indicators implemented ................................. 40
       7.1.4 Mesh adaptation .................................................. 42
           7.1.4.1 Edge marking ............................................... 42
           7.1.4.2 Mesh quality ............................................... 43
           7.1.4.3 Solution transfer ........................................ 43
       7.1.5 Implementation of adaptive driver ............................... 43
           7.1.5.1 Modification to “NSpre” ................................ 44
           7.1.5.2 Modification to flow solver “Phasta” .................. 45
           7.1.5.3 Modification to “Mesh Adapt” .......................... 45
           7.1.5.4 Pseudo code for the adaptive driver ................. 45
   7.2 Chapter summary ............................................................ 46

8. Incompressible Simulations ................................................ 47
   8.1 Simple advection of a cylindrical bubble .............................. 47
   8.2 Two-dimensional simulation of buoyant bubble motion at a medium Reynolds Number ................................................. 50
   8.3 Effect of Fluid viscosity on buoyant spherical bubble motion ............................. 54
       8.3.1 Formation of a cap bubble at low Reynolds number .... 54
       8.3.2 Formation of a toroidal bubble at high Reynolds number 57
   8.4 Three-dimensional simulation of two bubble coalescence ............ 60
9. Simulation of Compressible flows .............................................. 68
   9.1 Steady shock problem ....................................................... 68
   9.2 Effect of discontinuity capturing operator ............................. 70
   9.3 Adaptive mesh refinement for the shock problem ..................... 70
   9.4 Sod’s shock tube problem ................................................... 71
   9.5 Implosion of a spherical shock wave using an adaptive mesh strategy ... 73
10. Free Surface Flows ............................................................ 80
    10.1 Introduction ................................................................. 80
    10.2 Kelvin-Helmoltz stability criteria for stratified flows ............... 80
        10.2.1 The stability analysis of a full two-fluid model .............. 81
    10.3 Inviscid two-phase flow .................................................. 83
    10.4 Chapter summary ........................................................... 88
11. Study of Rayleigh-Taylor Instabilities ..................................... 89
    11.1 Two-dimensional simulation of Rayleigh-Taylor instabilities ....... 90
    11.2 Three-dimensional simulations of Rayleigh-Taylor instabilities ..... 92
    11.3 Chapter summary ........................................................... 95
12. Hydrodynamic Simulation of Air Bubble Implosion ..................... 97
    12.1 Introduction ................................................................. 97
    12.2 Bubble motion and flow field during the initial stages of implosion ... 98
    12.3 Imploding air bubble in water .......................................... 101
        12.3.1 Hydrodynamics of collapse ...................................... 116
13. Conclusions ........................................................................ 120
## LIST OF TABLES

7.1 Comparison of Adaptive Strategies .......................... 40
## LIST OF FIGURES

5.1 A typical triangular element containing interface ........................................ 28
5.2 A 1-D example of global and local numbering ............................................. 28
5.3 Initial pressure distribution ........................................................................... 30
5.4 Initial velocity distribution ............................................................................ 31
5.5 Pressure distribution at $t = 0.003$ ................................................................. 31
5.6 Velocity distribution at $t = 0.003$ ................................................................. 32
7.1 Adaptive Driver ............................................................................................... 44
8.1 The computational domain and bubble position at $t = 0.$ ............................ 48
8.2 Bubble position at various time instants ....................................................... 49
8.3 The bubble position at $t = 0$ ....................................................................... 50
8.4 Evolution of rising buoyant bubble at $t = 0.05$ ............................................ 51
8.5 Evolution of rising buoyant bubble at $t = 0.15$ ............................................ 52
8.6 Evolution of rising buoyant bubble at $t = 0.3$ .............................................. 53
8.7 Rise of the spherical bubble at low Reynolds number ..................................... 55
8.8 Vertical velocity contours shown on the $xy$-plane (low Re) .......................... 56
8.9 Rise of the spherical bubble at high Reynolds number ................................... 59
8.10 Change in the shape of the interface displayed on a $xy$-plane along with velocity vectors .......................................................... 60
8.11 Initial position of the bubbles ..................................................................... 61
8.12 Position of the bubbles at $t = 0.05$ ............................................................. 61
8.13 Contours of speed on a cut plane at $t = 0.05$ ................................................. 62
8.14 Position of the bubbles at $t = 0.10$ ............................................................. 62
8.15 Position of the bubbles at $t = 0.10$ along with the contours of pressure on a cut plane .............................................................. 63
8.16 About to merge bubbles at $t = 0.15$ .......................................................... 63
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.17</td>
<td>A different view of the bubbles which are about to merge at ( t = 0.15 )</td>
<td>64</td>
</tr>
<tr>
<td>8.18</td>
<td>The interface of about to merge bubbles on a cut plane at ( t = 0.15 )</td>
<td>64</td>
</tr>
<tr>
<td>8.19</td>
<td>Merged bubbles at ( t = 0.22 )</td>
<td>65</td>
</tr>
<tr>
<td>8.20</td>
<td>The interface of merged bubbles on a cut plane at ( t = 0.22 )</td>
<td>66</td>
</tr>
<tr>
<td>9.1</td>
<td>Pressure predictions without the DC operator: exact solution, SUPG solution</td>
<td>68</td>
</tr>
<tr>
<td>9.2</td>
<td>Pressure predictions with the DC operator: exact solution, SUPG with DC operator solution</td>
<td>69</td>
</tr>
<tr>
<td>9.3</td>
<td>Original coarse mesh</td>
<td>71</td>
</tr>
<tr>
<td>9.4</td>
<td>Error distribution along the channel estimated from the solution on coarse mesh</td>
<td>71</td>
</tr>
<tr>
<td>9.5</td>
<td>Refined mesh</td>
<td>72</td>
</tr>
<tr>
<td>9.6</td>
<td>Contours of pressure along the channel on an adaptively refined mesh</td>
<td>72</td>
</tr>
<tr>
<td>9.7</td>
<td>Comparison of the solution obtained by ( h )-refinement solution on finer grid, solution on coarser grid, Exact solution</td>
<td>73</td>
</tr>
<tr>
<td>9.8</td>
<td>Sod’s shock tube problem</td>
<td>74</td>
</tr>
<tr>
<td>9.9</td>
<td>Initial conditions for the imploding spherical shock wave</td>
<td>75</td>
</tr>
<tr>
<td>9.10</td>
<td>Adapted mesh at ( t=0 )</td>
<td>75</td>
</tr>
<tr>
<td>9.11</td>
<td>Spherically inward propagating shock at ( t=0.1sec )</td>
<td>76</td>
</tr>
<tr>
<td>9.12</td>
<td>Mesh adapted to the shock front at ( t=0.1sec )</td>
<td>76</td>
</tr>
<tr>
<td>9.13</td>
<td>Shock propagating towards origin at ( t=0.2sec )</td>
<td>77</td>
</tr>
<tr>
<td>9.14</td>
<td>Adaptively refined mesh at ( t=0.2sec )</td>
<td>78</td>
</tr>
<tr>
<td>9.15</td>
<td>Velocity and pressure distributions at ( t=0.3sec )</td>
<td>78</td>
</tr>
<tr>
<td>9.16</td>
<td>At the instant of reaching origin at ( t=0.4680sec )</td>
<td>79</td>
</tr>
<tr>
<td>9.17</td>
<td>Shock wave reflecting back from origin at ( t=0.6080sec )</td>
<td>79</td>
</tr>
<tr>
<td>10.1</td>
<td>Boundary Conditions</td>
<td>83</td>
</tr>
<tr>
<td>10.2</td>
<td>Geometry</td>
<td>84</td>
</tr>
<tr>
<td>10.3</td>
<td>Mesh used and the interface at ( t=0.008sec )</td>
<td>85</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

My graduate school experience at Rensselaer provided me an opportunity to work with several outstanding people. First and foremost, my thanks to my supervisors Professors Kenneth Jansen and R. T. Lahey, Jr. Working with them was an experience in itself. I consider myself fortunate to interact with such excellent advisers and a diverse group of committee members. Their input has helped me to focus my research.

Working with Prof. Jansen is a true privilege. He taught me the alphabets of CFD both inside and outside of classroom, for which I will be grateful forever. I was truly blessed to have Prof. Jansen as an adviser since he not only is knowledgeable and approachable but is also one of the finest human being. His support and encouragement throughout my graduate studies not only molded my research, but also allowed me to grow as a person. I owe many of my accomplishments to countless positive interactions with him. Not to mention the numerous marathon debugging sessions, which he endured with me patiently. His lucid explanations introduced me to the unexpected charms of CFD. His sincerity, discipline and the hard working attitude has throughout my graduate studies inspired me. I have always admired his quality of perseverance, personal and professional integrity and he is a role model for me. He is the most influential in shaping my career. I will be eternally grateful for all what he has done for me and making my PhD (an odyssey) both exciting and memorable. It was a joy to learn and work with him and words are inadequate to express my indebtedness to him.

I would like to express my most sincere gratitude to Prof. Lahey. He truly steered my thesis research towards a very fruitful and productive direction. His knowledge of multiphase flows is unmatched and the courses I have taken with him were the building blocks for this research. I highly esteem him for his achievements and being successful and yet humble, and working with him has been truly incredible. It is an honor to have him as an adviser. These years of working with him have been a tremendous pleasure, and the one reason was the help and support imparted by him. His enthusiasm and the energy level always made me wonder about the hidden pleasures of dealing with multiphase flows. He always kept me focused on the objective with not letting me get lost in the
numbers. Among many, his insight into the important issues in multiphase flows has been a great asset. His thoughtful and poignant questions were invaluable in my development. His guidance, support, encouragement have been most invaluable in the success of this effort and I profusely thank him for sharing his amazing years of experience.

My research in sonoluminescence has benefited greatly from the constructive ideas and feedback from Prof. I. Akhatov. His input largely helped in the exposition of some of the critical ideas in this area of research. I highly regard his thoughtful suggestions. The discussions and the theoretical analysis sessions I had with him were indispensable for my dissertation. His insights were extremely useful, and the time spent with him was rewarding and I have a deepest appreciation for his help.

I would like to express my sincere thanks to Prof. Shephard for helping and motivating me to learn mesh data base. The courses I have taken with him helped me to develop an understanding and motivated me to work in the filed of adaptive mesh techniques. The advancements and the infrastructure developed by him in the area of adaptive finite element analysis techniques is instrumental to this research. Along with his technical expertise and knowledge in mesh adaptive techniques, his invisible support in the form of providing and maintaining the best possible computational resources at SCOREC was an essential ingredient towards the achievements of this research.

I wish to express my thanks to the committee member Prof. Drew for his input. It was a pleasure to have him on the committee. I have also greatly benefited from various excellent books he has written in the subject of multiphase flows.

A big part of my life is amazing friends. I would like to first mention the friends I have made at Rensselaer. When I started learning CFD, it was Andres who used to be around the lab a lot. Sharing the office with him was advantageous as I did not even had to leave the office to clear all my doubts. The discussions I shared with him were enlightening, and I thank him a lot for it. I have to admit that by having Anil in the office was like having a personal tech support all the time right next to me. My special thanks to Anil for all the help he extended in C programming and making me understand the intricacies of compilation. Also, by talking with him, I kept my mother tongue speaking abilities alive and well. I would also like to thank Elaine, whose company made me realize sometimes the simple pleasures of life inside SCOREC and also has impressed me with
her balance of family and research at the same time. I want to mention my appreciation for support and encouragement by all of my group members Yuri, Fatih, Michael, Je Hoon, Pyong-Kwon and Juin. My heartfelt thanks to Xiangrong Li for his help with mesh adapt and also in the various aspects of mesh data base. His thorough knowledge and willingness to share helped me immensely in my research. I wish to express my thanks to Azat and Alisa who always came up with the best questions and worst bugs so that I would be more cautious next time. I also would like to thank Jens, although we interacted comparatively short time, it was very fruitful. I had lot of fun working with him on error indicators. I will always remember the support I got through my friends Hema, Dhabia, Lupita, Rohit and Binoj. Their fragrance of friendship made our stay at Troy more pleasant.

Much thanks is owed to my family friends Jim and Lee Tyler, Kris and Michael Steiner, Balvinder and Satinder, Rajnish and Arti, Vivek and Swati, Amit and Anu, Rajesh and Shikha, Vikas, Kapil and Kashish, Bhavana, Anjali, Dr. Baldev Raj & family, who have been there with me in all of my needs. They have motivated me spiritually and inspired me through out my struggles. They have reminded me that there is much more to life than my work.

To say my education and dissertation are a product solely of the classes and research would do horrible injustice to the truth. Far and away the most memorable and worthwhile experiences of my graduate education were the times spent with the eclectic personalities that comprise the Prof. Jansen, Prof. Lahey research group and SCOREC. I would like to begin by thanking Christophe, who always provided me with the ample disk space and best system support. Then I would like to thank Luo, Peng, Seol, Francisco Moraga, Nicolas, Suleyman, and Srinivas.

I cannot even comprehend the sacrifices by my dear parents and by my loving Mum-myji, and the separation they had to endure to see us where we are today. They not only nourished us with their unconditional love but also profoundly influenced who I am today. I conclude by thanking all the teachers from whom I learned and got inspired, and all the people I have known, over the years that made it possible for me to continue my educational pursuits.
ABSTRACT

Multiphase flows containing mixtures of liquids, gases, and solids abound in both nature and many technological processes. The present work is focused on developing a stabilized finite element method to solve the multi-phase compressible/incompressible flow problems in three dimensions using a level set approach. The Streamline-Upwind/Petrov-Galerkin method was used to discretize the governing flow and level set equations. The method developed enables accurate computation of flows with large density and viscosity differences and allows the fronts to self-intersect, merge, break, and change topology. An adaptive-mesh strategy was designed to study various problems with reduced computational cost. Additionally, the “Ghost Fluid Method” was generalized for the finite element framework to solve multi-fluid systems with vast physical property differences. Furthermore, a fourth order accurate, explicit time integrator has been implemented to advance in time. These strategies are novel in the field of computational fluid dynamics and are expected to have a major impact on the efficient finite element analysis of multiphase flows.

Various numerical studies were performed, namely, effect of viscosity and surface tension on single and multiple bubble dynamics, non-linear dynamics of free surface flows, single and two-phase shock tubes, converging spherical shock and Rayleigh-Taylor interfacial instabilities. Using the developed framework, an effort is made to develop a three-dimensional algorithm for the hydrodynamic simulation of single bubble sonoluminescence. As a preliminary step towards the simulation of single bubble sonoluminescence, the hydrodynamics of the collapse and rebound of a 10 micron air bubble in water is studied with direct numerical simulations. It was shown that due to the inertial effect of the liquid compressing the gas, a bubble implosion takes place. The air bubble reaches a minimum radius during the implosion and then bounces back due to the high internal gas pressure. The motion of the air/water interface during the initial stages of the collapse was found to be consistent with the Rayleigh-Plesset model. However, the three dimensional simulations show that during the final stage of energetic implosions, the bubble may become asymmetric.
CHAPTER 1

Introduction

Multi-phase flows, containing mixtures of liquids, gases, and solids, abound in both nature and many technologically processes. Rainfall, ocean spray, the combustion of sprayed fuels, boiling liquids, and cavitation bubbles are just a few of the cases that researchers would like to understand better. And such research is not just an academic exercise—these flows have great practical importance. Better understanding of multi-phase flows could improve processes ranging from heat transfer and atomization to suspensions, cavitation, and condensation processes.

Owing to their great practical importance, scientists and engineers have long constructed models to predict these multi-phase flows. But all such models must still rely on approximate, empirically-derived coefficients because they cannot resolve the fine scales where the details of the interactions between bubbles take place. While it is the overall, integral characteristics of such flows that have the greatest practical significance, they are strongly influenced by the evolution of the smallest scales in the flow, including the dynamics of bubbles. So, in many important cases, one cannot simply leave out the small scale dynamics of bubbles if one wants to understand and predict the overall characteristics of multi-phase flows.

In an effort to learn more about the dynamics of bubbles rising in a liquid and how they interact, researchers traditionally carry out experiments. But the motions of bubbles take place on short spatial scales and rapid time scales. Moreover, sometimes the experiments involving the compressible bubble dynamics are very sensitive and hard to setup and repeat. So despite advances in technology, experimentalists continue to have difficulties in getting a clear look at detailed bubble interactions, and the intricacies of their dynamics.

The search for new alternatives to understand the bubble dynamics led to the various numerical techniques. However, the development of efficient algorithms to understand the hydrodynamics of multiphase flow systems is still one of the most pressing engineering challenges facing the chemical, petrochemical, nuclear, combustion and biological in-
dustries today. Owing to the expeditious advances in numerical methods and as well as in computational performance, the tool of numerical experiments became increasingly popular and also a feasible tool in study of the complex multi fluid systems.

1.1 Sonoluminescence

Sonoluminescence (SL) is the spontaneous emission of ~ 100 picosecond pulses of broadband light from micron-size gas bubbles levitated in liquid by the application of an external ultrasonic sound field. The word sonoluminescence is made up by two words. The word ”sono” for the ultrasonic waves of sound, and ”luminescence” for the light emissions. In this interesting phenomena, the gas bubble which is acoustically levitated by a standing sound wave undergoes a complicated sequence of expansions and compressions during the each acoustic cycle. The expansion of the bubble from its original equilibrium radius of about $5\mu$ to a magnitude of one order greater occurs on a hydrodynamic time scale of pico seconds during the rarefaction half cycle of the incident pressure wave. This brings the bubble to its maximum radius, and the ensuing implosion can accelerate the bubble to supersonic interfacial velocities and compresses the gas in the bubble’s interior. The high pressure and energy densities thus generated during bubble collapse results in the emission of light. The phenomena of sonoluminescence can be classified into two categories. They are

1. single-bubble sonoluminescence (SBSL), where a single bubble is used to create the light

2. multiple-bubble sonoluminescence (MBSL), where multiple bubbles, including bubble clouds, create the emissions

Although SBSL may not represent the many practical situations in chemical/nuclear reactors, owing to its simplicity, it is that is often investigated in both physical and virtual laboratories.

1.2 Existing Numerical Methods

The experiments are limited when it comes to giving a detailed insight, as the dynamics occur at very small spatial and rapid time scales. With the advances in com-
putational methods and the growing power of super computers, researchers embarked upon investigating the fluid flows in the virtual fluid dynamic lab via numerical simulations. Direct numerical simulations (DNS) offer an excellent opportunity to understand the complicated multi-phase flows. The realistic simulations promises new insights into the key mechanisms at the small scales of multiphase flows.

There are various numerical methods that are available in literature to compute multiphase flows. They all are aimed mainly at understanding the most fundamental process, that is the single bubble dynamics. To the study of complex bubble dynamics, understanding the behavior of a single bubble rising through the liquid is in fact an essential exercise. This significant fundamental process in two-phase flows gives insight into the details of the bubble dynamics and also a good exercise to test the numerical algorithm. Hence in the literature one can find various numerical methods for the simulation of the incompressible multiphase flows. By assuming the various phases being incompressible, the governing equations becomes much simpler. The simplest of all the examples is the buoyancy driven motion of a bubble in a viscous liquid. This motion is accompanied by the deformation of the bubble. Three-dimensional studies of this phenomena becomes complex due to the effect of free boundaries, density and viscosity differences and with surface tension. The buoyancy-driven motion of a bubble in a viscous liquid is normally accompanied by the deformation of the bubble. Three-dimensional studies of this phenomena become complex due to the effect of free boundaries, significant density and viscosity differences and the effect of surface tension. Various experimental studies have been performed to address this problem. Observations by Hartunian and Sears [48], Walters and Davidson [65], Grace [20], and Bhaga and Weber [15] are important experimental studies in literature. Similarly, there exists various numerical techniques. Existing computational methods used to solve multiphase flow problems have their own advantages and disadvantages. In all multiphase numerical methods, the essential ingredients are an efficient technique to solve the phasic flow fields, and an accurate strategy to keep track of the interface. Various techniques are available for solving the governing partial differential equations (PDEs) of basic flow, namely: Finite Difference methods (FDM), Finite Volume methods (FVM), and Finite Element methods (FEM).

