

# **Developing a 3D incompressible vof based multiphase solver using OpenFOAM libraries**

By  
Sarang Abhay Holay  
Roll No. ME14MTECH11005  
M.Tech II Yr.-Fluids and Energy Systems

A Thesis Submitted  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Technology

**Department Of Mechanical Engineering  
Indian Institute Of Technology Hyderabad**



JULY 2016

## Declaration

I declare that this written submission represents my ideas in my own words, and where ideas and words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.



-----  
(Signature)

-----  
Sarang Abhay Holay

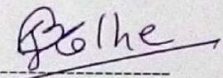
-----  
me14mtech11005

## Approval Sheet

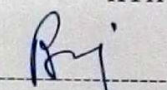
This thesis entitled "Developing a 3D incompressible vof based multiphase solver using OpenFOAM libraries " by Sarang Abhay Holay is approved for the degree of Master of Technology from IIT Hyderabad.



Dr. Saptarishi Majumdar  
Dept. of Chemical Engg.  
IITH



Dr. Pankaj Kolhe  
Dept. of Mechanical Engg.  
IITH



Dr. Raja Banerjee  
Dept. of Mechanical Engg.  
IITH

## Acknowledgements

This thesis would't have been possible without assistance and guidance from a lot of people. I take this opportunity to express my sincerest gratitude to all of them. First, I express my gratitude to my mentor and my guide Dr. Raja Banerjee for propelling me in the right direction every-time i got stuck up. His approach to find a solution to any problem be it technical or non-technical, has helped me evolve.

Next I would like to thank my committee members Dr. Saptarshi Majumdar, Dr. Harish Dixit and Dr. Pankaj Kolhe for their constructive comments and ideas that kept me motivated to see this project through. I also thank our lab assistant and system administrator, Mr. Madhu P. Joshoua , for helping me out in any technical issue in any way he can.

Next i would like to thank my fellow researchers Dr. Rajesh Reddy, Vatsalya Sharma, Nikhil Kalkote and Ashwani Assam for assisting me and guiding me in the fields of CFD with their experience and expertise. Last but not the least, I would like to express my deepest gratitude to my friends at IIT Hyderabad without whom this journey would have been very difficult if not impossible.

Finally, I thank God for this opportunity to work at one of the most renowned institutions of India.



*To my family ...*

# Abstract

In the recent past open source analysis software have been on the demand for studying various phenomenon related to fluids and solids. To counter act the booming prices of commercial CFD and FEM toolkits, open source software provide the key. In the field of computational fluid dynamics, open source cfd toolkit , OpenFOAM (Open source Field Operation And Manipulation) is a viable alternative to commercial cfd toolkits like FLUENT, StarCCM+, etc.

In the following thesis, an attempt has been made to address standard droplet breakup problem using OpenFOAM. The thesis aims at developing an incompressible multiphase vof (volume of fluid) based solver which works on a SMAC algorithm, using the native OpenFoam libraries.

The native vof solver of OpenFOAM and the newly developed explicit solver are tested against standard benchmark cases like 3D lid driven cavity, vortex in a box, dual vortex stretching, spurious currents, square bubble to circle, cubical bubble to sphere, dual vortex stretching and droplet splashing and compared in terms of accuracy and computational time. This knowledge led to deciding upon optimum case setup parameters which was then used to address the problem at hand: droplet breakup.

For the droplet breakup analysis, two droplets with different diameters ratios were allowed to collide for different Reynolds number and Weber numbers. The collisions were also studied for different offsets parameters.

Apart from above standard cases, other important features of OpenFOAM such as different linear solvers, numerical schemes, multicore operational capabilities, dynamic mesh refinement and static mesh refinement were also tested and reported.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Approval Sheet</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Why the interest in droplet dynamics ? . . . . .	1
1.2 Approach to investigation . . . . .	3
1.2.1 Analytical: . . . . .	3
1.2.2 Experimental: . . . . .	5
1.2.3 Numerical: . . . . .	6
1.3 Why OpenFOAM? . . . . .	9
1.4 Scope of this thesis: . . . . .	10
<b>2 Mathematical Formulation</b>	<b>11</b>
2.1 Why SMAC (explicit) algorithm? . . . . .	11

---

2.2	The finite volume method: . . . . .	12
2.3	The Governing Equations: . . . . .	13
2.4	Finite volume discretization: . . . . .	13
2.5	The SMAC algorithm: . . . . .	15
<b>3</b>	<b>Validations and Test Cases</b>	<b>18</b>
3.1	Testing the Navier Stoke's equation solving capabilities of the solver:	19
3.2	Testing the surface tension model implementation of the solver: .	21
3.2.1	3D: Cube to sphere test: . . . . .	22
3.2.2	2D: Square to circle test . . . . .	24
3.3	Static droplet test: a measure of Spurious currents . . . . .	27
3.4	Testing vof model: . . . . .	28
3.5	Testing NS, vof and ST modelling combined: . . . . .	34
3.5.1	2D bubble rise . . . . .	34
3.5.2	Droplet splashing . . . . .	36
3.6	Closure . . . . .	38
<b>4</b>	<b>Droplet Breakup</b>	<b>41</b>
4.1	Parameters affecting droplet collision: . . . . .	43
4.2	Case setup parameters: . . . . .	44
4.3	Results for different offset ratios and diameter ratio = 1.5 . . . . .	45
4.3.1	Offset ratio $x = 0$ . . . . .	45
4.3.2	Offset ratio $x = 0.25$ . . . . .	47
4.3.3	Offset ratio $x = 0.5$ . . . . .	48
4.3.4	Offset ratio $x = 0.75$ . . . . .	48
4.4	Results for different diameter ratios and offset ratio = 0.5 . . . . .	50
4.4.1	Diameter ratio = 1 . . . . .	50

---

4.4.2	Diameter Ratio = 1.25 . . . . .	51
4.4.3	Diameter ratio = 1.75 . . . . .	52
<b>A</b>	<b>Exploring OpenFOAM</b>	<b>55</b>
A.1	Smoothers, solvers and preconditioners: . . . . .	55
A.1.1	Pre-conditioners: . . . . .	56
A.1.2	Solvers: . . . . .	57
A.1.3	Smoothers: . . . . .	58
A.2	Multicore speedup analysis: . . . . .	60
A.3	Dynamic mesh refinement: . . . . .	61

# List of Figures

1.1	Stability theory breakup model . . . . .	4
3.1	Contours of $U_{mag}$ for Re 100 and Re 1000 . . . . .	20
3.2	Steady state plots for Re100 . . . . .	21
3.3	Steady state plots for Re 1000 . . . . .	21
3.4	Grid convergence study for Re1000 . . . . .	22
3.5	Surface tension test . . . . .	23
3.6	Errors using implicit solver : interFoam . . . . .	24
3.7	Bubble centroid vs time for different grids: interFoam . . . . .	25
3.8	Errors using explicit solver : rbsFoam . . . . .	26
3.9	Bubble centroid vs time for different grids: rbsFoam . . . . .	27
3.10	Streamlines at $t = 0$ . . . . .	29
3.11	Droplet shapes for different grids at times 1s and 2s . . . . .	31
3.12	Droplet shapes for different grids at times 4s and 8s . . . . .	32
3.13	Mass conservation for 8 cycles . . . . .	33
3.14	Mass conservation for 8 cycles . . . . .	34
3.15	2D bubble rise results for density ratio = 10 . . . . .	35
3.16	2D bubble rise results for density ratio = 1000 . . . . .	36
3.17	Results for droplet splashing - A . . . . .	38
3.18	Results for droplet splashing - B . . . . .	39