The first essential characteristic of FEM (the method used in the present work) is
that, the continuum field is subdivide into cells called “elements”, which form a grid. The elements can have a triangular or a quadrilateral faces. The most important thing is that the grid itself need not be structured neither the elements. Due to this unstructured form, very complex geometries can be handled with ease and the adaptive meshing for the efficient computations is feasible. This is clearly the most important advantage of the method and is not shared by the FDM, which needs a structured grid. The second essential characteristic of the FEM is that the solution of the discrete problem is assumed to have a particular form, and belongs to a functional space. Hence the solution is built by interpolating the nodal values using the function. The nodes are generally the vertices of the elements in the case of linear basis functions. Its important here to note that the representation for the function is strongly linked to the geometric representation of the domain, and this key link for instance is not strong in the FVM. The third essential characteristic of the FEM is that the method does not look for a solution of the PDE itself, but looks for a solution of some integral form of the PDE. In its most general form this integral formulation is obtained from a weighted residual formulation. By this formulation the method acquires the ability to naturally incorporate differential type boundary conditions. This property constitutes the second important advantage of FEM , which is not shared by any other method.

Over the last several decades, finite element methods have grown in popularity; particularly stabilized finite element methods are vastly applied for fluid dynamics applications. Starting with the SUPG method of Brooks and Hughes [11] through the work of Hughes et al. [29] on the Galerkin/least squares (GLS) method and up to recent work on multi-scale methods of Hughes [28] and the related work on residual-free bubbles by Russo [50] and Brezzi et al. [10], a number of stabilized FEM formulations have been proposed. A key feature of stabilized methods is that they have been proven to be stable and to attain optimal convergence rates with respect to the interpolation error (e.g, see Franca et al. [19], and Hughes et al. [29]). In the present work, a stabilized finite element method is employed for computing both flow, and interface motion.
1.3 Resolving an interface

In multiphase flow simulations, when it comes to the resolving of interface motion, two types of methods are employed, namely, interface tracking, or interface capturing methods. The interface/front tracking techniques use a deforming mesh or grid which conforms to the interface, or explicitly tracks the interface (e.g. marker particles). Examples are front tracking methods [63], boundary integral methods [6], arbitrary Eulerian-Lagrangian (ALE) methods ([26]), and deforming space-time finite element formulations([61], [62]). An advantage of front tracking methods is that they are accurate with comparatively lower mesh resolution. Also, they are efficient for rigid moving boundaries. However, the algorithms have to be modified to reconnect, or disconnect the interfaces separating the various fluids. In addition, significant re-meshing is needed to prevent the so-called marker particles (which track the interface) from coming together at points of large curvature. On the other hand, in interface capturing methods, an auxiliary function defined on the fixed domain describes the interface. The interface capturing methods are very robust with wide range of applicability, however they require higher mesh resolution. Examples are volume of fluid methods (VOF) [25], phase field methods, front capturing methods, and level set methods ([52], [57], [56]). Volume of fluid methods are based on solving the conservation law for the volume fraction and they have excellent conservation properties within each phase. However, in VOF method it is difficult to calculate the curvature of the interface from the computed volume fractions. Although the level set method does not possess the same inherent conservation properties as volume of fluid methods or front tracking techniques, the strength of this method lies in it’s ability to efficiently represent an arbitrarily complex interface very accurately, thereby allowing the computation of flows with surface tension and rapidly changing topology.

In the literature, Chen et al. [13] studied the deformation of a three-dimensional bubble using a modified VOF method with finite difference discretization. Sussman et al. [57] presented a level set approach to study the same problem, again using a finite difference approximation. Barth et al. [3] developed a finite element Petrov-Galerkin scheme for the level set equations on triangulated domains. The present work extends the aforementioned works to present a stabilized three dimensional, finite element approach using the level set method for solving both compressible and incompressible multiphase
flows.

The continuum surface force (CSF) model proposed by [7] was implemented to incorporate surface tension induced stresses. Also, a re-distancing strategy [43] was adopted to ensure the volume of the each phase is conserved during re-distancing. Although the resulting algorithm is known to predict the single-phase compressible gas dynamics quite well, for the compressible multiphase flows, it may lead to the spurious non-physical oscillations across the interface due to smeared density profiles. This can be attributed to the radical change in the equation of state across the interface. Hence, an approach similar to the ghost fluid method developed by [17] in the finite difference context, has been developed and implemented into our finite element formulation. This strategy allows us to treat compressible multiphase flows with large density differences. It should be noted that the strategy and FEM formulations presented in this work, employing both ghost fluid and level set methods, are novel in the field of computational fluid dynamics and are very suitable for the analysis of compressible 3-D multiphase flows.

1.4 Overview

This thesis report presents a numerical method for solving multiphase compressible/incompressible flows by using a level set approach in 3-D. The outline of the thesis is as follows. Chapter 2 introduces the level set method. Section 2.2 in Chapter 2 deals with the modeling of surface tension force across the interface. Chapter 3 deals with the finite element formulations for solving the Navier-Stokes equations which are discretized in space using a stabilized finite element method to obtain a nonlinear system of coupled ordinary differential equations. In this chapter, Section 3.1.2 presents the compressible formulations and Section 3.1.2.1 describes the discontinuity capturing operator implemented to efficiently perform the flow computations with sharp discontinuities or involving shocks. Section 3.2.2 presents the incompressible formulations. Chapter 4 presents the details of the modeling of the level set method and the finite element solution technique for solving both the level set and the re-distancing equations. To eliminate the spurious oscillations across the interface in compressible flows a novel approach on the lines of Ghost Fluid method has been adapted, and this is discussed in the Chapter 5. The discussion of the adaptive strategy developed is presented in Chapter 7. In Chapter 8
presents some of the results obtained from the incompressible multiphase flow simulations. Chapter 9 presents the compressible simulations. In Chapter 10, simulation of free surface flows is discussed and some of the preliminary results are presented. Chapter 11 presents the studies of Rayleigh-Taylor instabilities. Chapter 12 presents the hydrodynamic simulation of an air bubble implosion. Finally, in Chapter 13 some conclusions are drawn and the future directions are discussed.
CHAPTER 2
The Level Set Method

The level set approach represents the free surface as a zero level set of a smooth function, and simultaneously maintains a level set function, which by definition is the signed distance from the interface. Hence, instead of explicitly tracking the interface, the method implicitly “captures” the interface within a field which is interpolated with finite element basis functions like any other state variable (e.g., pressure, velocity, temperature). This enables us to represent the interface between the two phases accurately and to compute flows with large density ratios and surface tension. As mentioned in [57], conventional conservative methods suffer with excessive numerical diffusion which smears the sharpness of the front. The level set function is typically a smooth (Lipschitz continuous) function, denoted here as $\phi$, which eliminates the difficulties that conventional conservative schemes incur.

Probably the most important advantage of level set methods is that the interfaces can merge or break-up with ease. Furthermore, the level set formulation generalizes easily to three dimensions. The actual location of the interface is never computed, since the interface is embedded as a particular level set in a fixed domain.

2.1 Level set description

In our formulations, a smooth level set function, $\phi$, is used to track the interface between the gas phase and the liquid phase. The interface, $\Gamma'$, is the zero level set of $\phi$:

$$\Gamma' = \{ \bm{x} | \phi(\bm{x}, t) = 0 \}$$  \hspace{1cm} (2.1)

The level set function is considered to be positive in the liquid phase and negative in the gas phase. Hence,

$$\phi(\bm{x}, t) = \begin{cases} > 0, & \text{if } \bm{x} \in \text{the liquid}, \\ = 0, & \text{if } \bm{x} \in \Gamma', \\ < 0, & \text{if } \bm{x} \in \text{the gas}. \end{cases}$$  \hspace{1cm} (2.2)
Therefore, $\phi$ is initialized to be the signed normal distance from the interface. Since the interface moves with the fluid, the evolution of $\phi$ is governed by a transport equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0$$  \hspace{1cm} (2.3)

As will be discussed in Section 4.1, this additional advection equation for the level set scalar is solved in a manner similar to the equations of motion. The physical properties of the fluid in each phase are calculated as a function of $\phi$ as:

$$\rho(\phi) = \rho_1 H(\phi) + \rho_2 (1 - H(\phi)),$$  \hspace{1cm} (2.4)

and similarly,

$$\mu(\phi) = \mu_1 H(\phi) + \mu_2 (1 - H(\phi)),$$  \hspace{1cm} (2.5)

where $H(\phi)$ is the Heaviside function given by

$$H(\phi) = \begin{cases} 
0, & \text{if } \phi < 0, \\
\frac{1}{2}, & \text{if } \phi = 0, \\
1, & \text{if } \phi > 0.
\end{cases}$$  \hspace{1cm} (2.6)

### 2.1.1 Interface thickness

Use of the Heaviside function described above leads to poor results due to the assumed zero thickness of the interface. Instead, one can use an alternative description of the interface as proposed by [57], [63], [56]. Numerically, they substitute a smoothed Heaviside function $H_\epsilon(\phi)$ for the sharp Heaviside function $H(\phi)$. The smooth Heaviside function is defined as [56]:

$$H_\epsilon(\phi) = \begin{cases} 
0, & \text{if } \phi < -\epsilon, \\
\frac{1}{2} \left[ 1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin \left( \frac{\pi \phi}{\epsilon} \right) \right] \text{ if } |\phi| \leq \epsilon, \\
1, & \text{if } \phi > \epsilon.
\end{cases}$$  \hspace{1cm} (2.7)

where $\phi$ represents the signed normal distance to the interface. The $\frac{1}{2}$ contour of the sharp Heaviside function $H(\phi)$ creates jagged or staircase contours on any discrete mesh of spacing $\Delta x$. However, by giving the interface a thickness of $\epsilon = \alpha \Delta x$, where $\alpha > 1$, the...
sharp changes across the interface are smoothed.

2.1.2 Re-initialization or re-distancing of level sets

In the formulation described above, the interface will have a uniform thickness so long as \( \phi \) is maintained as a distance function. However, under the evolution of equation (2.3), the level sets that are adjacent to the zero level set may move with velocities different than that of the zero level set. Therefore, the \( \phi \) distance field gets distorted. Thus, one must re-initialize the level set function on regular intervals in order to rebuild/maintain the signed distance function. There are several ways to accomplish this re-distancing. The technique introduced by Sussman et al. [57] is used in this work. Its virtue is that the level set function can be re-initialized without explicitly finding the zero level set. The idea is to solve the partial differential equation

\[
\frac{\partial d}{\partial \tau^d} = S(\phi)(1 - |\nabla d|), \quad (2.8)
\]

where

\[
S(\phi) = \begin{cases} 
-1, \text{if } \phi < 0, \\
0, \text{if } \phi = 0, \\
1, \text{if } \phi > 0.
\end{cases} \quad (2.9)
\]

d(\textbf{x}, 0) = \phi(\textbf{x}, t) and \( \tau^d \) is a pseudo time. Given any initial data for \( \phi \), solving this equation to steady state provides the distance field \( \phi \) with the property \( |\nabla \phi| = 1 \), since convergence occurs when the right hand side of (2.8) is zero. Note that the sign function \( S(\phi) \) controls the flow of information. If \( \phi \) is negative, information flows one way, and if \( \phi \) is positive, information flows the other way. The net effect is to re-distance the level sets on either side of the zero level set. Furthermore, instead of the sharp sign function, one can use the smooth sign function defined as

\[
S(\phi) = 2 \left( H(\phi) - 1/2 \right) \quad (2.10)
\]

where \( H(\phi) \) is given by (2.7). The steady solutions of (2.8) are distance functions. In addition, since \( S(0) = 0 \), then \( d(\textbf{x}, \tau^d) \) has the same zero level set as \( \phi(\textbf{x}, t) \). Note, that this equation is relaxed in pseudo-time \( \tau^d \) which is not related to the physical time \( t \).
Hence (2.8) need to be solved only for $\tau^d = 0$ ... $\epsilon \times v$, because the level set function re-initialization is required only near the interface. The definition of the $v$ is obvious if (2.8) is rewritten as,

$$\frac{\partial d}{\partial \tau^d} + v \cdot \nabla d = S(\phi), \tag{2.11}$$

where,

$$v = S(\phi) \frac{\nabla d}{|\nabla d|} \tag{2.12}$$

Equation (2.11) is a non-linear hyperbolic equation with the characteristics pointing outwards from the interface in the direction of the normal. The strategy, which was adopted in the present study was to perform this re-distancing operation at the end of each time step.

### 2.2 Modeling of surface tension

For many fluid flow problems, interfacial motion induced by surface tension may play a significant role. The surface tension force is a result of the uneven molecular forces of attraction experienced by fluid molecules near the interface. Surface tension creates a microscopic, localized surface force that exerts itself in both tangential and normal directions. The continuum approach proposed by Brackbill et al. [7] was used, in order to represent the surface tension force as a body force. This model represents surface tension as a continuous three-dimensional effect across an interface, rather than as a boundary value condition at the interface. The resulting body force due to surface tension can be written as

$$f = -\frac{\kappa(\phi)\nabla H(\phi)}{W} \tag{2.13}$$

where $\kappa(\phi)$ is the curvature defined as

$$\kappa(\phi) = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \tag{2.14}$$

and $H(\phi)$ is the Heaviside function as defined in equation (2.7), and $W$ is the Weber number given by;

$$W = \frac{\rho_1 L U^2}{\sigma} \tag{2.15}$$
The surface tension term and local inter-facial curvature are easily represented in terms of the level set function. As the level set function in our formulations is a signed distance from the interface, the curvature can be accurately computed from the level set function. However, the calculation of curvature involves the second order derivatives. Since piecewise-linear basis functions are being used in the present work, these second derivatives are zero on the element interior. Hence, the piecewise-constant gradients were reconstructed to be continuous using $L_2$ projection, and then the second order derivatives were evaluated by differentiating the reconstructed gradient. This procedure is described in detail by Jansen et al. [31]. The form of the surface tension force as a body force used here is due to Chang et al. [12].

2.3 Chapter summary

In this chapter, the level set method was introduced. The thickness of interface is defined. Most importantly the chapter discusses the critical issue of re-distancing the level sets in detail so that a distance function can be maintained. At the end a continuum surface force method is discussed to model the surface tension forces across the interface. The finite element formulation for the level set method is discussed in future chapters.
CHAPTER 3
The Finite Element Formulation of the Flow

Since, the present work deals with the simulation of both compressible and incompressible multiphase flows, this chapter discuss formulations for both the flows.

3.1 Compressible flow formulations

This section specifically discusses the compressible formulations. First, the strong form is presented and then a weak form is presented along with the constitutive relations.

3.1.1 Compressible strong Form

Consider the compressible equations of motion (complete with the continuity and total energy equations) written in conservative form [32, 30]:

\[ U_{,t} + F_{i, i}^{adv} - F_{i, i}^{diff} = S \]  \hspace{1cm} (3.1)

where

\[ U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \rho \begin{bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{tot} \end{bmatrix} \]  \hspace{1cm} (3.2)

\[ F_{i}^{adv} = u_i U + p \begin{bmatrix} \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{bmatrix} \]  \hspace{1cm} (3.3)
The variables are: the velocity $u_i$, the pressure $p$, the density $\rho$, the temperature $T$ and the total specific convected energy $e_{\text{tot}}$. The constitutive laws relate the stress, $\tau_{ij}$, to the deviatoric portion of the strain, $S^d_{ij} = S_{ij} - \frac{1}{3}S_{kk}\delta_{ij}$, through a molecular viscosity, $\mu$. Similarly, the heat flux, $q''_i$, is proportional to the gradient of temperature with the proportionality constant given by the molecular conductivity. The formulations presented in this work can accommodate a general divariant fluid. A variety of equations of state can be considered, however in this study, the gas phase was assumed to be an ideal gas and the liquid phase was governed by a simple linear equation of state. Note that, although there are two fluids (with two viscosities, two conductivities and two constitutive laws), however the level set field, $\phi$, allows us to consider these fields as one global field for each varying quantity. Finally, $\mathbf{S}$ is a body force (or source) term, such as gravity, and the force due to the surface tension. The representation of the surface tension surface force as a body force applied on the interface is dealt with in Section 2.2.

For the specification of the methods that follow, it is helpful to define the quasi-linear operator (with respect to some yet to be determined variable vector, $\mathbf{Y}$) related to equation (3.1) as

$$
\mathbf{F}_i^{\text{diff}} = \left\{ \begin{array}{c}
0 \\
\tau_{1i} \\
\tau_{2i} \\
\tau_{3i} \\
\tau_{ij} u_j - q''_i
\end{array} \right\}
$$

and,

$$
\tau_{ij} = 2(\mu)(S_{ij}(\mathbf{u}) - \frac{1}{3}S_{kk}(\mathbf{u})\delta_{ij}), \quad S_{ij}(\mathbf{u}) = \frac{u_{i,j} + u_{j,i}}{2} \tag{3.5}
$$

$$
q''_i = -\kappa T_i, \quad e_{\text{tot}} = e + \frac{u_i u_i}{2}, \quad e = c_v T \tag{3.6}
$$

The variables are: the velocity $u_i$, the pressure $p$, the density $\rho$, the temperature $T$ and the total specific convected energy $e_{\text{tot}}$. The constitutive laws relate the stress, $\tau_{ij}$, to the deviatoric portion of the strain, $S^d_{ij} = S_{ij} - \frac{1}{3}S_{kk}\delta_{ij}$, through a molecular viscosity, $\mu$. Similarly, the heat flux, $q''_i$, is proportional to the gradient of temperature with the proportionality constant given by the molecular conductivity. The formulations presented in this work can accommodate a general divariant fluid. A variety of equations of state can be considered, however in this study, the gas phase was assumed to be an ideal gas and the liquid phase was governed by a simple linear equation of state. Note that, although there are two fluids (with two viscosities, two conductivities and two constitutive laws), however the level set field, $\phi$, allows us to consider these fields as one global field for each varying quantity. Finally, $\mathbf{S}$ is a body force (or source) term, such as gravity, and the force due to the surface tension. The representation of the surface tension surface force as a body force applied on the interface is dealt with in Section 2.2.

For the specification of the methods that follow, it is helpful to define the quasi-linear operator (with respect to some yet to be determined variable vector, $\mathbf{Y}$) related to equation (3.1) as

$$
\mathbf{L} \equiv A_0 \frac{\partial}{\partial t} + A_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i} (K_{ij} \frac{\partial}{\partial x_j}) \tag{3.7}
$$
which can be decomposed into time, advective, and diffusive portions,

$$\mathcal{L} = \mathcal{L}_i + \mathcal{L}_{adv} + \mathcal{L}_{diff}.$$  \hfill (3.8)

Here $A_i = F_{i,Y}$ is the $i^{th}$ Euler Jacobian matrix, $K_{ij}$ is the diffusivity matrix, defined such that $K_{ij} Y_j = F_{i,diff}$, and $A_0 = U, Y$ is the change of variables metric. For a complete description of $A_0, A_i$ and $K_{ij}$, the reader is referred to [21, 23]. Using these results, one can write (3.1) as simply, $\mathbf{LY} = \mathbf{S}$.