3.19	Results for droplet splashing - B . . . . .	40
4.1	General Layout . . . . .	45
4.2	Contour plots for $x=0$ . . . . .	46
4.3	Contour plots for $x=0.25$ . . . . .	47
4.4	Contour plot for $x=0.5$ . . . . .	48
4.5	Contour plot for $x=0.75$ : A . . . . .	49
4.6	Contour plot for $x=0.75$ : B . . . . .	50
4.7	Contour plot for $x=0.75$ : C . . . . .	51
4.8	Contour plot for diameter ratio = 1 . . . . .	52
4.9	Contour plot for diameter ratio = 1.25 . . . . .	53
4.10	Contour plot for diameter ratio = 1.25 . . . . .	53
4.11	Contour plot for diameter ratio = 1.25 . . . . .	54
4.12	Contour plot for diameter ratio = 1.75 . . . . .	54
A.1	Sample fvScheme dict . . . . .	56
A.2	Sample GAMG settings . . . . .	59
A.3	Different settings for <i>fvSchemes</i> file . . . . .	60
A.4	Time consumption for 0.1 Million mesh . . . . .	61
A.5	Time consumption for 1 Million mesh . . . . .	62
A.6	Comparing decomposition techniques: scotch vs. simple . . . . .	63
A.7	Comparing decomposition techniques: scotch vs. simple . . . . .	64
A.8	Terminal Bubble positions for different levels of mesh refinement dr 1000 . . . . .	65
A.9	Bubble centroid vs time for density ratio 10 . . . . .	66
A.10	Terminal Bubble positions for different levels of mesh refinement dr 1000 . . . . .	66

---

A.11 Bubble centroid vs time for density ratio 1000 . . . . .	67
A.12 Bubble centroid vs time for density ratio 1000 . . . . .	67
A.13 Bubble centroid vs time for density ratio 1000 . . . . .	68
A.14 Bubble centroid vs time for density ratio 1000 . . . . .	68

# List of Tables

1.1	Comparison of various interface capturing methods . . . . .	9
3.1	Surface area of sphere and corresponding error . . . . .	23
3.2	Surface area of sphere and corresponding error: rbsFoam . . . . .	26
3.3	Surface tension test results: spurious currents . . . . .	28
3.4	Error for T=2 sec using interFoam and rbsFoam . . . . .	30
3.5	Error for T=8 sec using interFoam and rbsFoam . . . . .	30
3.6	Error for dual vortex stretching test . . . . .	33
3.7	Case parameters for 2D bubble rise . . . . .	34
3.8	Clocktime comparision: explicit vs implicit solvers for density ratio 1000 . . . . .	37
4.1	Physical Properties . . . . .	44

# Chapter 1

## Introduction

### 1.1 Why the interest in droplet dynamics ?

Combustion is defined as a process of conversion of chemical energy of fuels into heat, light and sound. Combustion phenomenon in IC engines, gas turbines or any other prime mover that involves petroleum fuels, occurs in gaseous phase. No liquid fuel burns in its original state. Hence, atomization of liquid fuels is the primary step in any combustion process. Looking at IC engines as an area of interest, the injected fuel has very low residence time in combustion chamber. In this small time interval, the injected fuel, which is in liquid state, has to be broken down into smaller droplets which then evaporate to form a near homogeneous gaseous charge ready for combustion. Since evaporation is a surface phenomenon, larger the surface area to volume ratio, faster is the evaporation. As, the droplet size decreases, the surface area to volume ratio keeps on increasing. Thus, smaller the droplet, faster is its atomization implying better and complete combustion. The characteristics of spray formed significantly affects the engine performance. The local equivalence ratio of the fuel air mixture determines the kind of combustion that will occur in the chamber which in-turn depends upon the mass fraction of liquid and gas present in any given sample volume. Thus, all in all, formation of the charge for combustion in an IC engine is a complex multiphase phenomenon that involves droplet liquid sheet breakup, droplet atomization and breakup, droplet interaction with surrounding like and unlike droplets involving collision and child droplet formation, and finally its combustion. This makes droplet studies imperative for better understanding and improvement of IC en-

gine combustion. Similar arguments can also be made about physics behind gas turbine combustion and liquid jet propulsion systems. On top of all, with the introduction of newest fuel injection technologies like CRDI (Common Rail Direct Injection), MPFI (Manifold Port Fuel Injection) and GDI (Gasoline Direct Injection) it becomes imperative to know all that there is to be known about droplet and bubble dynamics. Also, with the reserves of fossil fuels hitting an all-time low, desperate attempts are being made to find alternatives to fossil fuels especially for locomotive purposes. Fuel blending is a promising step in this direction. So, to analyze the effect of different blends on droplet and jet breakup these studies are a must. Hence, it becomes imperative to study droplet interaction and their breakup at length.

Droplets have been an active area of research for decades. And still it has been very difficult to successfully address the actual physics of the phenomenon. Various attempts have been made over the years to explore droplet and bubble dynamics and a lot still remain unknown. The first studies on droplet collision, using water droplets in air at atmospheric pressure, have been conducted because of meteorological interest by Adam et al. [2]. They focused attention on the aerodynamic environment of the event and on the outcome of the collisions. Park [3] produced collisions between streams of water droplets traveling in still air and showed pictorially that near head-on collision between pairs of equally sized droplets of 700  $\mu\text{m}$ , resulted in stable coalescence, while off-centre collision at the same relative velocity resulted in a transient coalescence and finally in separation. As reported by Qian and Law [8] for water droplets, for head-on collisions at atmospheric pressure bounce is not observed; for the same conditions however, the collision outcome between hydrocarbon droplets may result to bouncing. Jiang et al. [9] provided a comprehensive quantitative assessment of droplet collisions of hydrocarbon droplets (heptane, decane, dodecane, tetradecane and hexadecane), and later Qian and Law [8] extended these investigations to include the effects of ambient pressure, density, viscosity and impact parameter (characterizing off-centre binary collisions). The following collision regimes were found with increasing Weber number; droplet bouncing, stable droplet coalescence, unstable droplet coalescence and droplet stretching separation, [8],[9]. Estrade et al. [10] published information about the number of satellite droplets, their sizes and velocities produced by bouncing collisions. Estrade et al. [10] published information about the number of satellite droplets, their sizes and velocities produced

by bouncing collisions. Moreover, they also developed a model for predicting the boundary between the bouncing and the coalescence regimes. Brazier-Smith et al. [4] carried out experiments on binary water droplet collisions and developed the threshold of the stability of water droplets against separation, while Arkhipov et al. [12] obtained a relation for the impact parameter separating stable coalescence from stretching separation. Willis and Orme [13] conducted experiments of droplet collisions in a vacuum, devoid of aerodynamic effects, focusing on the role of viscosity in the evolution of the collision phenomenon. Experimental results of Ashgriz and Poo [5], showed that for the same Weber number of the two colliding droplets, the number of satellite droplets resulting from the droplet separation increases with the increase of the impact parameter.

## 1.2 Approach to investigation

Now that the need for this study has been established the immediate question that needs to be answered is the kind of approach that one needs adopt for the study.

### 1.2.1 Analytical:

The analytical approach to droplet breakup proposes two theories linear stability theory and non-linear stability theory. To understand the analytical approach, the actual theoretical mechanism of droplet formation needs to be understood at length. Usually, droplets are formed from parent mass of fluid through ligament separation and pinching. As the parent mass of fluid deforms, a smaller mass of fluid starts separating from it. Initially, they remain connected by a small ligament of fluid. As the ligament stretches, its surface area keeps on increasing. If more surface area is present than the minimum required to contain the volume of fluid, the system has excess of surface energy. A system not at minimum energy state will attempt to rearrange so as to move toward the lower energy state leading to breakup of fluid into smaller masses to minimize the system surface energy by reducing surface area.

To analytically determine droplet breakup, linear stability theory was proposed. The theory states that the breakup process starts with the development of certain perturbations on the free liquid surface of the fluid. These disturbances are