### 3.1.2 Weak form-finite element discretization

To proceed with the finite element discretization of equations of motion (3.1), one must define the finite element approximation spaces. Let $\Omega \subset \mathbb{R}^N$ represent the closure of the physical spatial domain (i.e., $\Omega \cup \Gamma$ where $\Gamma$ is the boundary) in $N$ dimensions; only $N = 3$ is considered here. The boundary is decomposed into portions with natural boundary conditions, $\Gamma_h$, and essential boundary conditions, $\Gamma_g$, i.e., $\Gamma = \Gamma_g \cup \Gamma_h$. In addition, $H^1(\Omega)$ represents the usual Sobolev space of functions with square-integrable values and derivatives on $\Omega$.

Next, $\Omega$ is discretized into $n_{el}$ finite elements, $\Omega^e$. With this, the trial solution space may be defined for the semi-discrete formulations as,

$$\mathcal{V}_h = \{v|v(\cdot,t) \in H^1(\Omega)^m, t \in [0,T], v|_{x \in \Omega^e} \in P_k(\Omega^e)^m, v(\cdot,t) = g \text{ on } \Gamma_g\}, \hfill (3.9)$$

and the weight function space as,

$$\mathcal{W}_h = \{w|w(\cdot,t) \in H^1(\Omega)^m, t \in [0,T], w|_{x \in \Omega^e} \in P_k(\Omega^e)^m, w(\cdot,t) = 0 \text{ on } \Gamma_g\}, \hfill (3.10)$$

where $P_k(\Omega^e)$, is the space of all polynomials defined on $\Omega^e$, complete to order $k \geq 1$, and $m$ is the number of degrees of freedom ($m = 5$).

To derive the weak form of (3.1), the entire equation is dotted with a vector of weight functions, $W \in \mathcal{W}_h$, and integrated over the spatial domain. Integration by parts is then performed to move the spatial derivatives onto the weight functions thus decreasing the continuity requirements. This process leads to an integral equation (often referred to
as the weak form): find $\mathbf{Y} \in \mathcal{V}_h$ such that

$$
0 = \int_{\Omega} (\mathbf{W} \cdot \mathbf{A}_0 \mathbf{Y}_t - \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\text{adv}} + \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\text{diff}} + \mathbf{W} \cdot \mathcal{S}) \, d\Omega
- \int_{\Gamma} \mathbf{W} \cdot (-\mathbf{F}_i^{\text{adv}} + \mathbf{F}_i^{\text{diff}}) \, n_i \, d\Gamma
+ \sum_{e=1}^{n_e} \int_{\Omega_e} \mathbf{L}^T \mathbf{W} \cdot \mathbf{\tau} (\mathbf{LY} - \mathcal{S}) \, d\Omega
$$

(3.11)

The first and second lines of (3.11) contains the Galerkin approximation (interior and boundary) and the last line contains the least-squares stabilization. Note that SUPG [11] stabilization is obtained by replacing $\mathbf{L}^T$ by $\mathbf{L}_i^{T_{\text{adv}}}$. The stabilization matrix $\mathbf{\tau}$ is an important ingredient in these methods and it has been well documented by [53, 19]. Note that we have chosen to find $\mathbf{Y}$ instead of $\mathbf{U}$. As discussed by [22], $\mathbf{U}$ is often not the best choice of solution variables, particularly when the flow is nearly incompressible. For the calculations performed herein, the SUPG stabilized method was applied with linearly interpolated pressure-primitive variables which are discussed in detail by [69].

$$
\mathbf{Y} = \begin{pmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
Y_4 \\
Y_5
\end{pmatrix} = \begin{pmatrix}
p \\
u_1 \\
u_2 \\
u_3 \\
T
\end{pmatrix}
$$

(3.12)

By inspecting (3.4)-(3.6) it is clear that all quantities appearing in (3.11) may be calculated using (3.12).

To develop a numerical method, the weight functions ($\mathbf{W}$), the solution variable ($\mathbf{Y}$), and it’s time derivative ($\mathbf{Y}_t$) are expanded in terms of basis functions (typically piecewise polynomials); all calculations described here in were performed with a linear basis function. The extension of the quadratic and cubic basis can be done as well on the validated quadratic and cubic bases implementation by [68]). The integrals in (3.11) are then evaluated using Gauss quadrature resulting in a system of non-linear ordinary
differential equations which can be written as,

\[ M\dot{\mathbf{y}} = \mathbf{N}(\mathbf{y}) \]  \hspace{1cm} (3.13)

where the under bar is added to make clear that \( \mathbf{y} \) is the vector of solution values at discrete points (spatially interpolated with the finite element basis functions) and \( \dot{\mathbf{y}} \) are the time-derivative values at the same points. Finally this system of non-linear ordinary differential equations are solved using a fourth order explicit Runge-Kutta time integrator( 6).

### 3.1.2.1 Discontinuity capturing operator

Despite the success of the SUPG, it is well known to be insufficient for flows which contain discontinuities. For example, SUPG does not preclude overshooting and undershooting about sharp layers. Therefore, [27] suggested a simple technique for improving resolution of sharp layers while maintaining the optimal rate of convergence. The method adds an additional “discontinuity capturing term” (DC) which has a form similar to the streamline term, but acts in the direction of the solution gradient rather than in the direction of the streamline. The dependence of this term on the solution gradient results in a discrete method, which is nonlinear even when the original PDE was linear. This is of little consequence to the problems that are considered here, since the flow non-linearity is more severe than the non-linearity introduced by this operator.

The discontinuity operator adds an additional term to the original variational formulation, (3.11), as documented in [53]. This term provides additional control over gradients in the discrete solution and considerably increases the robustness of the methodology. The DC operator needs to satisfy: (i) in order to control the oscillations, this operator should act in the direction of the gradient, (ii) for consistency it should be proportional to the residual, (iii) for accuracy, it should quickly vanish in smooth regions of the solution. The term derived by Hughes et al. is of the form,

\[ \sum_{e=1}^{n_{el}} \int_{\Omega_e} \nu^h \hat{\nabla}_\xi W^h \cdot [\hat{A}_0] \hat{\nabla}_\xi V^h d\Omega \]  \hspace{1cm} (3.14)

here \( \hat{\nabla}_\xi \) is the generalized local-coordinate gradient operator, and \( \nu^h \) is scalar discon-
tinuity capturing factor of dimension one over time, \( \tilde{A}_0 \) is the conversion matrix from conservative variables to the entropy variables and \( \mathbf{V} \) indicates the entropy variables [53]. The DC operator can be written more explicitly (for our choice of variables \( \mathbf{Y} \)) as,

\[
\sum_{e=1}^{n_e} \int_{\Omega_e} \nu^h g^{ij} W^h_i \cdot [\tilde{A}_0] Y_{,j} \, d\Omega
\]

where \( \mathbf{Y} \) is the pressure primitive variables described in equation (3.12) and \( g^{ij} \) is the inverse of the Jacobian of the mapping. The discontinuity factor \( \nu^h \) is defined as

\[
\nu^h = \max \left\{ 0, \left[ \frac{(\mathbf{L} \mathbf{Y} - \mathbf{S}) \cdot \tilde{A}_0^{-1} (\mathbf{L} \mathbf{Y} - \mathbf{S})}{g^{ij} Y_{,i} \cdot A_0^{DC} Y_{,j}} \right]^\frac{1}{2} - \left[ \frac{(\mathbf{L} \mathbf{Y} - \mathbf{S}) \cdot \tilde{r} (\mathbf{L} \mathbf{Y} - \mathbf{S})}{g^{ij} Y_{,i} \cdot A_0^{DC} Y_{,j}} \right] \right\}
\]

### 3.2 Incompressible flow formulations

This section presents the formulations for incompressible multiphase flows. First, the strong form is presented and then the finite element discretization is discussed.

#### 3.2.1 Incompressible fbw

When both phases of the flow can be considered incompressible, and as well as immiscible, the governing equations for the flow are the well known equations of motion represented as ([32], [30]);

\[
\rho (\dot{u}_i + u_j u_{i,j}) = -p_i + \tau_{ij,j} + f_i \quad \text{(3.17)}
\]

\[
u_{i,j} = 0 \quad \text{(3.18)}
\]

The density and the viscosity are simply convected by the fluid velocity, yielding the following state equations for density and viscosity.

\[
\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0 \quad \text{(3.19)}
\]

\[
\frac{\partial \mu}{\partial t} + \mathbf{u} \cdot \nabla \mu = 0 \quad \text{(3.20)}
\]
The variables are: the velocity $u$, the pressure $p$, the density $\rho$. Constitutive laws relate the stress tensor, $\tau_{ij}$, to the deviatoric portion of the strain, $S_{ij}^d = S_{ij} - \frac{1}{3}S_{kk}\delta_{ij}$, through a molecular viscosity, $\mu$. For an incompressible flow, from (3.18) the divergence of the flow is zero, hence the stress tensor is simply the symmetric strain rate tensor. Hence

$$\tau_{ij} = \mu (u_{i,j} + u_{j,i}) \quad (3.21)$$

Finally $f$ is a volumetric source term, such as gravity and the interfacial force resulting from the surface tension. The representation of the surface tension surface force as a body force applied on the interface is dealt with in Section 2.2 in detail. Equations (3.19) and (3.20) indicates that density and viscosity are constant along the stream lines. Further, it is assumed that the density and viscosity are constant in each phase and they vary only across the interface. Hence, their variation could be governed by a single smooth function, namely the level set function, which is discussed in detail in Section 2.1.

### 3.2.2 Weak form—Finite element discretization

To derive the finite element discretization of the weak form of the equations of motion (3.17, 3.18), discrete weight and solution function spaces must be introduced. Let $\mathcal{\bar{\Omega}} \subset \mathbb{R}^N$ represent the closure of the physical spatial domain (i.e. $\Omega \cup \Gamma$ where $\Gamma$ is the boundary) in $N$ dimensions; where only $N = 3$ is considered here. The boundary is decomposed into portions with natural boundary conditions, $\Gamma_h$, and essential boundary conditions, $\Gamma_g$, i.e., $\Gamma = \Gamma_g \cup \Gamma_h$. In addition, $H^1(\Omega)$ represents the usual Sobolev space of functions with square-integrable values and derivatives on $\Omega$.

Subsequently $\Omega$ is discretized into $n_v$ finite elements, $\mathcal{\bar{\Omega}}_e$. With this, one can define the discrete trial solution and weight spaces for the semi-discrete formulation as:

$$\mathcal{S}_h^k = \{ u | u(\cdot, t) \in H^1(\Omega)^N, t \in [0, T], u|_{x \in \mathcal{\bar{\Omega}}_e} \in P_k(\mathcal{\bar{\Omega}}_e)^N, u(\cdot, t) = g \text{ on } \Gamma_g \}, \quad (3.22)$$

$$\mathcal{W}_h^k = \{ w | w(\cdot, t) \in H^1(\Omega)^N, t \in [0, T], w|_{x \in \mathcal{\bar{\Omega}}_e} \in P_k(\mathcal{\bar{\Omega}}_e)^N, w(\cdot, t) = 0 \text{ on } \Gamma_g \}, \quad (3.23)$$

$$\mathcal{P}_h^k = \{ \partial| \partial(\cdot, t) \in H^1(\Omega), t \in [0, T], \partial|_{x \in \mathcal{\bar{\Omega}}_e} \in P_k(\mathcal{\bar{\Omega}}_e) \} \quad (3.24)$$
where $P_k(\tilde{\Omega}_e)$ is the space of all polynomials defined on $\Omega^e$, complete to order $k \geq 1$.

Let us emphasize that the local approximation space, $P_k(\tilde{\Omega}_e)$, is same for both the velocity and pressure variables. This is possible due to the stabilized nature of the formulation to be introduced below. These spaces represent discrete subspaces of the spaces in which the weak form is defined.

The stabilized formulation used in the present work is based on the formulation described by Taylor et al. [59]. Given the spaces defined above, the semi-discrete Galerkin finite element formulation applied to the weak form of (3.17) is:

Find $u \in \mathcal{S}_h^k$ and $\vartheta \in \mathcal{P}_h^k$ such that

$$B_G(w_i, q; u_i, \vartheta) = 0 \quad \text{(3.25)}$$

$$B_G(w_i, q; u_i, \vartheta) = \int_{\Omega} \{ w_i (\dot{u}_i + u_j u_{i,j} - f_i) + w_{i,j} (-\partial \delta_{ij} + \tau_{ij}) - q_i u_i \} \, dx$$

$$+ \int_{\Gamma_h} \{ w_i (\partial \delta_{in} - \tau_{in}) + q u_n \} \, ds \quad \text{(3.26)}$$

for all $w \in \mathcal{W}_h^k$ and $q \in \mathcal{P}_h^k$. The boundary integral term arises from the integration by parts and is only carried out over the portion of the domain without essential boundary conditions.

Since the Galerkin method is unstable for the equal-order interpolations given above, additional stabilization terms are introduced: Find $u \in \mathcal{S}_h^k$ and $\vartheta \in \mathcal{P}_h^k$ such that

$$B(w_i, q; u_i, \vartheta) = 0 \quad \text{(3.27)}$$

$$B(w_i, q; u_i, \vartheta) = B_G(w_i, q; u_i, \vartheta)$$

$$+ \sum_{e=1}^{n_e} \int_{\tilde{\Omega}_e} \{ \tau_M(u_j u_{i,j} + q_i) \mathcal{L}_i + \tau_C w_{i,j} u_{i,k} \} \, dx$$

$$+ \sum_{e=1}^{n_e} \int_{\tilde{\Omega}_e} \{ w_i \Delta u_j u_{i,j} + \tau \Delta u_j w_{i,j} u_{i,k} \} \, dx \quad \text{(3.28)}$$

for all $w \in \mathcal{W}_h^k$ and $q \in \mathcal{P}_h^k$. For simplicity, $\mathcal{L}_i$ is used to represent the residual of the $i^{th}$
momentum equation,
\[ \mathcal{L}_i = \ddot{u}_i + u_j u_{i,j} + \partial_i - \tau_{ij,j} - f_i \] (3.29)

The second line in the stabilized formulation, (3.28), represents the typical stabilization added to the Galerkin formulation for the incompressible set of equations (e.g. Franca and Frey [19]). The description of the individual terms and the stabilization parameters for continuity and momentum are discussed in detail by Whiting and Jansen [68]. The same reference also provides the remaining flow discretization details.

### 3.3 Chapter summary

This chapter introduced the stabilized finite element formulations for both compressible and incompressible multiphase flows. The implementation of a discontinuity capturing technique to control the large gradients in the flow is described. The next chapter presents the finite element formulations for level set method.
CHAPTER 4
Finite Element Formulations for Level Sets

4.1 Discretization

Equations (2.3) and (2.8) can be represented by a single scalar advection equation of the form

\[
\frac{\partial \theta}{\partial t} + \mathbf{a} \cdot \nabla \theta = S
\]

(4.1)

In case of the level set equation, \( \theta = \phi \), \( \mathbf{a} = \mathbf{u} \) and the forcing function \( S \) is zero. On the other hand, in the case of re-distancing equation, \( \theta = d \), \( \mathbf{a} = \mathbf{v} \), and \( S \) is given by (2.10). This section presents the general finite element formulation for the scalar advection equation, (4.1). The solution strategy for solving this equation is similar to that used for the equations of motion, as discussed in Sections 3.2.2 & 3.1.2.

Again, the spatial discretization is performed using the finite element method, and the finite element approximation spaces, namely the solution and the weight function space, are as defined in Chapter 3. In keeping with the approach of the finite element method described above, we write (2.3) in the residual form.

\[
\frac{\partial \theta}{\partial t} + \mathbf{a} \cdot \nabla \theta - S = 0
\]

(4.2)

To proceed with the weak formulations, let us define the scalar counter parts of the vector solution and weight spaces defined in Chapter 3.

\[
\Theta_h = \left\{ \mathbf{\nu} | \mathbf{\nu}(\cdot, t) \in H^1(\Omega)^N, t \in [0, T], \mathbf{\nu}|_{x \in \tilde{\Omega}_e} \in P_k(\tilde{\Omega}_e)^N, \mathbf{\nu}(\cdot, t) = \mathbf{g} \text{ on } \Gamma_g \right\},
\]

(4.3)

\[
\Psi_h = \left\{ \psi | \psi(\cdot, t) \in H^1(\Omega)^N, t \in [0, T], \psi|_{x \in \tilde{\Omega}_e} \in P_k(\tilde{\Omega}_e)^N, \psi(\cdot, t) = 0 \text{ on } \Gamma_g \right\}
\]

(4.4)

To derive the weak form of residual form, (4.2) is multiplied by a smooth weighting function \( \psi \) belonging to a space of functions \( \psi \in \Psi_h \). The product is then integrated over a spatial domain. As the equation is solved in the convective form, we do not integrate by parts thus there are no boundary integrals. The resulting weak form is: find \( \theta \in \Theta_h \) such
that,
\[
\int_{\Omega} (\psi \theta_i + \psi a_i \theta_{,i} + \psi S) \, d\Omega + \sum_{e=1}^{n_e} \int_{\Omega_e} \{ \tilde{L}^T \psi \tau (\theta_{,i} + a_i \theta_{,i} - S) \} \, dx = 0 \tag{4.5}
\]
for all \( \psi \in \Psi_h \). Here \( \tau \) is the stabilization parameter defined as
\[
\tau = \frac{C}{\sqrt{c_1/\Delta t^2 + c_2 u_i g_{ij} u_j}} \tag{4.6}
\]
where \( C, \ c_1, \) and \( c_2 \) are defined based on the one-dimensional, linear advection-diffusion equation using a linear finite element basis and \( g_{ij} = \xi_{k,i} \xi_{k,j} \) is the covariant metric tensor related to the mapping from global to element coordinates. Note that by replacing \( \tilde{L}^T \) by \( \tilde{L}^{adv} \) (for the scalar equation \( \tilde{L}^{adv} = a_i \frac{\partial}{\partial x_i} \)), we obtain the SUPG (Streamline Upwind Petrov Galerkin) stabilization[11]. Next, the weight functions \( (\psi) \), the solution variable \( (\theta) \), and it’s time derivative \( (\theta_{,i}) \) are expanded in terms of linear basis functions. The integrals are then evaluated using Gauss quadrature resulting in a system of ordinary differential equations which can be written as
\[
M \dot{\theta} = N(\theta) \tag{4.7}
\]
In case of compressible formulations, the system of ODE’s (linear, when \( \theta = \phi \), and non-linear when \( \theta = d \)) is converted to an algebraic system by introducing an explicit time integrator (4th order Runge-Kutta for \( \theta = \phi \), and forward Euler for \( \theta = d \)). Then, the two coupled systems (flow and scalar) are solved successively one after the other with in each time step. The algorithm can be described as follows:
Flow-Level set Solution (Compressible):
* The discrete equations
  - Flow equations (3.11)
  - Scalar advection equation (4.5)
  - Scalar advection equation with forcing function (4.5)
* Solved consecutively within a time step
  loop over the number of time steps
solve flow equations

advance the level sets

relax the distance field

end of the loop over time steps

In case of incompressible formulations, the system of ODE’s (linear, when $\theta = \phi$, and non-linear when $\theta = d$) is converted to an algebraic system by introducing a implicit time integrator (generalized alpha for $\theta = \phi$, and backward Euler for $\theta = d$). Then, the two coupled systems (flow and scalar) are iterated in a staggered manner until convergence is achieved. The algorithm can be described as follows:

**Staggered Flow-Level set Solution (Incompressible):**

- Three discrete equations
  - Flow equations (3.28)
  - Scalar advection equation (4.5)
  - Scalar advection equation with forcing function (4.5)
- Solved consecutively within nonlinear iteration loop

loop over the number of time steps

  loop over the nonlinear iteration loop
  
  solve flow equations
  
  advance the level sets
  
  end of the loop over nonlinear iterations

  relax the distance field

end of the loop over time steps

4.2 Redistancing - Volume constraint

During the above described staggered iteration of solving flow equations and level set equation, the interface is convected with the local flow speed, which, as mentioned before distorts the distance function. As discussed earlier, this distance function is restored by the solution of (2.11) via a finite element discretization as discussed in Section 4.1. During this re-distancing step, additional care is taken to restrict the interface
from moving. To constrain the interface, we implemented the strategy proposed by Sussman et al. [57] for the finite difference method. The principle behind the constraint calculations is to enforce the volume occupied by each phase in an element to remain constant, when the re-distance step is applied. The volume in each element can be defined as

$$V^k = \int_{\Omega^e} H(d^k) d\Omega^e$$  \hfill (4.8)

Where $H$ is the Heaviside function as given by (2.6), and $d^k$ is the distance field at the $k^{th}$ iteration of redistancing in pseudo time, that is at $\tau^k$. Since, we want to impose the constraint that the volume should not change, we should have $V^k = V^0$. In other words we can write

$$V^k - V^0 \approx (\tau^k - \tau^0) \int_{\Omega^e} \frac{dH'(d^0)}{d\tau} d\Omega^e \hfill (4.9)$$

where,

$$H'_e(d) = \begin{cases} 
0, \text{if } |d| > \epsilon, \\
\frac{1}{2} \left[ \frac{1}{\epsilon} + \frac{1}{\epsilon} \cos \left( \frac{\pi d}{\epsilon} \right) \right], \text{if } |d| \leq \epsilon.
\end{cases}$$  \hfill (4.10)

Hence, to minimize the volume variation, the current values of the level set function, denoted as $\tilde{d}^k$, are projected onto new values, denoted by $d^k$, which satisfy:

$$\int_{\Omega^e} H'_e(d^0)(d^k - d^0) d\Omega^e = 0. \hfill (4.11)$$

It is assumed that the new distance field $d^k$ has the form,

$$d^k = \tilde{d}^k + \lambda_{\Omega^e} (\tau^k - \tau^0) H'_e(d^0) \hfill (4.12)$$

where $\lambda_{\Omega^e}$ is assumed to be constant in $\Omega^e$, and is given by,

$$\lambda_{\Omega^e} = \frac{-\int_{\Omega^e} H'_e(d^0) \left( \frac{\delta^k - d^0}{\tau^k - \tau^0} \right) d\Omega^e}{\int_{\Omega^e} (H'_e(d^0))^2 d\Omega^e}. \hfill (4.13)$$

In the current algorithm the integrals for estimating $\lambda$ in (4.13) are evaluated at the el-
element level and are projected onto the global nodes by $L_2$ projection. The details of this projection is described in [31]. Then (4.12) is solved to obtain the constrained re-distanced level set function. This step is applied after each re-distancing iteration of the level set field.

4.3 Chapter Summary

The finite element formulation for the level set method is discussed. The pseudo algorithms to solve compressible and incompressible multiphase flows are presented. This chapter also discuss the volume constraint technique to ensure the volume of each is phase conserved during the transient simulations. The next chapter presents the novel ghost fluid method.
CHAPTER 5
Ghost Fluid Method

Although the Eulerian scheme works well for most compressible flows, they have been shown to admit spurious non-physical oscillations near the material interfaces. [17] proposed a new numerical method for treating interfaces in Eulerian schemes that maintains a Heaviside profile of the density with no numerical smearing. They also used the level set function to track the interface in their work. In addition, they used the ghost cells (actually ghost nodes in their finite difference framework) to prevent the smearing of the density across the interface. The motivation for their method stems from the fact that the non-physical oscillations encountered by Eulerian schemes across the material interface are due to the radical change in the equation of state across the material surface. On the other hand, Lagrangian schemes don’t smear the density profile, and it is clear which equation of state is valid at each location. But these Lagrangian schemes have their own problems when subjected to large deformations. A good summary of both Eulerian and Lagrangian schemes is given by [5]. The original method proposed by [17] tracks the interface with the level set function which gives the exact sub cell interface location. At this interface, they solved an approximate Riemann problem similar to the methods in [18] and [14].

5.1 Description

The approach taken in this work is based on the method proposed by [17]. The ghost fluid method previously developed for finite difference schemes was appropriately modified for our FEM formulations. In our method we also use the level set function to keep track of the interface. The zero level set represents the location of the interface as discussed in previous section, positive level sets representing the heavier fluid and the negative values representing the lighter fluid. Each fluid satisfies a different equation of state on each side of the interface. The clear demarcation can be made from the exact representation of the interface location by zeroth level set. Next, we define a ghost element in addition to the existing element for every element which is intersected by the interface.
The ghost elements which contain the interface have the mass, momentum and energy of both the fluids. This is done by evaluating the integrals in (3.11) accordingly.

Figure 5.1 shows a typical two dimensional triangular element which is intersected by the interface. For example, assume that the nodes 1, 2, are in the liquid phase, whereas node 3 lies inside the gas phase. Hence, for this element, we define a ghost element containing a ghost fluid. The integrals over this element are evaluated first assuming that the whole element is occupied by the liquid phase and then by assuming it is occupied by the gas phase. While building the nodal contributions to both the right hand side and the left hand side, a choice is made between the liquid or gas integral depending on whether the node lies in liquid or gas phase. Hence, in the above example while evaluating the local residual at node 1, we choose the liquid integral, on the other hand, the local residual at node 3 is evaluated based on the integral evaluation for gas phase.

The method can be further explained by the following simple 1-D example. Con-
Consider the simple 1-D domain shown in Figure 5.2. The domain is divided into three elements, and the second element is intersected by the interface. Let us assume that the region to the left of the interface contains the liquid phase and the region to the right contains the gas phase. The figure shows both the local and the global node numbering. The finite element discretization leads to an ordinary differential equation,

\[ \sum_{B=1}^{nel_{npt}} W_B G_B = 0 \]  

(5.1)

This implies that, if the weight functions are chosen to be arbitrary, then \( G_B = 0 \), where \( G_B \) is obtained by assembling the local \( G^e_b \). Note the 'B' indicates the global node numbering, the subscript 'b' indicates the local node number. The super script 'e' denotes the element number. For the above 1-D example, the global assembly operation will be defined as follows,

\[ G_1 = G^1_1 \]  

(5.2)

\[ G_2 = G^1_2 + G^2_1 \]  

(5.3)

\[ G_3 = G^3_2 + G^3_1 \]  

(5.4)

The element level residual \( G^e_b \) can be defined as,

\[ G^e_b = \int_{\Omega^e} g^e_b(\tilde{\xi}) d\Omega^e = \sum_{k=1}^{nqpt} g^e_b(\tilde{\xi}_k) \cdot W_k \]  

(5.5)

where,

\[ g^e_b(\tilde{\xi}_k) = \sum_{b=1}^{nelt} g^e(\tilde{\xi}_k) \cdot N_b (or \ N_{b,i}) \]  

(5.6)

In the above equations, 'nqpt' denotes the number of Gaussian quadrature points and the 'nelt' indicates the number of element nodes. The ghost fluid method calculates \( g^e(\tilde{\xi}_k) \) twice for the elements that contain the interface. Once assuming the whole element is occupied by the liquid phase and once by assuming this occupied by the gas. While evaluating \( g^e_b(\tilde{\xi}_k) \), depending upon the node b whether it contains the liquid phase or gas phase, \( g^e(\tilde{\xi}_k) \) is selected. In essence the method assumes a ghost fluid in all the elements.
that contain interface. This ghost fluid behaves as it is a pure liquid for the nodes that are in the liquid phase and behaves as pure gas when seen from the nodes that are in gas phase. In this way, we have continuous pressure and velocity field while avoiding the numerical dissipation due to different phasic equations of state. The resulting numerical method allows us to keep the density profile from smearing out, while still keeping the robustness of the numerical method.

5.2 Two-phase shock tube problem

To demonstrate the novel ghost fluid method, a two-phase shock tube problem is solved using the developed algorithm. This test case is similar to the cases investigated by [17, 66]. In this problem, we consider a one dimensional shock tube of length 10m in x-direction. The interface is located at 5m. The initial conditions for pressure and velocity are shown in Figures 5.3 and 5.4. The location of the interface is marked with a '*' in the plots. To the left of the interface is gas, and to the right is liquid. The initial conditions on the immediate left and right side of the interface are: \( \rho_l = 1 \text{ kg/m}^3 \), \( \rho_r = \)
Figure 5.4: Initial velocity distribution

Figure 5.5: Pressure distribution at $t = 0.003$
1000 $kg/m^3$, $p_l = 1.0 \times 10^5 \ Pa$, $p_r = 1.0 \times 10^5 \ Pa$, and $u_l = u_r = 0 \ m/sec$. In addition, both liquid and gas phases have a shock located at 0.94 m, and at 9.6 m, respectively. The shock in the gas phase is moving to the right and the conditions behind the shock are: $\rho = 8.27 \ kg/m^3$, $p = 1.0 \times 10^7 \ Pa$, and $u = 2949.97 \ m/sec$. On the other hand, the shock in the liquid phase is moving to the left, and the conditions behind the shock are: $\rho = 1004.13 \ kg/m^3$, $p = 1.0 \times 10^7 \ Pa$, and $u = -6.38 \ m/sec$.

With time, the two shocks move towards the interface, however the fast traveling shock in the gas phase strikes the interface first and sends a reflecting wave back into the gas, and transmits a shock into the liquid phase. The newly generated shock merges with the existing shock in liquid phase. The solution at $t = 0.003 \ sec$ obtained using our numerical method is shown in Figures 5.5 and 5.6. At this time, the interface has moved to 5.031m, and there is a shock wave moving to the left in the gas phase and a shock wave moving to the right in the liquid phase. The computed results are in good agreement with the results presented in [17] and [66].
5.3 Chapter summary

In this chapter the implementation of ghost fluid strategy for the finite element framework is discussed. A 1-D example is given to explain the method more clearly. A two phase shock tube problem is solved using this strategy. The results obtained are found to be in good agreement with that of in literature. The next chapter discuss the adaptive mesh refinement strategy.
CHAPTER 6
Explicit Solver

In order to develop efficient, highly accurate approximation algorithms, designing of higher order methods is a must. The higher order methods considered in the present work are higher order one step Runge-Kutta (RK) methods. Runge-Kutta methods are similar in motivation to Taylor series methods, but do not require the computation of higher order derivatives.

These methods have a number of virtues. To proceed to time $t_{k+1}$, they require no history of the solution prior to $t_k$, which makes them self starting at the beginning of integration, and also makes it easy to change step size during the integration. Another principal attribute of the method is it depends purely on the known data making the method faster compared to the implicit integrators which require the solving coupled system of equations.

6.1 General formulation of Runge-Kutta methods

In general, an s-stage Runge-Kutta method for the ODE system

$$y' = f(t, y)$$

(6.1)

can be written in the form

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f(t_{n-1} + c_j h, Y_j), \quad 1 \leq i \leq s$$

(6.2)

$$y_n = y_{n-1} + h \sum_{i=1}^{s} b_i f(t_{n-1} + c_i h, Y_i).$$

(6.3)

The $Y_i$’s are intermediate approximations to the solution at times $t_{n-1} + c_i h$, which may be correct to a lower order accuracy than the solution $y_n$ at the end of the step. Note that $Y_i$ are local to the step from $t_{n-1}$ to $t_n$, and the only approximation that the next “sees” is $y_n$. The coefficients of the method are chosen in part so that error terms cancel
in such a way that $y_n$ is more accurate. The method can be represented conveniently in a short hand notation

$$
\begin{align*}
  c_1 & \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1s} \\
  c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\
  \cdot & \cdot & \cdot & \ldots & \cdot \\
  \cdot & \cdot & \ldots & \ldots & \cdot \\
  c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\
  b_1 & b_2 & \ldots & b_s
\end{bmatrix} \\
\end{align*}
$$

Additionally, $c_i$’s are chosen such that

$$
c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, \ldots, s. \quad (6.4)
$$

The Runge-Kutta method is explicit iff $a_{ij} = 0$ for $j \geq i$, because then each $Y_i$ in (6.2) is given in known quantities. Hence, the Forward Euler is given by

$$
\begin{align*}
  0 & \begin{bmatrix} 0 & 0 & 0 & 0 \\
  \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
  \frac{1}{2} & 0 & \frac{1}{2} & 0 \\
  1 & 0 & 0 & 1 \\
  \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{bmatrix} \\
\end{align*}
$$
6.2 Implementation of fourth order explicit solver

The local residual at a time step can be written from (3.11) as

\[ G^e_b = 0 = \int_{\Omega^e} \left( N_b A_0 Y_{\cdot,t} - \mathcal{S} + \mathcal{L}^T N_b \tau (\mathcal{L} Y - \mathcal{S}) - N_{b,i} F_i \right) d\Omega \]

\[ - \int_{\Gamma} N_b F_i n_i d\Gamma \]  

(6.5)

Grouping terms such that unknown terms onto the left and the known terms onto right yields

\[ \int_{\Omega^e} N_b A_0 Y_{\cdot,t} d\Omega = \int_{\Omega^e} \left( N_b \mathcal{S} + N_{b,i} F_i - \mathcal{L}^T N_b \tau (\mathcal{L} Y - \mathcal{S}) \right) d\Omega \]

\[ - \int_{\Gamma} N_b F_i n_i d\Gamma \]  

(6.6)

We can write the above as

\[ \bar{M}^e_{ba} Y_{\cdot,t} = \bar{F} \]  

(6.7)

where

\[ \bar{M}^e_{ba} = \int_{\Omega^e} N_b A_0 N_a d\Omega \]  

(6.8)

However, one can readily see that \( \bar{M}^e_{ba} \) is not diagonal due to \( A_0 \) and the product \( N_b N_a \). If we let \( A_0 \) be constant over element, we can take \( A_0 \) out of the integral, which gives rise to a consistent matrix. Further, to extract the diagonal matrix from the consistent matrix, a special lumping technique proposed by Hughes et al. is used, yielding

\[ \bar{M} \bar{Y}_{\cdot,t} = \bar{A}_0^{-1} \bar{F} \]  

(6.9)

Runge-Kutta method uses the successive approximations of the derivative \( \bar{Y}_{\cdot,t} \) over a step to improve accuracy. The implemented fourth order Runge-Kutta method has a
form given below.

\[ f(\ddot{Y}) = M^{-1}A_\varepsilon^{-1}F \]
\[ k_1 = (\Delta t)f(\ddot{Y}) \]
\[ Y^{n+1} = Y^n + \frac{k_1}{6} \]
\[ k_2 = (\Delta t)f(\ddot{Y}) + \frac{k_1}{2} \]
\[ \dot{Y}^{n+1} = \dot{Y}^n + \frac{k_2}{3} \]
\[ k_3 = (\Delta t)f(\ddot{Y}) + \frac{k_2}{2} \]
\[ \dot{Y}^{n+1} = \dot{Y}^n + \frac{k_3}{3} \]
\[ k_4 = (\Delta t)f(\ddot{Y}) + k_3 \]
\[ \dot{Y}^{n+1} = \dot{Y}^n + \frac{k_4}{6} \]

(6.10)

6.3 Chapter summary

In this chapter an explicit time integrator is presented. The implementation of a fourth order Runge-Kutta solver is discussed. The next chapters presents the numerical studies of various problems.
CHAPTER 7
Adaptive Mesh Refinement

7.1 Adaptive strategy

Adaptive strategies are an important tool in efficient finite element computations. Adaptivity affords the opportunity to obtain numerical solutions with a controlled accuracy with minimum degrees of freedom. In addition adaptive refinement procedures have become important for increasing the reliability and reducing the cost of numerical computations in many engineering problems. The quest for more accurate numerical simulations of the Navier Stokes equations has motivated the study of higher order piecewise-polynomial basis functions, also known as $p$-version finite element methods, as a mean to attain accuracy in the most cost effective manner. For the problems with discontinuities, to achieve cost effective simulations, however some combination of ‘$h$’ and ‘$p$’ refinement is necessary. In the literature ([1], [40]) it is stated that the smooth regions of the flow benefit from $p$-refinement, while regions with discontinuities or sharp layers are better resolved using $h$-refinement. The current research deals with the problems that involve multiple shocks with varying strength, regions with discontinuous solutions, and rapidly deforming interfaces. For such kind of problems the error in discretization may results into a poor prediction of the actual physics. Hence it is important to maintain a high resolution of the mesh in the regions that have large solution gradients. However, despite the increase in the computational resources, uniformly refined mesh for large scale three dimensional problems will lead to an in-efficient and may be impractical computations. Hence, in the present work, an adaptive strategy is implemented to refine and coarsen the mesh. The essential ingredients of an adaptive procedure are

1. A tool for assessing the error of the solution computed with a given mesh

2. An algorithm to define a new spatial discretization via $h$ and/or $p$-refinement
7.1.1 Error estimation: A brief review

It is evident from the large number of publications available on error estimation and adaptivity, the research on this part of computational mechanics is still very active. However, most of the well-known literature on error estimates deals with self-adjoint problems [1]. Fluid mechanics computations, which involve non self-adjoint operators, are more difficult and concepts such as the energy norm are not always suitable for measuring the error. For this reason most of the studies in the fluid dynamics rely on local indicators to refine the mesh without specifying a total error. For instance, with this approach, the mesh can be refined considering the curvature of any representative variable or combination of variables [44].

However, in recent works, it has been shown that the gradient-based refinement is a better choice than that based on curvature in many situations of fluid mechanics [72]. This appears to be especially true for problems with strong discontinuities such as shocks, which is the current scope of the research. Handling ‘one-dimensional’ (functions vary in only one direction) localized features such as shocks and boundary layers in fluid mechanics or strain localization in geomechanics through adaptivity has become an active area of research after the work of Peraire et al. [44], it has been shown that gradient based refinement can give better refinement characteristics and is cheaper than that of curvature based refinement. Also past experience shows that the mesh converges faster with fewer cycles of refinement than those of curvature.

7.1.2 Refinement strategy

Any adaptive procedure needs a powerful refinement strategy to properly refine the mesh [40]. In the literature, two main directions are followed:

1. that of $p$-refinement in which the order of the interpolation is varied

2. $h$-refinement in which the element sizes are varied. Here some times three possibilities are distinguished:

   (a) $r$-refinement in which the structure and the connectivity of the mesh remain unaltered and so does the number of nodes
A brief comparison of the $r$-, $h$-, $p$-adaptivity is given in the following table.