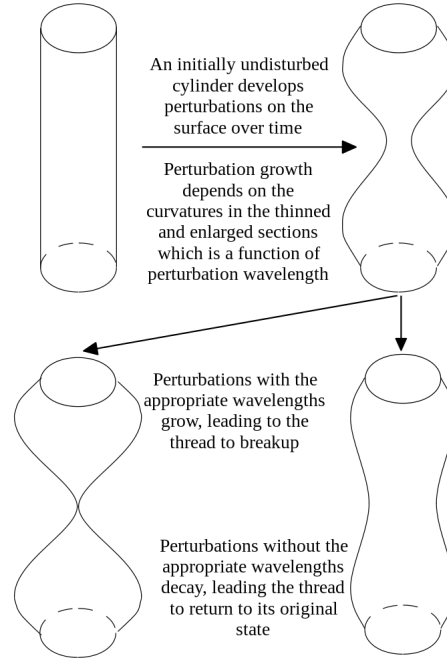


Figure 1.1: Stability theory breakup model

very arbitrary in nature. Hence they are analyzed using fourier transform which converts the arbitrary disturbances into perturbations of a single wavelength. Consider a cylinder as shown in figure 1.1. The perturbations cause a change in radius at different locations. At the region of less curvature, the pressure is high which forces the liquid from this region into a that of high curvature since the pressure in this region is low. However, the radius of curvature in the larger region is very important to breakup process. For some perturbation wavelengths, the pressure in the larger region may surge overcoming the effect of pressure in the smaller region and thus forcing the fluid back thus returning the fluid to its original undeformed shape. In other cases, the perturbation wavelength assists the pressure in the smaller region leading to complete flushing of liquid out of the smaller region finally resulting into pinching.

While linear theory is useful in considering the growth of small disturbances on the free surface, when the disturbances grow to have a significant amplitude, non-linear effects begin to dominate breakup behavior. The non-linear behavior of the thread governs its final breakup and ultimately determines the final shape and number of the resulting fluid masses.

The relationships obtained using these stability theories give empirical relationships involving the wave numbers. But the proportionality constants need to be evaluated using experimental data. Hence, to conclude, the stability theory can only predict which wavelength will be dominant and responsible for droplet breakup. But, it cannot address the transient nature of the process nor the statistics of the outcome like child droplet diameter or their number. Hence, experimental techniques were devised.

### 1.2.2 Experimental:

Since decades, scientists and researchers all over the world have been using various experimental techniques to address multiphase phenomenon. Experimental studies of multiphase flows are significantly more difficult than single phase flows. For sprays in atomization regime, length scales and time scales are short and optical access to primary atomization zone is limited. The primary atomization step is most difficult to address. Hence, most of the earlier experimental studies were more focused on characterizing sprays by analyzing the spray cone angle, tip penetration length, breakup length, mean droplet diameter etc. Over the years the experimental techniques have become more reliable, more accurate, more sophisticated but also more expensive. These techniques include: Laser Doppler Anemometry (LDA), PDPA (Phase Doppler Particle Analyzer), PIV (Particle Image Velocimetry), LIF (Laser Induced Fluorescence) [16], Mie scattering experiment [17], Schlieren photography [18], shadowgraphy [19], Laser Rayleigh Scattering (LRS) [25]. In general, all these experiments require a source of light that will illuminate the region under observation and a high speed camera that will capture the phenomenon for further post processing. Both these components are extremely expensive.

Although any experimental technique in multiphase flows is absolute, i.e. it doesn't require any separate validation, it has some inherent drawbacks. Using experimentation, the flow can be analyzed only in the regions where there is sufficient concentration of the phase of interest. Once the phase becomes too dispersed, it becomes lesser and lesser visible to the apparatus. And hence cannot be further investigated. Also, on the other hand, if the region is very dense, visible light is scattered by dense region causing large errors in measurements. Thus, experimental techniques are inadequate at regions very close to the nozzle (in

case of spray problems) where the phase is very dense and also far away from the nozzle where the phase is very disperse. Also, for a real automobile injector, it is very difficult to capture the high speed sprays at sufficient spatial and temporal resolution. Studies on flow visualization in nozzles are also available in literature. Kim et al. [20] performed visualizations of the internal flow in diesel injection nozzles by using transparent model nozzles that are ten times larger than real diesel injectors. Tamaki et al. [21] conducted an experimental work on water jets produced by transparent nozzles of different geometrical characteristics. They used micro flash spark light for internal flow visualization. Detailed review of the status of experimental investigation of primary atomization of liquid streams is reported in [22].

### 1.2.3 Numerical:

The reason behind the choice of experimentation in earlier decades was that computers of that era were not powerful enough to handle the hefty computations needed to capture multiphase flow phenomenon. But now with the advent of faster and powerful computing resources, numerical simulations have gained a lot of popularity as well as reliability. With the introduction of parallel computing, a lot of impossible tasks of yesterday have become a common day reality of today. Also, with the recent debut of GPU based parallelization the world of parallel computing has reached its peak. Numerical investigations have their own advantages when weighted against experimental techniques. Significant reduction in cost and space requirement is one of them. Also with better numerical models developed day by day the reliability of these simulations is always on the rise. Hence, a numerical approach is adopted in this thesis for the droplet study.

Any general multiphase flow solver needs two components. First, an algorithm to solve the Navier Stokes equation. These include SMAC, PISO, SIMPLE or their derivatives. And second, an algorithm to deal with the different phases that come into the picture.

Two principle approaches to the numerical study are Eulerian-Eulerian and Eulerian-Lagrangian methodology. In the Eulerian-Eulerian approach both liquid and gas phases are solved on a fixed grid. Eulerian-Eulerian approach for spray modeling is a recent addition in this category and has made significant progress in the

past decade [24],[25],[26]. Locating and accurate capturing of the continuously deforming interface is the most challenging task faced by such an algorithm. In Eulerian-Lagrangian methodology, for a droplet study, the gaseous phase is solved on a fixed grid while the liquid droplets are given Lagrangian treatment. For a bubble study, the roles are reversed.

The multiphase flow simulations in Eulerian-Eulerian framework can proceed via two approaches: Multi fluid approach and One-fluid approach.

In the multi-fluid approach, the complete computational domain is divided into sub-domains. All the governing equations are valid in these sub-domains and all these sub-domains are connected to their neighbors numerically by appropriate boundary conditions. This usually required a computationally expensive iterative algorithm. When two fluid model is used a number of interfacial transport coefficients are defined and require constitutive relation models to complete the overall model [26]. This approach has an advantage that the actual transport processes can be rigorously defined, however, the disadvantage is that one is required to model these kinetic processes in detail, which implies a much greater depth of experimental data and insight.

On the other hand, in a single fluid approach, one set of governing equations are solved over the entire domain. The velocity field is assumed to be continuous and the boundary conditions are implicitly contained within the equations of motion. The physical properties of the fluid in a particular cell are computed from the physical properties of the individual phases with the help of an appropriately defined weighted function from the location of the interface. That indicates a need for interface tracking/capturing algorithm. Detailed classification of interfacial flow modelling techniques is given in [27]. The interfacial solvers have two main categories: one involves solution on moving and deforming meshes in which the grid points move according to local flow characteristics while the other category involves solution on fixed meshes. In the moving mesh methods, the grid points move according to local flow characteristics. Moving mesh category includes methods such as finite element based Lagrangian [28] interface fitting (or boundary fitted coordinate) [29] and boundary integral (more popular as boundary element methods) [30].

Fixed grid methods are very useful while dealing with large deformations of interfaces. In these methods an extra indicator function is used to track the interface. Based on the type of indicator function used, these methods can be classified

as: interface capturing and interface tracking methods. Although there is a common practice in literature to use these methods interchangeably there is a difference between these two methods [27]. Interface capturing method represents position of interface in an indirect manner. Whereas, interface tracking method provides a direct description of interface usually, by specifying a set of marker points located on the free surface [31]. Because of Lagrangian approach, interface tracking method is very accurate. But, unlike interface tracking method, a great advantage with interface capturing method is their inherent ability to handle interface merging and breakup. The most widely used interface tracking method is Glimms Front Tracking method [32]. While, the most widely used interface capturing methods include Marker And Cell (MAC) [33], volume of fluid (vof) [34],[35], level set [36],[37] and diffuse interface (or phase field) [38],[39] method. Also, there are some hybrid methods intended to harness the advantages of both interface tracking as well as interface capturing methods. These include Tryggvasons front tracking [40] and sharp interface methods [41]. Fuster et al. [44] have reviewed the current advances in multiphase flows using VOF method. According to them, an ideal multiphase flow solver should have the following properties:

- Robust representation of evolving topologically complex interfaces
- Accurate representation of surface tension, which requires accurate estimates of interface normal and curvature
- Robust and accurate handling of large density and viscosity ratios
- Efficient representation of evolving flow features of widely different characteristic spatial scales.