In the present work, the mesh adapt tools of SCOREC ([4]) were used which are essentially follow the strategy 2b. Basic principle of mesh enrichment is that as the solution proceeds towards the steady state, the nodes are added locally based on the error indicator. The portion to be refined is identified locally and the nodes are added in the region chosen. The process of enrichment is continued until the solution reaches the required accuracy.

### 7.1.3 Error indicators implemented

Two different approaches can be used for assessing the error: error estimators or error indicators. Error estimators approximate a measure of the actual error in a given norm. Error indicators, on the other hand, are based on the heuristic considerations. Hence an error indicator only gives relative information about the error, but are typically easy to compute and implement in the analysis code. In fluid mechanics computations, which
involve non self-adjoint operators, little progress has been made on error estimators. For this reason most of the studies in the fluid dynamics rely on local error indicators to refine the mesh without specifying a total error. For the present work, error indicators are utilized for the adaptive procedure.

There are many error indicators available in the literature. Most of them are similar and based on the second gradient of a key variable. The refinement indicator based on the residual of the equation was used by Jiang and Grey [33]. Residual based estimates and $h-p$ adaptivity is for fluid mechanics problems are discussed in detail by Oden and co-workers [42]. Few studies in the literature use simple error norms such as energy norm. In the present work a total of 10 error indicators are generated at all the nodes in the computational domain. The first three error indicators are based on the $L_2$ norm of the residual of the three components of the momentum equation. The next three are $L_2$ norm of the normal component of the shear stress. The last four are based on the difference in the velocity and pressure at the current time step to the average of the of the solutions obtained from the two previous time steps. Although there is no general theory to guide the use of the last four error indicators, the preliminary results seems promising. The ten error indicators used in the present work:

\[
\begin{align*}
    e_i &= \int_{\Omega^e} N_\alpha (\mathbf{Cu}_i - f_i)^2 d\Omega & i = 1, 2, 3 \\
    e_i &= \int_{\Omega^e} N_\alpha (\tau_{ij})^2 d\Omega & i = 4, 5, 6 \\
    e_i &= \sum_{n=1}^{n_{step}} (u_i - \bar{u}_i)^2 & i = 7, 8, 9 \\
    e_i &= \sum_{n=1}^{n_{step}} (p - \bar{p})^2 & i = 10
\end{align*}
\] (7.1) (7.2) (7.3) (7.4)

Where $\bar{u}_i$ and $\bar{p}$ are the average velocity and pressure interpolated between the solution values at $n$th and $n + 1$ time steps. When computing two-phase flows an additional error indicator must be considered. The above indicators are only weakly linked to the quality of the representation of the the interface. Though the level set field is smooth, it does have a discontinuous slope at the interface. The second derivative of the scalar field is therefore a viable choice for indicating the location in need of refinement/high-resolution. This can
be readily achieved with the already available information in the algorithm. Our aim is not only to refine, but refine it smooth enough around the interface. For this purpose, the level set heaviside function property to smooth the refinement at and near the interface can be used. Results from this strategy will be given in Section 12.3.

### 7.1.4 Mesh adaptation

In this work the mesh tools developed at SCOREC ([4]) are extensively used. The mesh enrichment is done such that the individual elements are subdivided without altering their original location. As a first step the model and the original mesh are loaded into the adaptive code. Then the parameters to control the adaptive procedure given through an input file and the values of the error indicators at each node are read in. Based on this information the quantitative parameters for re-meshing are chosen. Then the edges are marked based on the error measure. The marking is allowed to propagate for a specified (typically one or two) number of layers. This serves two purposes. First it decreases the frequency that adaptivity must be carried out. Second, it allows the mesh to change size in a smoother manner which is known to reduce error. It is ensured that the original matched model faces are marked consistently. Once the edges are marked, “mesh adapt”, a SCOREC tool ([4]) is used to refine the mesh, and once the refinement is done, the mesh is smoothed.

#### 7.1.4.1 Edge marking

The decision as to whether a mesh edge is to be marked or not is made based on both error value and model classification information. Only those mesh entities whose errors are greater than a threshold value and are classified on the closure of a prescribed entities are marked. Three types of marking strategies are adapted.

1. mark a mesh vertex, if it’s error is greater than the threshold

2. Mark a mesh edge if error at it’s either of the bounding vertices is greater than than the threshold value.

3. Mark a mesh face if error at it’s any of the bounding vertices is greater than the threshold value.
7.1.4.2 Mesh quality

Maintaining the mesh quality is one of the vital factors for a successful adaptive procedure. Among many other factors to preserve the quality of the mesh, the following are few of the general conventions followed:

1. Mark the third edge if two of the edges of a face are marked

2. Always mark the longest edge if any of a mesh face is marked

3. If the original model has matched faces, always keep the marked edges on those faces consistent.

7.1.4.3 Solution transfer

Once the modified mesh is obtained the task is to interpolate the solution from the original mesh to this modified mesh. It is known fact that during the refinement only extra vertices are created or during coarsening the original one are deleted, keeping the original positions. Hence a simple interpolation between the vertices can yield a suitable initial condition for the fine mesh. This is done using an algorithm which attaches the solution obtained on the original mesh to each vertex before its being refined/coarsened. During the adaptive procedure, whenever new vertices are created during the edge based refinement, it looks for the adjacent vertices of the newly created vertex and average the solution at these vertices and attach it as a solution on this new vertex. If the mesh is coarsened in any part of the domain, information at those vertices is simply deleted. In this way at least element level conservation is preserved. In the future one may look into the possibilities of maintaining the global conservation over the entire domain.

7.1.5 Implementation of adaptive driver

The three main ingredients of the adaptive analysis code are:
the pre-processor for the flow solver – “NSpre”
the fluid flow solver –” PHASTA C/IC”
the adaptive meshing tool – “Mesh Adapt”

These three are to be integrated in such a way that the user interaction can be kept at minimum level. For this purpose an adaptive driver routine is developed, which drives the
analysis code based on the generated information from the solver. The idea of the driver is explained in the form of a flow chart given in Figure 7.1.

![Figure 7.1: Adaptive Driver](image)

The implementation of the common driver routine is done in a very general way keeping the future works in mind. Although at present only the steady state problems are solved, it can be easily modified to solve the unsteady problems too, once the Mesh adapt is modified to incorporate the solution interpolation along with the remeshing. The information is passed through pointers by addressing the memory only, rather than reading and writing files. For this purpose modifications are made to “Mesh Adapt”, “NSpre”, and to the flow solver “Phasta”. The details are given in the following subsections.

7.1.5.1 **Modification to “NSpre”**

The pre-processor creates the finite element data for the numerical analysis. The pre-processor sets the boundary conditions, connectivity and communication data structures during the pre-processing stage of the analysis. The preprocessor “NSpre” has to be
modified slightly, as it is no longer require to delete the loaded mesh and model. Once
the mesh and model are loaded, they can be used during the mesh modification by “Mesh
Adapt”. As the pre-processor “NSpre” needs a lot of libraries, to generate the mesh
database and the input files for the solver, it is in fact converted as the main driver routine,
and the flow solver and the “Mesh Adapt” are made static libraries and are passed through
a module named ‘adapt.h’. The module is included in the appendix.

7.1.5.2 Modification to flow solver “Phasta”

The main solver is changed to calculate the error indicators and to check for the
satisfaction of a tolerance (and/or after certain time step in case of unsteady problems), if
the mesh has to be refined, get the current time step and the solution at this time step and
the error at all the nodes to the main driver routine level. Also care has taken to deallocate
all the memory allocated earlier, so that if the solver is called after the remeshing, it can
allocate memory according to the new mesh requirements.

7.1.5.3 Modification to “Mesh Adapt”

The mesh adapt algorithm developed at SCOREC is modified such that instead of
loading the model and mesh in the mesh adapt, they are loaded in the NSpre. Further,
Mesh adapt is made as static library which can be called from the driver routine. All the
information such as mesh, model, errors, and solution at the current time step are passed
through pointers instead of reading and writing through files.

7.1.5.4 Pseudo code for the adaptive driver

main( ) {
load the model
load the original mesh
do {
    NSpre ( ) // run the pre processor
    Input( ) // Read the input data
    Process( ) // solving the problem
    Get the solution and check for tolerance
    Get the error indicators
}
If (solution does not meet specific accuracy)
    iadapt = 1
else
    iadapt = 0
Mesh adapt() // Refining the mesh
Optimize the refined mesh
} while (iadapt=1)
delete the mesh
}

7.2 Chapter summary

This chapter discussed the adaptive strategy for efficient computations. Different error indicators and a refinement strategy are presented. An adaptive driver for fluid flow problems consisting pre-processor, fluid solver and the mesh adaptation tools is discussed. The results using these strategies are presented in the following chapters.
CHAPTER 8

Incompressible Simulations

Though the goal of this research is to simulate single and multiple bubble sonoluminescence, it would be fool hardy to take the new method described heretofore into such a complex problem without first validating that all of the relevant physics can be predicted on more simple, well understood problems. This being the motivation for the incompressible multi-phase flow simulations presented in this chapter. Through this simulations the ability of the level set method to represent the interface and robustness of the method for the topological changes has been tested and compared to the available results in the literature.

8.1 Simple advection of a cylindrical bubble

To demonstrate the ability of the algorithm, a simple test problem of advecting a cylindrical bubble through a rectangular domain was considered. The liquid is flowing with a constant uniform velocity of $1 \text{ m/sec}$ in the vertical direction. The buoyancy force is not activated. Hence the bubble should simply advect through the domain with the velocity of the liquid. The domain and the initial position of the bubble are shown in Figure 8.1. The dimensions of the computational domain are $-2.0 \leq x \leq 2.0$, $-2.0 \leq y \leq 8.0$, and $0 \leq z \leq 1$ and the mesh consists of 640 hexahedral elements ($16 \times 40 \times 1$). Periodic boundary conditions are applied in $x-$ and $z-$ directions. An uniform velocity of $1 \text{ m/sec}$ is specified at the inlet and a constant pressure boundary condition is applied at the exit of the domain. The bubble radius is 1.25m, and the bubble location is such that the bottom of the bubble is positioned at $-1.25 \text{ m}$. Initially an uniform velocity of $1 \text{ m/sec}$ and zero pressure is imposed throughout the domain. The motion of the bubble through the domain is shown in Figure 8.2. Clearly the bubble is advected with the fluid velocity of $1 \text{ m/sec}$ and also the shape of the bubble is preserved along with the conservation of the volume of the bubble. It should be noted that formulations that utilize a discontinuous scalar are unable to convect the interface without any oscillations or distortions, on such a coarse mesh.
Figure 8.1: The computational domain and bubble position at $t = 0$. 
Figure 8.2: Bubble position at various time instants
8.2 Two-dimensional simulation of buoyant bubble motion at a medium Reynolds Number

The rise and the deformation of a two dimensional gas bubble in an otherwise stationary liquid contained in a vertical, rectangular container is investigated using the level set method incorporating surface tension forces.

Figure 8.3: The bubble position at $t = 0$

The density of the water and air were taken to be, $1000 \frac{kg}{m^3}$, and $1.226 \frac{kg}{m^3}$, respectively, which gives a density ratio of approximately 1,000. The viscosity of liquid phase is taken as $3.5e^{-1} \frac{kg}{m \cdot sec}$, and the viscosity of the air to be, $3.58e^{-3} \frac{kg}{m \cdot sec}$, so that the viscosity ratio between the two phases is approximately 100. The domain width is $0.1 m$ and the domain length is $0.5 m$. The position of the bubble is shown in Figure 8.3. The initial radius of the 2-D bubble was $2.5 cm$. With the above parameters, if we define the Rey-
Figure 8.4: Evolution of rising buoyant bubble at $t = 0.05$

R. Reynolds number as, $Re = \frac{(2R_b)^{3/2}}{\mu_c} \sqrt{\frac{\rho_c}{\rho}}$, and Bond Number as, $Bo = \frac{4\rho_c g R_b^2}{\sigma}$ then $Re = 100$, and $Bo = 200$. Grace [20] presented a diagram showing the effect of fluid properties and the equivalent bubble diameter on the shape and the terminal velocity of an isolated bubble. According to their experiments for the parameters chosen above (Eötvös number, $Eo = Bo = 200$, Morton number, $M = Bo^3/Re^4 = 0.08$), formation of a skirted bubble was reported. The mesh used in simulations consists of 100 x 500 x 1 elements, and a periodic boundary condition was imposed in $z$ direction. On the side planes, the normal component of the velocity ($x$-directional velocity) is specified as zero, and a zero traction was also imposed. A constant pressure boundary condition was prescribed at the outflow, and a zero velocity was assigned on the bottom of the container. Pressure is initialized to that of hydrostatic head.
When the bubble begins to rise, due to the buoyancy force acting on the bubble, the pressure gradient at the lower surface of the bubble is higher than on the top surface. The vortex sheet which develops at the surface has a rotation (Figure 8.4) which induces the motion of a jet of water that pushes into bubble from below. This phenomena is captured within the numerical simulations presented in Figures 8.4 and 8.5. These are the solutions at the time instances of $t = 0.05$ and $t = 0.15$, respectively. At this stage, the liquid jet does not effect the liquid above the bubble. The velocity of the upper surface of the bubble in comparison to the rest of the bubble is low resulting in bubble distortion. With time, the water jet from the bottom further pushes the bubble, which, causes the lower interface to move more towards the upper cap of the bubble (Figure 8.6), forming a so-called skirted bubble. Eventually the liquid jet pinches off the bubble and shreds satellite bubbles as
Figure 8.6: Evolution of rising buoyant bubble at $t = 0.3$

shown in Figure 8.6. This result is consistent with the findings of Delnoij et al. [16] who also observed the formation of skirted bubbles and shredding of satellite bubbles. Walter and Davidson [65] also observed the detachment of two small bubbles at the lower extremities of the main bubble, during the rise of two dimensional bubbles. The solutions obtained from this method are also in good agreement with the Sussman et al. [57], who carried out simulations at the same Reynolds and Bond number. In the simulations presented here, actual piercing of the bubble did not occur due to the relatively low Reynolds number. As seen in Figure 8.6 for a low Reynolds number, the liquid jet below the bubble is not strong enough to pierce causing the bubble instead to rise as a cap. Due to the re-distancing strategy adopted, the level sets remains a distance function despite the change in bubble topology. The advantage of the levels sets is also demonstrated as the shredded
bubble’s motion was predicted without altering the algorithm. However, resolving the shredded bubbles completely requires much higher resolution, thus the interest in future work on adaptivity.

8.3 Effect of Fluid viscosity on buoyant spherical bubble motion

To further demonstrate the capability of the algorithm, a three-dimensional simulation of the motion of the bubble was performed. Starting from a perfectly spherical 3-D bubble which was initially at rest, buoyancy induced motion of the bubble is studied by tracking the interface using the level set method. The fluid viscosity plays an important role in determining the involved bubble dynamics, thus manifesting the change in the shape of the bubble. Hence the effect of Reynolds number on the bubble motion is an interesting phenomena to study. We consider two cases, one with a low Reynolds number of 10, and the other case with Re=100.

For the three-dimensional simulations presented in this section, the domain is a rectangular box with a length of 6.48 $m$, and the width and thickness of the box is 2.07 $m$. The calculations are performed on a $50 \times 150 \times 50$ hexahedral elements mesh. Initially the bubble of 0.5 $m$ radius is positioned at 0.7 $m$ from the bottom of the box. The initial position of the bubble is shown in the Figure 8.7(a). All of the surfaces are modeled as slip walls. Initially the fluid is at rest, hence a zero velocity is imposed throughout the domain, where as pressure is initialized to that of the hydrostatic head.

8.3.1 Formation of a cap bubble at low Reynolds number

To investigate the viscous effects on the bubble motion, a Reynolds number of 10 and a Bond number of 5 is chosen. The ratio of density of the liquid to gas is 40, and the ratio of the viscosities is also chosen as 40. From the parameters it is clear that this is a flow with high viscosity and surface tension. Again, the results are in good agreement with the experimental findings of Grace [20]. For the chosen parameters (Eo=5, M=0.0125), the experiments resulted in the formation of a spherical cap bubble.

Figure 8.7 presents the bubble shape at different time instants obtained from simulations. Initially, the bubble rises to form an elliptical shape bubble as shown in Figure 8.7(b) ($t = 0.0625$sec) due to the pressure gradient between the upper and the lower
Figure 8.7: Rise of the spherical bubble at low Reynolds number
Figure 8.8: Vertical velocity contours shown on the $xy$-plane (low Re)
surfaces of the bubble. The bottom view at the same time instant (Figure 8.7(c)) shows the slight roll up of the lower surface. As the time progress, the liquid jet below the bubble pushes the lower surface further as shown in Figures 8.7(d) and 8.7(e). Figure 8.8(a) presents the contours of the vertical velocity at time t=1.25sec on a $xy$-plane through the center of the domain. As seen in the figure, the upper surface travels faster than the lower and also, the bottom surface gets much flatter giving the bubble a spherical cap shape 8.7(e). Further in time, as shown in Figures 8.7(f), 8.7(g) and 8.8(b), the upper surface deforms more into the cap, as the tip of the top surface travels fast compared to the rest of the bubble. The jet from the bottom is not strong enough, and hence gets decelerated due to the viscous and surface tension forces as shown in Figure 8.8(c). Also from the figure we can see that the jet speed is more diffused and hence making the bubble to broaden slightly. The bubble then continues to rise as spherical cap bubble(8.7(h), 8.8(d)).

8.3.2 Formation of a toroidal bubble at high Reynolds number
In this section, we consider a high Reynolds number of 100, and a Bond number of 50 (compared to Section 8.3.1) to study how the bubble shape can be affected by varying (decreasing in this case) the viscous and surface tension effects. Both the density and the viscosity ratios are chosen to be 80 (similar to the parameters presented in [13]). The initial conditions are similar to that of in Section 8.3.1. Figure 8.9 presents the shape of the bubble at various time instants, and Figure 8.10 display the bubble interface on a $xy$-plane cut through the domain, along with the velocity vectors.