It is very difficult to find all these features in a single solver. A generic comparison of all interface tracking methods is given in table 1.1. As mentioned before, in fixed grid methods an indicator function is required to track the interface. Hence, in the volume of fluid method this tracker is the volume fraction. The cells which completely filled with a particular fluid are assigned a value of 0 or 1. While the cells which are partially filled, i.e. the cells which have the interface, have a value between 0 and 1. The interface is then approximated from this volume fraction value. Various reconstruction methods are available and are

Method	Advantages	Disadvantages
Volume of fluid	Good mass conservation	Accurate curvature estimation techniques are required.
	Maintains sharp interface	
	Inherently handles droplet merging and breakup	
Level-set	Relatively easy to extend to 3D	Poor mass conservation
	Inherently handles droplet merging and breakup	Reinitialization of level set function is difficult
Front tracking	Extremely accurate	Difficult to handle merging and breakup of interface
		Computationally expensive
Phase field	Can handle droplet merging and breakup	Interface is diffused over multiple cells
Marker particle	Extremely accurate	Computationally expensive
	Can handle droplet merging and breakup	Redistribution of marker particles is required

Table 1.1: Comparison of various interface capturing methods

reported in brief. The oldest one is the Simple Line Interface Calculation (SLIC) [34] where the interface in each cell was assumed to be either horizontal or vertical. Later came the Piecewise Linear Interface calculation (PLIC) [45],[46],[47] method which had improved interface reconstruction capabilities because the interface in any cell was represented by the general equation of line (in 2D) or plane (in 3D). The result of the marker function advection strongly depends on interface reconstruction [79] The accurate estimation of curvature is very important because it also plays an important role in determining the surface tension force incorporating the continuous surface force model.

Because of its inherent volume conserving property, a vof model is a lucrative choice to model multiphase flow. OpenFOAMs multiphase solver also uses vof model. Also, the research group responsible for this thesis has a lot of experience dealing with vof model based approach to spray characterization and droplet breakup. Hence, vof model was an obvious choice for this thesis.

### 1.3 Why OpenFOAM?

With the significant amount of time and resources spent in the development of numerical methods Computational Fluid Dynamics and gained a lot of popularity and respect. With that, a lot of commercial and open source CFD toolkits have emerged over the last decade. These include: Ansys-Fluent, StarCCM+, Gerris, OpenFOAM, EnSight; to name a few. All the commercial cfd toolkits come at a significant price. Hence, in the modern days, open source tools like OpenFOAM, Gerris, etc. have become very popular. Another important factor driving the drift towards open source softwares is that the commercial softwares are a black-



box. Apart from the provision to write a user-defined function, the user cannot touch the source code of the software. Hence, some specific tailor made operations are difficult to perform using a commercial code. On the other hand, an open source code provides all kinds of freedom to the user to edit and modify and improvise upon any domain he deems fit. OpenFOAM, because of its versatility, has become very popular in recent times. A lot of multiphase studies have been successfully performed using OpenFOAM

## 1.4 Scope of this thesis:

In most of the numerical studies involving spray and atomization, the interaction of the liquid with its surrounding atmosphere is well addressed. But, the effect of other surrounding droplets has not been taken into account. During liquid sheet breakup, spray formation and atomization, the droplets may interact with each other resulting into stable or unstable coalescence, pinching and child droplet formation. So, this thesis is an attempt to have a closer look at droplet interaction. Also, a common decision was made within the group to use OpenFOAM cfd toolkit for this purpose. The main aim of this thesis, is to judge OpenFOAM's multiphase solver on different grounds and compare the results with standard benchmark cases. With these results in hand, a droplet collision study is presented. Also, the research group further intends to modify the vof model used in OpenFOAM and implement a coupled level set and vof methodology. (CLSVOF) Hence, this thesis will also serve as benchmark to comment upon the improvements in the new model.