As described earlier the liquid jet from the bottom tries to push the lower surface of the bubble (Figure 8.10(a)), which can also be seen in Figure 8.10(b). The deformation of the bubble from sphere to elliptic cap due to the jet can be seen in Figure 8.9(b). As the time progress the jet gets much stronger, and rolls the bottom interface further as seen in in Figures 8.9(d), and 8.10(c). Figure 8.9(e) shows the further deformation of the bubble at $t=0.5$ sec. At this instant of time the impinging jet from the bottom can be seen in 8.10(d). As the velocity along the axis of the bubble is higher as compared to the rest, and also the top surface of the bubble traveling faster, the bubble assumes a bell shape (8.9(e)). We can see from the figure 8.9(e) that, although the liquid pushes the bottom surface closer to the top, it does not yet pierce the top surface. Further in time at $t=0.625$sec, the jet finally pierces the top surface and detaches it from the rest of the bubble. This can be seen in Figure 8.9(f). The bottom view of the same presented in Figure 8.9(g) shows clearly the part of the interface detaching from the rest of the bubble. This can also be seen in the two dimensional view presented in Figure 8.10(e). Eventually at time $t=0.75$sec, the jet penetrates the liquid bubble and lower surface pierces the top surface completely, which can be seen clearly in Figure 8.9(f). Thus the formation of a toroidal bubble is complete as seen from the bottom view of the bubble in Figure 8.9(f). The vorticity in the bubble surface is transferred to circulation about this annular toroidal bubble (Figure 8.10(e)). The observations made in this section are in good agreement with the features of three-dimensional bubbles reported from the experiments of Walters et al. [65]. Similar numerical results were reported by Chen et al. [13].
Figure 8.9: Rise of the spherical bubble at high Reynolds number
Three-dimensional simulation of two bubble coalescence

8.4 Three-dimensional simulation of two bubble coalescence

To demonstrate the capability of the algorithm in computing interface singularities such as merging and reconnection, we consider the interaction of two bubbles of the same density under the influence of a buoyancy force. The domain is a unit cube, and the initial position of the bubbles is shown in Figure 8.11. The upper bubble is of radius 0.15 and is centered at (0.5, 0.6). The lower bubble is centered at (0.5, 0.35) with a radius of 0.1. The non-dimensional fluid density inside the bubbles is 1.0, whereas the fluid density outside the bubbles is 10.0. The viscosity of the fluid inside the two bubbles is 0.00025, and the
Figure 8.11: Initial position of the bubbles

Figure 8.12: Position of the bubbles at $t = 0.05$
Figure 8.13: Contours of speed on a cut plane at $t = 0.05$

Figure 8.14: Position of the bubbles at $t = 0.10$
Figure 8.15: Position of the bubbles at $t = 0.10$ along with the contours of pressure on a cut plane

Figure 8.16: About to merge bubbles at $t = 0.15$
Figure 8.17: A different view of the bubbles which are about to merge at $t = 0.15$

Figure 8.18: The interface of about to merge bubbles on a cut plane at $t = 0.15$
viscosity of the fluid outside the bubbles is 0.0005. The effect of surface tension was not considered here. Initially the fluid is at rest. Periodic boundary conditions are applied in all the three directions. A gravitational force is applied in the vertically downward direction. The body force was adjusted such that the net momentum flux through the boundaries of the domain was zero. In order to achieve this, an additional force term defined by $\rho_0 g$, was subtracted from the original body force term, where $\rho_0 = \alpha \rho_b + (1 - \alpha) \rho_f$ is the mean density, and $\alpha = \sum_{i=1}^{N_b} \pi d_i^2 / 6L^3$ is the void fraction, $N_b$ is the number of bubbles (here $N_b = 2$) and $d_i$ is the diameter of the $i^{th}$ bubble, and $L$ is the length of the cubic domain. This additional term prevents the uniform vertical acceleration of the whole flow field and allows for periodic boundary conditions to be used. This term is analogous to the pressure gradient generated by the base of the container, which balances the total gravitational force on the fluid. We have chosen a uniform mesh of $80 \times 80 \times 80$ hexahedral elements.

As the bubbles are lighter than the surrounding fluid, they will rise with time. Figure 8.12 shows the position of the bubbles at $t = 0.05$, the dimensionless time. Figure 8.13
Figure 8.20: The interface of merged bubbles on a cut plane at $t = 0.22$

displays the contours of speed at which the bubbles travel on a $xy$-plane cut through the center of the domain. We can see from the figure that the bottom interface of the bubbles travels faster compared to the front. However the second bubble, which is in the wake of the larger bubble, tries to move fast from the front too. As the time evolves, the upward moving jet produced by the lower bubble not only affects the larger bubble transients, but also dictates the shape of the smaller bubble as shown in the Figure 8.14. This jet creates opposite signed vorticity fields in the wake of the large bubble as shown in Figure 8.15. From the pressure contours displayed on a $xy$-plane (at time $t = 0.10$), one can observe that the lower pressure field behind the large bubble causes a strong flow on the bottom portion of the large bubble and as well affecting the front portion of the smaller bubble following the larger bubble. One can see a similar wake at the bottom of the smaller bubble too. Figures 8.14 and 8.15 clearly shows also the narrowing of the smaller bubble in the process. At time $t = 0.15$ (Figure 8.16), the front portion of the small bubble almost catches up with the bottom portion of the larger bubble. Also the vorticity created at the bottom of the larger bubble causes the bubble to roll up as shown in Figure 8.17. The
incipient merging of the interfaces is seen more clearly in Figure 8.18, in which a plane ($xy$) is cut through the domain. Figure 8.19 shows the merged bubbles at $t = 0.27$. Figure 8.20 depicts the interfaces on a plane clearly showing the merged interfaces and also some of the entrained heavier fluid inside the incident new bubble.
CHAPTER 9
Simulation of Compressible flows

This chapter focuses on the compressible flow simulations. In particular, in this chapter it is demonstrated that the newly developed algorithm can: (i) capture shocks correctly, (ii) represents the compressible gas dynamics accurately, (iii) track very complex bubble motions including the bubble implosion very efficiently, (iv) make use of unstructured and adapting grids to do so with the highest resolution and the least amount of computational effort.

9.1 Steady shock problem

To demonstrate the discontinuity capturing operator, and the refinement strategy, a classical standing shock problem is considered and solved. The selected problem consists

![Figure 9.1: Pressure predictions without the DC operator: * exact solution, - SUPG solution](image.png)
Figure 9.2: Pressure predictions with the DC operator: —— exact solution, ---- * SUPG with DC operator solution

of two flow regions separated by a shock. If the flow properties on the two sides of the shock satisfy the normal shock conditions, the shock will be stationary. The following initial conditions satisfy these conditions.

\[
\begin{align*}
M &= 2 \\
\rho &= 1 \\
u &= 1 \\
p &= 0.1785 \\
\end{align*}
\quad \text{for } x < 0; \quad \begin{align*}
M &= 0.57735 \\
\rho &= 2.6667 \\
u &= 0.37500 \\
p &= 0.80357 \\
\end{align*}
\quad \text{for } x > 0 \quad (9.1)
\]

\((\gamma = 1.4 \text{ and } c_v = 716.5)\)

The boundary conditions are set to the above data, and the \(y\)- and \(z\)-components of the velocity are set to zero on the entire domain. The computational domain dimensions are \(-19.5 \leq x \leq 19.5 \quad -0.5 \leq y \leq 0.5\), and \(0 \leq z \leq 1\) and the initial, mesh consisted of 39 hexahedral elements \((39 \times 1 \times 1)\).
9.2 Effect of discontinuity capturing operator

The computed pressure using the Galerkin/least squares without the discontinuity term is shown in the Figure 9.1. From the figure it is clear that the original position of the shock is maintained by the method, which indicates that the methods is indeed a flux conservative method. But the solution has small undershoots and overshoots near the shock. These are however, very localized and do not corrupt the solution, a small distance away from the shock. However, for the problems we are interested in, it is essential to capture these discontinuities to represent the relevant physics of the problem accurately.

Figure 9.2 presents the computed pressure using the discontinuity capturing operator. The solution obtained is free of under-shoots and overshoots. This shows that the discontinuity capturing operator is able to control the oscillations as desired. One can see from the figures that the steady shock conditions are preserved resulting in a non-smeared shock.

9.3 Adaptive mesh refinement for the shock problem

The original coarse mesh used in the simulation is shown in the Figure 9.3. It consists of 326 regions, 819 faces, 665 edges and 173 nodes. The mesh is uniform. The solution obtained from this mesh after 20 time steps converges to the order of 0.001. The contours of error distribution is shown in the Figure 9.4.

With a 20 percent refinement (i.e we refine 20 percent of the total mesh edges) specified, the refined mesh generated by the adaptive algorithm is shown in the Figure 9.5. The refined mesh has 466 regions, 1154 faces, 911 edges and 224 nodes. From the figures, we can see that the mesh is refined in those regions where error in the solution is large. That is, for the present problem, at the center of the tube, where the shock is standing. Upstream of the shock is supersonic while downstream of the shock is sub sonic. The pressure contours obtained with the refined mesh is shown in the Figure 9.6. A comparison of the solutions obtained on coarse and finer mesh are presented in Figure 9.7. From the results one can conclude that the adaptive h-refinement yielded a better solution with only a slight increase to the number of degrees of freedom. Note, if p-refinement is used to capture the shock instead (i.e., using a higher order basis function), a less sharp shock would be computed. Higher polynomial order methods tend to smear the shock where as
geometric refinement leads to a sharp shock without any under and over shoots.

9.4 Sod’s shock tube problem

The purpose of this test is to demonstrate the capability of the code to predict the compressible problems having the sharp discontinuities in the solution. A good test problem to verify the accuracy of the numerical method and the resolution of the shock is a shock tube problem, suggested by Sod [55]. The problem consists of a two different material states separated by a diaphragm. Initially the fluid is at rest and \( \rho_1 = 1.0, p_1 = 1.0, u_1 = 0.0 \) on one side of the shock and \( \rho_2 = 0.125, p_2 = 0.1, u_2 = 0.0 \) on the other
side of the shock. At time $t > 0$ the diaphragm is broken, and a shock wave travels to the right and a rarefaction wave moves towards left. Farther down the tube there will be a contact discontinuity. The solution obtained with a resolution of 100 elements at $t=0.2$sec is shown in the Figure 9.8. From the figure it is clear that the method is very well able to resolve all the shocks and the discontinuities very accurately. Also we can see that as expected across the contact discontinuity the pressure and the velocity are continuous although the density and the specific energy are discontinuous. Both the shock wave moving to the right and the rarefaction wave are predicted well at correct positions. The predicted solutions are free of numerical oscillations. The solution obtained compare very well with the numerical results obtained using high resolution schemes, and modern upwind schemes.
9.5 Implosion of a spherical shock wave using an adaptive mesh strategy

This is the simulation of a strong spherical shock wave traveling to the center of the spherical domain. Initially the spherical domain consists of a gas of uniform density and pressure. A strong shock wave is generate at $t > 0$. The wave could have been generated, for example, by a “spherical piston” which pushed the gas inward, imparting to it certain amount of energy. The origin of the shock wave does not affect the solution, except the wave generated is spherically symmetric. As the wave converges to the center the energy becomes concentrated at the front, and the wave gets strengthened. At the instant the shock front reaches the center of the spherical domain, the problem becomes nearly singular. This exercise clearly demonstrates the code’s capability to handle strong
shocks. To resolve this kind of strong shock we need a refined mesh. As the problem is fully three-dimensional, if we do the uniform refinement the mesh would be enormous. As the regions away from the shock front does not need the fine resolution we can refine the mesh just around the shock front using an adaptive refinement strategy. As the shock moves in coarsen the regions which were refined earlier and refine the regions where the new shock front is located. This strategy shows excellent savings in the computational cost.

The initial conditions are shown in the Figure 9.9. These conditions generate a shock wave moving with speed of 800m/sec, which corresponds to a mach number of 2.3. As initially the shock is located at the boundary of the domain, the mesh is refined near the boundary as shown in Figure 9.10. The refined mesh consists of 715651 regions (elements). The Figure 9.11 shows the shock front at t=0.1 sec. At this point the

Figure 9.8: Sod’s shock tube problem
Figure 9.9: Initial conditions for the imploding spherical shock wave

Figure 9.10: Adapted mesh at t=0
Figure 9.11: Spherically inward propagating shock at t=0.1sec

Figure 9.12: Mesh adapted to the shock front at t=0.1sec
shock is strengthening slowly, as one can see from the increase in the pressure at the front. Again the mesh is regenerated by coarsening farther away at the boundary as the shock no longer exists there Figure 9.12, and refining the region in the vicinity of the shock. Again the number of regions in the modified mesh is 779946. Figure 9.13 presents the strengthening shock front with time at $t=0.2$ sec, and the Figure 9.14 presents the mesh refined at this time to further resolve the shock. The velocity and the pressure distributions at $t=0.3$ seconds is presented in Figure 9.15. We can clearly see a shock accelerating towards origin. Figure 9.16 depicts the instant where the shock reached the origin. From the pressure distribution we can see a very high pressures were reached at the instant of shock collapse. This collapse occurred after 0.4680 sec indicating the rapid time scales involved. After the collapse of the shock at origin the shock wave bounces back and travels away from origin as shown in Figure 9.17. The results predicted are compared against the self-similarity solution, and the results from the simulations were in good agreement with the analytical results.
Figure 9.14: Adaptively refined mesh at t=0.2sec

Figure 9.15: Velocity and pressure distributions at t=0.3sec
Figure 9.16: At the instant of reaching origin at $t=0.4680\text{sec}$

Figure 9.17: Shock wave reflecting back from origin at $t=0.6080\text{sec}$
CHAPTER 10
Free Surface Flows

10.1 Introduction

Free surface flows are widely encountered not only in nature, but also in various industrial and environmental applications. The exchange of air between the atmosphere and water is called air entrainment or aeration. Entrainment may occur naturally or artificially. It can be observed in chemical, coastal, hydraulic, mechanical and nuclear engineering applications. Natural free surface aeration includes self-aeration in high velocity open channel flows, including air entrainment at hydraulic jumps. On the other hand artificial aeration (or forced aeration) can be created by air injection at submerged nozzles.

Many research groups have approached the study of these complex flows with numerical simulations. Computational investigations using DNS models are starting to play an increasingly important role in free surface flow research. In addition, these investigations hold the potential to enable the evaluation of the constitutive relations of two-fluid models.

10.2 Kelvin-Helmoltz stability criteria for stratified flows

Kelvin-Helmholtz (KH) linear stability analysis has been used frequently for determining whether a smooth stratified flow is stable or unstable. In the literature, two main directions are followed:

The viscous Kelvin Helmholtz (VKH) analysis, which uses the full two-fluid model and takes into account the shear stresses ([3]).

The inviscid Kelvin-Helmholtz (IKH) theory, in which the shear stresses are, neglected ([58]).

One would expect that the inviscid theory would be a good approximation for liquid of low viscosity, where as for high viscosities one will have to use the full two-fluid model in order to get correct results. Surprisingly the paper by [3] shows the results are just the opposite. For liquid of high viscosity the results of IKH theory are applicable, while there
is a large discrepancy in the results for the stability criterion between the IKH and VKH theories for low liquid viscosity.

### 10.2.1 The stability analysis of a full two-fluid model

The continuity equation for the liquid and gas are:

\[
\frac{\partial}{\partial t}(\rho_l A_i) + \frac{\partial}{\partial x}(\rho_l A_i u_i) = 0 \tag{10.1}
\]

\[
\frac{\partial}{\partial t}(\rho_g A_g) + \frac{\partial}{\partial x}(\rho_g A_g u_g) = 0 \tag{10.2}
\]

The momentum equations for each phase are

\[
\frac{\partial}{\partial t}(\rho_l A_i u_i) + \frac{\partial}{\partial x}(\rho_l A_i u_i^2) = -\tau_l S_i + \tau_l S_i - A_i \frac{\partial p_{li}}{\partial x} - \rho_l A_l g \cos \beta \frac{\partial h_i}{\partial x} - \rho_l A_l g \sin \beta \tag{10.3}
\]

\[
\frac{\partial}{\partial t}(\rho_g A_g u_g) + \frac{\partial}{\partial x}(\rho_g A_g u_g^2) = -\tau_g S_g - \tau_l S_i - A_g \frac{\partial p_{gi}}{\partial x} - \rho_g A_g g \cos \beta \frac{\partial h_i}{\partial x} - \rho_g A_g g \sin \beta \tag{10.4}
\]

In the above equations, \( A \) is the cross-sectional area, \( h \) is the liquid level or gas gap, \( p \) is the pressure, \( u \) is the axial average velocity, \( \tau \) is the shear stress, \( S \) is the perimeter over which \( \tau \) acts, \( \rho \) is the density and \( \beta \) is the angle of inclination from the horizontal (positive for the upward flow). The subscripts \( l \) and \( g \) denotes the liquid and gas respectively, and the subscript ‘i’ denotes the interface. Assuming incompressible flow and combining the two momentum equations by eliminating the pressure terms using the approximate relation

\[
p_{ig} - p_{li} = \sigma \frac{\partial^2 h_i}{\partial x^2} \tag{10.5}
\]

where \( \sigma \) is the surface tension, yields the following three equations:

\[
\frac{\partial h_i}{\partial t} + \frac{A_i}{A_i'} \frac{\partial u_i}{\partial x} + u_i \frac{\partial h_i}{\partial x} = 0 \tag{10.6}
\]

\[
\frac{\partial h_i}{\partial t} - \frac{A_g}{A_g'} \frac{\partial u_g}{\partial x} + u_g \frac{\partial h_i}{\partial x} = 0 \tag{10.7}
\]

\[
\rho_l \frac{\partial u_i}{\partial t} - \rho_g \frac{\partial u_g}{\partial t} + \rho_l u_i \frac{\partial u_i}{\partial x} - \rho_g u_g \frac{\partial u_g}{\partial x} + (\rho_l - \rho_g) g \cos \beta \frac{\partial h_i}{\partial x} - \sigma \frac{\partial^3 h_i}{\partial x^3} = F \tag{10.8}
\]
A linearization procedure, which follows the general approach proposed by ([2]) yields

\[\frac{A_i}{A_t} \sigma \frac{\partial^2 h_i}{\partial x^2} + \left[ \rho_i \frac{\partial \sigma}{\partial x} + \frac{\rho_i u_i^2}{R_i} + \frac{\rho_g u_g^2}{R_g} - (\rho_i - \rho_g) g \cos \beta \frac{A_t}{A_i} \right] \frac{\partial^2 h_i}{\partial x^2} + 2 \left( \frac{\rho_i u_i}{R_i} + \frac{\rho_g u_g}{R_g} \right) \frac{\partial^2 h_i}{\partial x \partial t} + \left[ \frac{\rho_i}{R_i} + \frac{\rho_g}{R_g} \right] \frac{\partial^2 h_i}{\partial t^2} = \left[ - \frac{A_t}{A_i} \frac{\partial E}{\partial h_i} \frac{\partial h_i}{\partial x} + \frac{v_i}{R_i} \frac{\partial E}{\partial u_i} + \frac{v_g}{R_g} \frac{\partial E}{\partial u_g} \right] \frac{\partial h_i}{\partial x} + \left[ - \frac{1}{R_i} \frac{\partial \sigma}{\partial u_g} + \frac{1}{R_g} \frac{\partial \sigma}{\partial u_i} \right] \frac{\partial h_i}{\partial t} \right] (10.9)

Where \(\tilde{h}_i\) is the perturbed liquid level and \(R\) is the phase hold up. Note all the terms that are in square brackets in (10.9) are evaluated at the steady-state unlike the terms in (10.1) - (10.6), where these terms are local transient values. Substituting the perturbed liquid level, \(\tilde{h}_i = c(e^{i(\omega t-kx)})\) (10.10) into (10.9) yields the following dispersion relation:

\[\omega^2 - 2(ak - bi)\omega + c k^2 - d k^4 - ek i = 0\]

\[a = \frac{1}{\rho} \left( \frac{\rho_i u_i^2}{R_i} + \frac{\rho_g u_g^2}{R_g} \right) ; \quad b = \frac{1}{2 \rho} \left( \frac{\partial E}{\partial u_i} \right)_{u_{g*},R_i} - \left( \frac{\partial E}{\partial u_g} \right)_{u_{g*},R_g} \right] \right) (10.11)

\[c = \frac{1}{\rho} \left( \frac{\rho_i u_i^2}{R_i} + \frac{\rho_g u_g^2}{R_g} - (\rho_i - \rho_g) g \cos \beta \frac{A_t}{A_i} \right) \]

\[d = \frac{\sigma A_t}{\rho A_t} ; \quad e = -\frac{1}{\rho} \left( \frac{\partial \sigma}{\partial R_i} \right)_{u_{g*},u_{g*}} ; \quad \rho = \frac{\rho_i}{R_i} + \frac{\rho_g}{R_g} \]

The solution for \(\omega\) is,

\[\omega = (ak - bi) \pm \sqrt{ \left( (a^2 - c) k^2 - b^2 + d k^4 + (ek - 2ab k) i \right) } \] (10.12)

So the steady state solution is unstable whenever the imaginary part of \(\omega\) is negative leading to exponential growth of the perturbation. For the case of inviscid flow a simple expression for \(\omega\) is obtained:

\[c \equiv \frac{\omega}{k} = \frac{\left( \frac{\rho_i u_i}{R_i} + \frac{\rho_g u_g}{R_g} \right)}{\left( \frac{\rho_i}{R_i} + \frac{\rho_g}{R_g} \right)} \pm \sqrt{ \left( \frac{(\rho_i - \rho_g) g}{R_i R_g} \left( u_g - u_i \right)^2 \right) + \frac{\sigma k^2}{R_i R_g} } \] (10.13)

As long as the discriminant in the square root is positive, the amplification in this case is zero. When the discriminant is negative, two complex conjugate solutions exists, and the one with negative sign is the one that contributes to the instability. For the case of viscosity, the condition for the marginal stability can be obtained from (10.12) for the
special case when imaginary part is equals zero. Thus leading to the following stability criteria for the viscous case:

\[
\left( \frac{e}{2b} - a \right)^2 - \left( a^2 - c \right) - dk^2 < 0
\]  \hspace{1cm} (10.14)

The last three terms in (10.13) can easily be recognized to be same as the well-known KH instability of one-dimensional case with no viscous effects. The first term reflects the additional effect of the shear stresses, which tends to amplify any disturbance in the film thickness. The above described analysis was used in the analysis of the results obtained from the simulations.

**Figure 10.1: Boundary Conditions**

### 10.3 Inviscid two-phase flow

The free surface flow in a rectangular domain is studied in this section. The problem consists of two flow regions separated by an interface. The initial and boundary condition as shown in the Figure 10.1. The length of the domain is chosen such there are exactly ‘n’ integer number of waves in the domain. The length is calculated based on the fast growing wave length predicted from the stability analysis described above. If the relative velocity between the phases is less than a certain critical velocity, the interface remains smooth and stable, and if some instability arises they won’t grow. On the other hand, if the relative velocity exceeds that value, the instabilities grow and eventually the interface gets distorted to form rolling waves. In all the cases the boundary conditions are set as
shown above, and initially the y- and z-component of the velocities are set to zero on the entire domain.

In this case the viscous forces are neglected by taking the viscosity of both the phases to be zero. The ratio of liquid density to gas density is 1000. The relative velocity between the phases is set to 15m/sec, liquid being still, and gas moving with a velocity of 15m/sec. For this relative velocity the analysis shown an unstable range of lengths being from 1.98mm to 142mm. The domain length is chosen as 98.5mm, and the height of the domain as 2m, with the interface at zero. The model and the grid used in the simulations is shown with the interface in Figures 10.2 & and 10.3. In Figure 10.3, the interface position at $t=0.008$ sec is also shown along with the mesh. The velocity distribution at $t=0.0082$ sec and $t=0.0086$ sec are shown in the Figures 10.4 & and 10.5 respectively. One can see from the results that the interface instabilities grow quickly. The velocity and
Figure 10.3: Mesh used and the interface at $t=0.008$ sec

Figure 10.4: Velocity Distribution at $t=0.0082$ sec
Figure 10.5: Velocity Distribution at t=0.0086sec

Figure 10.6: Velocity Distribution at t=0.009sec
Figure 10.7: Pressure Distribution at t=0.009sec

Figure 10.8: Interface and Level sets at t=0.009sec
the pressure distributions at \( t=0.009 \text{ sec} \) are shown in the Figures 10.6 & and 10.7. To verify that the level sets are following the interface closely, the interface location along with the level sets is shown in the Figure 10.8 at \( t=0.01 \text{sec} \). From the Figures 10.6 & and 10.7, one can see that the wave formation is in consistent with the velocity and the pressure predictions. One can also observe the vortex formation inside the rolling waves due to the low pressure in the interior of the waves.

From these simulations, the dominant unstable wave length is found. The wave length obtained is 24.6mm which is in good agreement with the analysis in (10.14).

### 10.4 Chapter summary

This chapter demonstrated the application of developed algorithm for solving free surface flows. The Kelvin-Helmholtz (KH) linear stability analysis for stratified flows was discussed. The next chapter presents the hydrodynamic solution of a Rayleigh-Taylor instability.
CHAPTER 11
Study of Rayleigh-Taylor Instabilities

Although the nonlinear evolution of the Rayleigh-Taylor instabilities have been studied widely, this problem is far from being completely resolved. A Rayleigh-Taylor instability occurs when an interface between two different fluids with different densities experiences a pressure gradient which opposes the density gradient. This interface is unstable and any disturbances tend to grow, leading to the penetration of both fluids into each other. If the initial interface displacement is random, the Rayleigh-Taylor instability usually evolves into complicated turbulent mixing.

The Rayleigh-Taylor instability has attracted much attention because of its significance in both fundamental research and engineering applications. In fundamental research, the Rayleigh-Taylor instability provides an ideal system to study multiphase hydrodynamical instabilities, turbulent mixing, and supernovae dynamics. In engineering applications, the Rayleigh-Taylor instability plays a key role in inertial confinement fusion.

Much knowledge concerning the Rayleigh-Taylor instability has been gained in the past 50 years. Following Sharp ([54]), we can roughly divide the growth of the instability into four stages. In the first stage, an amplitude of a surface perturbation is much smaller than its wave length and the amplitude grows exponentially with time([60]). The Rayleigh-Taylor instability in this stage can be studied using linear stability theory and the linear growth rate has been found as a function of the density ratio, the viscosities, the surface tension, and the compressibility ([60]). Linear perturbation theory fails when the amplitude of the perturbation grows to 10% - 40% of its wavelength. This failure of linear perturbation theory marks the beginning of the second stage of the Rayleigh-Taylor instability. During the second stage, the perturbation grows nonlinearly to form “bubbles” of light fluid rising into the heavy fluid and spikes of heavy fluid falling into the light fluid. This stage of development is strongly influenced by three-dimensional effects and the density ratio. The third stage of Rayleigh-Taylor instability is characterized by the development of additional structures on the spikes and by interactions among the
bubbles. The Kelvin-Helmholtz instability begins to develop at this time and the heavy fluid begins to roll up along the sides of the spikes to form “mushrooms”. However, this phenomena is more pronounced when the density ratio is low. Eventually this evolves into turbulent or chaotic mixing which dominates the fourth stage of the Rayleigh-Taylor (RT) instabilities.

11.1 Two-dimensional simulation of Rayleigh-Taylor instabilities

To study the details of the interface dynamics numerically, a two-dimensional Rayleigh-Taylor (RT) instabilities were simulated. The problem was solved in a domain with height to width ratio of 4:1. The interface is located right at the center of the column (i.e. at \( y = 0 \)). The upper part (\( y > 0 \)) contains the heavier fluid with a density of 2.0, and the lower part (\( y < 0 \)) contains the lighter fluid with a density of 1.0. Hence, the Atwood number is \( A = \frac{(\rho_2 - \rho_1)}{(\rho_h - \rho_l)} = 0.3333 \). Both the fluids were governed by the ideal gas law. The pressure is initialized to the hydrostatic equilibrium. The initial instability is caused by perturbing the vertical component of velocity. All of the domain boundaries were modeled as slip walls. A body force of 1.0 was applied in the vertically downward direction. For a Mach number \( M \ll 1 \), the flow scale is determined by the initial perturbation. To generate the steady motion in RT instability, the imposed initial perturbation must be of small amplitude. Hence we consider an amplitude of \( a_0 = M_0 \sqrt{\frac{T}{T}} \), and \( M_0 = 0.1 \). The initial velocity perturbation is given by, \( v = a_0 \cos^6(\pi y) \cos(4\pi x) \).

Figure 11.1 shows the contours of density at successive intervals of time obtained on a \( 50 \times 200 \times 1 \) mesh. For the small values of Atwood number used here, the instabilities evolve slowly. From the figure, we can see that liquid falls initially as a column, and with time, it develops a spike at the walls, and a “bubble” at the center. Eventually, the typical “mushroom cap” RT instabilities evolved. However one can see that the Kelvin-Helmholtz instabilities are not that dominant, hence the liquid falls without generating eddies.

Furthermore, to test the effect of mesh refinement on these instabilities, the solution obtained on \( 50 \times 200 \times 1 \), \( 100 \times 400 \times 1 \), and \( 200 \times 800 \times 1 \) meshes is compared in Figure 11.2. For a two-fluid system, after a short period of time of exponential growth, the bubble velocity approaches a terminal value. The first important observation from the
Figure 11.1: Two-dimensional RT instabilities
Figure 11.2: Comparison of RT instabilities obtained under mesh refinement

Figure 11.2 is that the position of mushroom head at $t = 2.5$ is about the same for all the three meshes. This indicates that the velocity indeed does not change with refinement, indicates a converged solution. In addition, with refinement, the secondary structures evolved. With refinement, more eddies are captured, and an increased zone of mixing is predicted.

11.2 Three-dimensional simulations of Rayleigh-Taylor instabilities

In this study, direct numerical simulation of three-dimensional RT instability were also carried out using level set approach. This study focuses on the evolution of three dimensional structure of the interface, and the non-linear growth of 3D interface. The dimensions of the domain are such the the ratio of lengths in $x-$, $y-$, and $z-$ directions is
Figure 11.3: Evolution of 3D interface from a single mode perturbation
Figure 11.4: Cross-sectional views of the interface at a central vertical plane
Again the heavier fluid is located at the upper half of the domain with the interface in the middle of the domain. The heavier fluid was of density 1.5 and the lighter fluid was of density 1.0, hence the Atwood number was 0.5. The instabilities were initialized by perturbing the interface with a single mode perturbation, \( h(x, y) = a_0 W [\cos\left(\frac{2\pi x}{W}\right) + \cos\left(\frac{2\pi y}{W}\right)] \), where, \( h \) is the height of the interface, and \( W \) is the box width. The simulation was carried out on a \( 50 \times 200 \times 50 \) mesh.

The evolution of interface in the three-dimensional RT instability is shown in Figure 11.3. As expected, the heavy and light fluids penetrate into each other as time increases. During the early stage the interface develops very slowly. With time, the light fluid rises to form a “bubble” and the heavy fluid falls to generate a spike. Furthermore, there is an additional feature that distinguishes this interface from that observed in a 2D RT instability; that is, the existence of saddle points at the middle of the sides of the box. The evolution of the interface around these saddle points is one of the unique features of the three-dimensional RT instability. For this typical case, the Kelvin-Helmholtz instability is not dominant and won’t develop until later stages. At \( t=0.6 \) sec (Figure 11.4(e)), the first roll-up of the fluid appears in the neighborhood of the interface. The roll-up at the edge of the spike starts at a later time (Figure 11.4(f)). A similar evolution of interface was observed in other numerical simulations.

To look more closely at the interface evolution, a cross-sectional view of the interface is presented in Figures 11.4. The interface looks very much like those in the two-dimensional RT instability. However, the two layer roll-up phenomena is captured only in the case of three-dimensional simulations. These two layer roll-ups should not be confused with the multiple roll-ups observed in the two-dimensional case. The former are attributed to the instabilities at two different locations, namely, the spike tip and the saddle point; while the latter is due to the Kelvin-Helmholtz instability in the shear layers.

### 11.3 Chapter summary

In this section the ability of capturing the instabilities is demonstrated through the two-dimensional simulation of Rayleigh-Taylor instabilities. Additionally, the effect of refinement on secondary instabilities is shown. Evolution of 3D structures is studies using the three-dimensional simulation of RT instabilities. In addition to the “bubble” and spike
fronts, we observed the evolution of a two layer roll-up in case of 3D simulation. The predictions are in good agreement with the results of other investigations.
CHAPTER 12
Hydrodynamic Simulation of Air Bubble Implosion

12.1 Introduction

The violent collapse of vapor/gas bubbles leads to a number of phenomena of interest in science and engineering including; cavitation damage, sonoluminescence, sonochemistry, sonofusion, etc. Owing to the great practical importance of this phenomena, scientists and engineers have made a substantial effort to understand the hydrodynamics involved through both experimental and theoretical investigations. Seminal work on cavitation and bubble dynamics was done by [49], who considered both empty and the gas filled cavities collapsing in an incompressible liquid, neglecting surface tension and viscous effects. The Rayleigh-Plesset equation [49, 34, 46, 37] extended Rayleigh’s analysis by include surface tension and viscosity. Later, many researchers have refined these analyses by considering the effects of thermal conduction in the gas [41, 39], and accounting for the compressibility of the surrounding liquid [24, 47, 35]. However, in all these studies the hydrodynamic equilibrium of the gas phase was assumed.

In addition, many of these models also assumed that the gas was heated isentropically. During bubble implosions, however, experimental findings indicate gas temperatures on the order of several thousand degrees, which implies a very high compression of the gas phase. To address this issue, researchers [70, 36, 38] have considered more complete solutions to the equations of motion. The numerical studies of [70, 36] showed that a shock is expected inside the gas bubble. However these studies were performed assuming spherical symmetry (i.e. considering only the radial direction). The most detailed numerical study published to date was that of [36] and [71]. Unlike [70], they did not obtain the dynamics of the bubble from the Rayleigh-Plesset equation. Rather they performed a complete analysis of the nonlinear hydrodynamics of the liquid surrounding the bubble using finite difference discretization of the conservation laws. However, analysis by [36] was unsteady but one dimensional. Furthermore, they did not consider viscous and surface tension effects. On the other hand, [71] solved Navier Stokes equations in 3-D using a finite difference/front tracking technique. But they assumed the surrounding
fluid as incompressible, the pressure inside the bubble as uniform and neglected thermal effects.

Although the previous studies recognized the fact that the analysis based on the spherical symmetry cannot be rigorously correct [46], very few attempts [71] have been made to solve the full three-dimensional (3-D) nonlinear problem. The analysis of the collapse and rebound of an imploding bubble is complex due to the highly transient and nonlinear nature of the problem. In this section, we have a detailed 3-D analysis is presented considering the complete set of the conservation laws (i.e., conservation of mass, momentum, and energy) for both liquid and gas phases (Section 3.1.1). The motion of the bubble is implicitly obtained as a part of the solution rather than from a Rayleigh-Plesset analysis. Additionally, the effects of viscosity and surface tension are included in the simulations.

12.2 Bubble motion and flow field during the initial stages of implosion

Assuming sphericity, the collapse of a gas filled cavity in an incompressible fluid can be determined using the Rayleigh equation [49]. Knowing the evolution of the bubble radius, one can calculate the evolution of pressure and flow field. This is well described by [38]. The following analysis describes how to obtain the flow field based on the bubble motion. DNS does not necessarily require this solution, however the following analysis can be used to generate the appropriate initial conditions for the early stages of bubble implosion. The details of its use are given in Section 12.3.

Consider the spherically symmetric radial flow of an incompressible liquid inside a spherical domain. A gas bubble is located at the center of the domain. The momentum balance in radial direction is given by,

\[
\rho_l \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} \right) = -\frac{\partial p}{\partial r}
\]  

(12.1)

where, \( r \) is the radial distance from center of the bubble, and \( u \) is the velocity in the radial direction. If \( R(t) \) is the instantaneous bubble radius, then the conservation of mass
dictates that,
\[ u = \frac{R^2 \dot{R}}{r^2} \]  
(12.2)

Since the radial velocity can be expressed in terms of velocity potential as, \( u = \frac{\partial \phi}{\partial r} \), we can write,
\[ \phi = -\frac{R^2 \dot{R}}{r} \]
(12.3)

Substituting the potential for \( u \) in (12.1) yields,
\[
\rho_l \left[ \frac{\partial}{\partial r} \left( \frac{\partial \phi}{\partial t} \right) + \frac{\partial}{\partial r} \left( \frac{u^2}{2} \right) \right] + \frac{\partial p}{\partial r} = 0
\]
(12.4)

Integrating the above equation with respect to \( r \) yields the Bernoulli equation,
\[
\rho_l \left[ \frac{\partial \phi}{\partial t} + \frac{u^2}{2} \right] + p = F(t)
\]
(12.5)

As \( r \to \infty, \phi = 0, p = p_\infty \), so, \( F(t) = p_\infty \). Therefore,
\[
p = p_\infty - \rho_l \left( \frac{\partial \phi}{\partial t} + \frac{u^2}{2} \right)
\]
(12.6)

Substituting for \( \phi \) and \( u \) in terms of \( R \), (12.2) and (12.3),
\[
p(r, t) = p_\infty - \rho_l \left[ -\frac{R^2 \ddot{R}}{r} - \frac{2R \dot{R}^2}{r} + \frac{\dot{r}^2 \dot{R}}{2r^4} \right]
\]
(12.7)

Integrating (12.7) from \( r = R(t) \) to \( \infty \), we obtain [49],
\[
R \ddot{R} + \frac{3}{2} \dot{R}^2 = \frac{p_a - p_\infty}{\rho_l}
\]
(12.8)

where \( p_a \) is the liquid pressure at the liquid/bubble interface (uniform pressure is assumed for the sake of initial conditions, and this is a valid assumption for the initial phase of bubble collapse) given by,
\[
p_a = \left( p_0 + \frac{2\sigma}{R_0} \right) \left( \frac{R_0}{R} \right)^{3\gamma} - \frac{2\sigma}{R}
\]
(12.9)

\( R_0 \) is the initial bubble radius at \( t = 0 \) and \( p_0 \) is the liquid equilibrium pressure when
bubble has its initial radius. $\gamma$ is the polytropic index of the gas. For isothermal conditions $\gamma = 1$, where as for adiabatic conditions $\gamma$ is the ratio of the specific heats. Combining (12.8) & (12.7) yields an equation governing the pressure distribution:

$$
p(r, t) = p_\infty \left[1 - \frac{R}{r}\right] + p_a \frac{R}{r} + \rho_i \frac{R \dot{R}^2}{2r} \left[1 - \frac{R^3}{r^3}\right]
$$

(12.10)

The FEM based hydrodynamic shock code, which is primarily discussed in this paper can be initialized using (12.7) and (12.2).
12.3 Imploding air bubble in water

Let us now consider the 3-D hydrodynamic shock code analysis of an imploding air bubble which is initially spherical. The primary purpose of this study is to demonstrate the numerical method for the study of bubble implosion, to obtain the full 3-D transient DNS solution and to appraise any shape instabilities in the fully non-linear analysis.

We consider a spherical air bubble which is initially at atmospheric pressure and is driven by the surrounding pressurized liquid at 100 atmospheres. The initial radius of the bubble was 10 $\mu$m, and the radius of the computational domain was 50 $\mu$m. The initial conditions chosen are typical parameters in sonoluminescence and cavitation. To save the computational time, we have derived the initial conditions analytically to start the simulations. In particular, we have solved the ordinary differential equations presented in Section 12.2 until $t = 0.06 \mu$s, and prescribed the obtained pressure and the velocity as the initial conditions to start the DNS. The usage of this initial conditions saves computational time, but is not a requirement to start the calculations. The rationale behind choosing to
Figure 12.2: Early stages of bubble implosion
Figure 12.3: Imploding bubble - later stage
solve only up to a particular time is that we don’t exceed a reasonable Mach number for the interface velocity. For the particular choice of $t = 0.06 \mu s$, the radial interface velocity is $80.77\text{ m/sec}$, hence the gas Mach number is approximately 0.3. At this instance the predicted value of the radius of bubble is $7.95\text{ \mu m}$. The spherically symmetry is still valid at this instance and indeed until much later in time, as confirmed from our numerical results.

The nonuniform mesh used for the simulation is shown in Figure 12.1. The figure shows a plane view of the three-dimensional mesh. One can see from the figure, various levels of mesh refinement. The inner most fine mesh affords adequate resolution of the singularity when the converging shock collapses at the center. The next level of finer mesh resolves the interface and the vicinity of the shock accurately (a series of grid refinements were done to verify local accuracy needs). The early stages of the bubble implosion are shown in Figure 12.2. A plane is cut through the center of the domain and pressure contours are displayed on this plane. The solid black line on the contour plots represents
Figure 12.5: Collapsing shock
Figure 12.6: Shock collapse and rebound

the interface location on this plane. Also shown in each sub figure is the three-dimensional bubble to show the changes in bubble size and shape. The initial pressure profile imposed to start the numerical calculations is shown in Figure 12.2(a). During the initial stages of implosion, the inertial motion of the liquid compress the gas bubble. As a result the pressure inside the bubble begins to increase (Figure 12.2(c)). Figures 12.2(a) & 12.2(b) & 12.2(c) clearly show the decrease in the size of the bubble. We note that during this early stage of the bubble collapse, the bubble remains spherical, and the pressure varies smoothly inside the liquid phase. Figure 12.3 shows later stages of the implosion, and
Figure 12.7: Diverging shock and rebounding bubble
Figure 12.8: Velocity distribution during the implosion displayed on a $xy$-plane
Figure 12.9: Velocity distribution during the implosion displayed on a $yz$-plane
Figure 12.10: Velocity distribution during the implosion displayed on a $zx$-plane
Figure 12.4 shows the variation of pressure along the radius of the bubble in the early stages of implosion. In this time frame, the bubble continues to decrease in the size and the pressure wave in liquid phase is approaching the interface. As it moves towards the interface, the intensity increases due to the converging geometry. The pressure starts to build at the interface (Figure 12.4) and due to compression an increase in the gas pressure inside the air bubble can be observed. However, the pressure is uniform inside the air bubble. Although the exact mechanism for the generation of a shock has not been
identified [36], with time a shock is being developed. At t=0.091 $\mu s$ (Figure 12.4), a shock is formed near the interface. Comparing the Figures 12.3(a) & 12.3(b) & 12.3(c), one can also observe that the bubble is beginning to deviate from its spherical shape. The collapse of the shock is shown in Figure 12.5. In this figure, in addition to the actual bubble, a zoomed view of the bubble is also presented to show the deformed ellipsoidal shape more clearly. Also in this figure, the pressure distribution inside the gas phase is shown separately. At time t=0.092 $\mu s$, the shock at the interface moves into the gas...
Figure 12.13: Solution plotted along z-axis

phase, Figure 12.5(a) & Figure 12.4, and the bubble starts collapsing rapidly. As the bubble collapses, the spherically convergent gas shock becomes much stronger and the pressure inside the gas interior is of the order of $10^4$ atmospheres (Figure 12.5(b)). In addition, the interface velocity increases due to spherical convergence.

During the implosion process, the bubble is compressed and the gas inside the bubble accelerates. The velocity distribution and the interface configuration are shown for three different planes in Figures 12.8, 12.9 and 12.10. These figures display the velocity
Figure 12.14: Comparison of radii

vectors in the vicinity of the bubble along with the interface indicated by the dark line. The region focused on in each sub-figure varies, as we want to show the distributions around the interface, the location of which is constantly changing. The density of the vectors also vary as the mesh size varies in those regions. The fluid moves inward until \( t = 0.095\mu s \) (Figures 12.8, 12.9 & 12.10). Figure 12.6(a) shows the instant of the shock collapse at the center of the bubble at \( 0.095\mu s \). At this instant the air is predicted to be compressed to a thousand times normal density. However, the perfect gas law is no longer valid for these conditions. After the collapse of the shock at the center, we can see from the Figures 12.8(d), 12.9(d) & 12.10(d), that the gas reverses its direction and starts moving outward inside the bubble.

Figures 12.11, 12.12 & 12.13 display different projections of the velocity, pressure, and temperature profiles along with the temporal variation of bubble radius. The velocity, pressure, and temperature are plotted along the radius of the domain. The \( x \) axis represents the radial distance from the center of the domain in \( \mu m \). Figures 12.11(a), 12.12(a) & 12.13(a) show that the fluid moved inward until \( t = 0.095 \). Also from the figures, it can be observed the velocity is almost linear inside the gas phase. The pressure profiles shown in Figures 12.11(b), 12.12(b) & 12.13(b) depicts the converging shock as well as the diverging shock wave. However, in the later stages of implosion \( (t \geq 0.093) \), the pressure inside the gas phase is almost uniform. This is due to the high speed of sound in gas and the radius of the bubble being small at these instants. The high pressures attained
Figure 12.15: Variation of distortion amplitude

at the instant of shock collapse is on the order of $10^4$ atmospheres. In Figures 12.11(b), 12.12(b) & 12.13(b) the interface location is marked by ‘*’. Figures 12.11(c), 12.12(c) & 12.13(c) show the high temperatures near the center of the bubble during the implosion. The reflection of the converged shock from the center creates a spherically divergent shock. Figure 12.6(b) shows the rarefaction wave moving away from the center, however, at this instant the bubble is still collapsing. The large pressures and temperatures decrease
rapidly behind the shock due to hydrodynamic expansion. Once the reflected shock moving outward and crosses the interface (Figure 12.6(c)), the bubble stops collapsing and starts rebounding. Figure 12.7 shows the rebounding of the bubble after collapse. Figures 12.7(b) & 12.7(c) shows that the expanding bubble is regaining its spherical shape.

**12.3.1 Hydrodynamics of collapse**

Although we have stated that the bubble is still collapsing from $t = 0.095\mu s$ to $t = 0.096\mu s$, it is not obvious from Figures 12.6(a) & 12.6(b). However, this would be more clear if we take a look at the temporal variation of the radius. The temporal variation of the bubble radius was measured along three different axis and compared with the analytical predictions based on Rayleigh-Plesset analysis in Figure 12.14. This figure shows that during this period the size of the bubble measured along $x$ and $z$ axis is almost constant. On the other hand, because the shape of the bubble is no longer spherical, the radius measured along $y$ axis is still decreasing. At $t = 0.097\mu s$, the outward moving shock crossed the bubble’s interface, and the radius measured along $y$ axis also starts decreasing(Figure 12.12(d)). Therefore, the bubble’s radius continues to decrease in size until the reflected shock wave crosses the interface.

As seen in Figures 12.3 & 12.6, it is clear that bubble is no longer spherical during the final stages of the implosion. Figures 12.8, 12.9 & 12.10 also shows the non-spherical bubble shape in different planes. For the first time, through DNS, this non-sphericity is captured. Although there have been postulations about the asphericity of the bubble collapse leading to sonoluminescence (SL), none of the previous works actually showed its existence. Weninger et al. [67] postulated that the angular correlations in sonoluminescence path suggest the non-sphericity of a collapsing bubble. They conclude that the long time decay of the angle dependent correlation is due to the refraction of the SL photons by the interface of a non-spherical bubble. Their studies predicted that the dipole component in the broadband measurements is due to the asphericity of the collapse.

The asphericity of the bubble can be attributed to many factors, such as inherent dynamical instability of contracting bubbles, the proximity of solid boundaries or free surfaces, and buoyancy effects [46]. Our DNS shows that the deviation from spherical shape is due to the excitation of non-spherical acoustic modes of the gas bubble during
Brenner et al. [9] in their work on acoustic energy storage in SBSL, showed that the most easily excited acoustic modes of gas are not spherically symmetric. Moreover, the observance of elliptic shape bubble can be further explained as ellipticity is the leading order, quadrupolar, form of a convolution instability [67]. However, it was not clear whether the fourth order perturbations would be dominant during the collapse.

Hence, to further investigate the shape instabilities observed during the collapse stage, we performed an analysis on the growth of instabilities for a spherical interface. Plesset et al. [45] studied the problem of a spherical interface between two immiscible fluids of different densities in accelerated motion analogous to the planar interface studies by Taylor [60]. Later, Plesset and Prosperetti [46] further addressed the problems associated with the dynamics of non-spherical bubbles. The radius vector of the distorted bubble’s interface can be represented as,

\[ r_s = R + \sum a_n Y_n \] (12.11)

where \( R(t) \) is the instantaneous radius of the interface given by (12.8). \( Y_n \) is a spherical harmonic of order \( n \) and the \( a_n(t) \)'s are the amplitude of the spherical harmonic components of order \( n \). The stability of the spherical interface can be determined based on whether the interface distortions of small amplitude grow or diminish. For small perturbations to the interface (\( |a_n(t)| \ll R(t) \)), the growth in amplitude of the perturbation is governed by [45], [46],

\[ \ddot{a}_n + \frac{3}{R} \dot{R} \dot{a}_n - A a_n = 0 \] (12.12)

where,

\[ A = \frac{[n(n-1)\rho_i - (n+1)(n+2)\rho_g] \ddot{R} - (n-1)n(n+1)(n+2)\sigma/R^2}{[n\rho_i + (n+1)\rho_g]R} \] (12.13)

Solving (12.12) together with (12.8) yields the evolution of perturbation amplitude with time. Figure 12.15(a) shows the growth of amplitude with time for different \( n \). We can see that the instabilities do not grow appreciably until the final stages of collapse. Once they start growing during the final stage of collapse, the fourth order harmonic is indeed the dominant one. The variation of the distortion amplitude with mean bubble radius is
shown in Figure 12.15(b). One can observe that as the radius of the bubble decreases, the distortion amplitude oscillates in sign with increasing frequency and magnitude. From both Figures 12.15(a) & 12.15(b) it can be noted that the fourth order harmonic has the maximum absolute magnitude compared to other orders. Hence the observance of shape instabilities implies excitation of the acoustic modes of gas. Furthermore, the elliptic shape of the bubble can be explained by the fact that the fourth order harmonics grow faster resulting in an ellipsoidal shaped bubble.

From the pressure distributions (Figures 12.11(b), 12.12(b), 12.13(b)), one can see that the collapse of the shock and the peak pressure occurs at the same instant in all the directions. Hence the shock itself is spherically symmetric although the bubble is not. This is an important observation, since it implies that the bubble distortion does not degraded the compression process inside the imploding bubble. The temperature profiles (Figures 12.11(c), 12.12(c), 12.13(c)) reflect the compression experienced by the fluids. Inside the gas phase, the temperatures rises to thousands of Kelvin as the gas becomes highly compressed. The converging shock first rapidly increases the velocity inside the bubble (Figures 12.11(a), 12.12(a), 12.13(a)), but near to the final convergence of the shock wave, the velocity is reduced due to the resistance of the gas to be compressed further. The flow reverses its direction outward, once the shock has collapsed at the center of the bubble. These observations are similar in all the three directions. However the change in radius of the bubble measured along the three axial directions are different. Importantly, the radius measured along the $y$-direction decreases even after the shock collapse, while the radii measured along $x$ and $z$ directions is almost stagnant during this interval. Figure 12.14 compares the analytical (Rayleigh) predictions with the numerical predictions in several planes. We can see that although the slope remains similar, there is a shift in the curve during the final stage of implosion and this continues through the rebound. This can be attributed to the nonlinearities playing a role in case of the 3-D predictions. We can note that the radius does vary in all three directions due to the spherical asymmetry.

In addition, the simulation results clearly indicates interfacial instabilities along with shape instabilities. The onset of interfacial instabilities can be attributed to the higher interfacial velocities. In the present simulations, they are prominent only during the time
interval in which, the interfacial velocities are higher than the speed of sound of the gas. Until $t = 0.09 \mu\text{sec}$ the interface velocity is below the speed of sound in the gas and the bubble interface remains smooth and spherical. However, once the interface velocity gets to a Mach number higher than unity, during the period of violent collapse, Figures 12.8, 12.9 & 12.10 show that instabilities become significant. Furthermore, interfacial instability vanishes once the bubble’s interface velocity reduces during the later stages of rebound. It is important to note that the observance of the rough bubble interface is due to the physical instabilities rather than insufficient mesh resolution. Indeed care has been taken to have a quality mesh with the required number of elements, to represent the bubble’s interface even at the minimum radius of the bubble. Although we have not presented the results here, the simulations were performed for different resolutions of the mesh, and it did not change the qualitative nature of the results on the interfacial and shape instabilities. We would also like to emphasize that neither the time at which the minimum bubble volume was achieved, nor the characteristics of the collapse and rebound changed with the resolution of the mesh (i.e., nodal convergence was achieved).
CHAPTER 13
Conclusions

An adaptive, stabilized finite element method was developed for evaluating transient multiphase flows using a level set approach. The formulation was applied to evaluate multiphase incompressible/compressible flows in two and three dimensions. The development and implementation of the formulations are general, allowing three-dimensional simulations on arbitrary, unstructured meshes to be carried out on parallel computers. Additionally, the formulation derived is purely Eulerian, and no explicit tracking of the interface is required. The interface is readily obtained by locating the zero level set of the level set field. The robustness of the method to compute the flows with the large density and viscosity differences is demonstrated. The surfaces tension forces were also accounted for in the simulations. The re-distancing strategy allows the distance field to be maintained and assures mass conservation. The two- and the three-dimensional results clearly show the ability of the method to handle the bubble coalescence and breakup. Furthermore, a fourth order accurate, explicit time integrator has been implemented to advance the semi-discrete weak form in time.

The research presented clearly demonstrates the following achievements:

1. Implemented the level set method in a finite element framework
2. Successfully predicted multiphase flows with rapidly changing interfaces
3. Employed discontinuity capturing for evaluating flows with strong shocks
4. Demonstrated the ability to represent compressible gas dynamics
5. The Novel implementation of Ghost fluid method in a finite element framework
6. Efficient computations using an adaptive mesh strategy
7. Validated available results in literature

Using the technique developed, studies of a bubble rising in viscous fluid were carried out with the level set method. The method enabled the study of a complex and
highly transient formation of a toroidal bubble from an initially spherical shape. Instead of specifying a toroidal (e.g. Lundergren & Mansour 1991) geometry a priori as done in previous studies, the present study involves simulations of the entire process from the initial deformation of a spherical bubble to the formation of a toroid, including the development of a liquid jet below the bubble and its impact on the top surface of the bubble. The present work also accounts for the effect of surface tension and viscosity of the liquid on the formation of the toroidal bubble. The computed results have shown that as the effect of viscosity and surface tension increase, the impinging liquid jet beneath the bubble becomes more broad and diffuse. The jet can no longer penetrates the bottom surface of the bubble: under such conditions, the bubble takes the shape of spherical cap or skirted bubble.

In addition, the current work presents a robust computational algorithm to handle compressible multiphase flows with strong shocks. The ability of this novel FEM-based strategy using the level set approach and ghost fluid method is demonstrated. Using this method, a 3-D hydrodynamic shock code simulation of an imploding air bubble in a compressible liquid (i.e., water) was done. A transient three-dimensional solution was presented for the implosion and the rebound of the bubble. The results were compared with the Rayleigh-Plesset analysis and found to be in good agreement for most of the implosion process. The deviation during the final stage of collapse and the early part of the rebound was attributed to the aspherical shape of the bubble. The results show the presence of a strong spherical shock inside the gas phase. The spherical asymmetries were captured and the existence of shape and interfacial instabilities were found.

Among different problems which are of interest for single bubble sonoluminescence (SBSL), the stability of the shape is one of the most significant. The stability of the nonlinear oscillations of the bubble are strongly influenced by the bubble shape. In fact, researchers have independently studied the effect of bubble shape perturbations on the ensuing oscillations by perturbing the bubble shape numerically[51]. The work presented here shows that the bubble collapses with multiple harmonics rather than with a single spherical harmonic. This advances the understanding of the bubble dynamics involved during implosions. These results will hopefully encourage more efforts to use DNS as a tool to capture the complex dynamics involved in bubble implosions. Future studies
should use more realistic equations of state and concentrate on the DNS simulations of the forced oscillations of a bubble. This study could be done by imposing the time dependent boundary conditions at the spherical domain boundary. These studies would show the effect of non-spherical shape on the oscillations. Furthermore, one could also perform the parametric studies of the cavitation of bubble clouds to help understand the huge pressure intensifications that have been observed [8] within imploding bubble clouds. Studies of this kind can help determine the key microscopic variables that control the various associated phenomena [64].

There is a lot of scope to extend this thesis work. In the present work mass transfer was not considered and phase change was ignored. These could be considered for a more comprehensive study of sonoluminescence and/or sonofusion. Future work can also focus on extracting the advantage of FEM by evaluating multiphase flows in complex geometries with an automated adaptive meshing strategy. The simulations presented in Section 8.4 would certainly benefit from additional resolution with adaptive remeshing to resolve the merging interfaces more efficiently. Importantly, since the algorithm is implemented in general way, it can be applied to numerous multiphase problems.

For example, flows with moving interfaces are of great research interest. Free surface flows are encountered widely not only in nature, but also in various industrial and environmental applications: mold filling, flows past surface ships and free jets are just a few examples. In many of these flows, the shape of the surface or interface is important for the whole behavior of the physical system. The shape of the interface can be very complex and thus difficult to determine numerically. Development of efficient and reliable adaptive finite element techniques for the solution of quite general transient three-dimensional free surface flow problems is an active potential research area. The dynamics of these systems are determined by capillarity forces, surface stresses, and body forces. Numerical investigation of these flows can give an insight into the complex interplay of various phenomena.

Another interesting direction to extend this work would be in the area of surface tension driven flows. Surface tension is important for fluid dynamics on small scales. The classic example is flow creeping up a thin capillary tube. For micro-fluidic devices (networks of channels, valves and mixers), surface tension effects may be exploited to
move the fluid into desired channels. Numerical simulations can play a great role in the optimal design of channels dominated by surface tension. Through the simulations, one can find the flow profiles that minimize the potential energy due to surface tension.

Furthermore, another research area that could be built upon the current work is study of the effect of Marangoni forces. The phenomena of “swimming bubbles” and the wetting and spreading of a water droplet on solid hydrophobic surfaces is frequently encountered in many industrial applications, such as coatings, printing and painting. Surfactants play an important role in enhancing a liquid’s ability to wet and spread on solid surfaces. It has been known for many years that surface tension gradients and surface wetting, which are controlled by molecular forces, have a marked effect on this phenomenon, but its mechanism is still not well understood. The classical physical concepts for the spreading of liquids on a solid surface can be traced back to two basic mechanisms. The first mechanism is influenced by a disparity in the surface or interfacial tensions at the three-phase contact line. Another mechanism of droplet spreading is due to the Marangoni effect. It has long been known that the spatial variation in surface tension at a liquid/gas surface results in added tangential stresses at the surface. Through the numerical analysis, the effect of Marangoni forces on the spreading can be quantitatively determined.

In addition, a broad range of scientific and engineering problems involve multiple scales. Traditional monoscale approaches have proven to be inadequate, even with the largest supercomputers, because of the range of scales and the prohibitively large number of variables involved. Thus, there is a growing need to develop systematic modeling and simulation approaches for multiscale problems.