# Chapter 2

## Mathematical Formulation

This chapter describes in brief the mathematical formulation of Navier Stokes equations. The current code is based upon the SMAC (Simplified Marker And Cell algorithm). The same will be discussed at length below. Also the code is based upon finite volume method. Hence the discretization will be discussed based upon this method.

### 2.1 Why SMAC (explicit) algorithm?

Its an undisputed fact that implicit algorithms are numerically very stable as compared to explicit. Yet, this thesis emphasizes on the use of an explicit solver working on SMAC algorithm. The reasons behind this venture are explained below: Explicit solvers can be used when we have very low CFL number (Courant Fredric and Lewy).The CFL number is given by:

$$CFL = \frac{U\Delta t}{\Delta x}$$

Where:  $\Delta t$  = time step and  $\Delta x$  = grid size and  $U$  = velocity in the cell under consideration. The CFL condition for applicability of explicit solver states that for the solution to converge  $CFL \ll 1$

The newly developed solver is supposed to address droplet breakup problem. The non-dimensional numbers that are usually considered while addressing this phenomenon are Weber number and Reynolds number. Weber number is defined as the ratio of aerodynamic forces to viscous forces.

$$We = \frac{\rho U^2 L}{\mu}$$

Droplet breakup is a high Weber number phenomenon. Hence, large inertial forces come into play implying large velocities. Hence, to capture the physics of the flow properly, very small time steps need to be taken ( $\Delta t$  is of the order of  $10^{-9}$ ). Also, since there is droplet breakup, we need very fine grid to capture the child droplets. That sets  $\Delta x \ll 1$ . Usually  $\Delta x$  is of the order of  $10^{-7}$ . Comparing the orders of time and spatial resolution concludes that CFL number for a droplet breakup problem is well below 0.1. Hence, the use of explicit solver is justified. Also, explicit solvers are faster as compared to implicit solvers. Hence, a lot of computational efforts can be saved by using the explicit solver. So, an explicit incompressible multiphase solver based on vof method was build using OpenFoam libraries.

## 2.2 The finite volume method:

In general, there are three methods for discretizing any given governing equation. They are: finite difference method (FDM), finite element method (FEM) and finite volume method (FVM). Of these, FDM utilizes the equation in their strong form or differential form while FEM and FVM utilizes the equations in their weak form or integral form. The compiled vof solver utilizes OpenFOAM libraries that involve discretization using the finite volume method. Most of the commercial codes use FVM as a preferred choice of discretization. The reasons are as follows: [54], [55]

- It is easier to implement
- It provides a more natural treatment of Neumann boundary conditions as well as that of discontinuous source terms due to their reduced requirements on the regularity or smoothness of the solution.
- It is better suited to deal with complex geometries in multidimensional problem as the integral formulations do no rely on any special mesh structure.

## 2.3 The Governing Equations:

Any general multiphase incompressible flow problem has following principle governing equations:

**The Navier Stoke's Equation:**

$$\frac{\partial \rho U}{\partial t} + U \cdot \nabla U = -\nabla p + \nabla \cdot (\mu \nabla U) \quad (2.1)$$

In equation 2.1 ,  $\rho$  is the density of fluid,  $U$  is the velocity field,  $t$  is time,  $p$  is pressure field,  $\mu$  is the dynamic viscosity of fluid.

**The continuity equation:**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \quad (2.2)$$

If the flow is incompressible the equation 2.2 translates to

$$\nabla \cdot U = 0$$

**Scalar transport equation:**

$$\frac{\partial \alpha}{\partial t} + U \cdot \nabla \alpha = 0 \quad (2.3)$$

Here,  $\alpha$  is the volume fraction. As mentioned before, this equation serves as the marker function used to capture the interface.

## 2.4 Finite volume discretization:

The equations 2.1 2.2 and 2.3 are in their strong form (differentiable form) . The finite volume method of discretization needs the equations to be in their weak form (integral form). To convert the equations from their strong form to weak form, finite volume integration is performed over a given control volume.

Mathematically,

$$\iiint_V \frac{\partial \rho U}{\partial t} + U \cdot \nabla (\rho U) dV = \iiint_V -\nabla p + \nabla \cdot (\mu \nabla U) dV$$

As the flow is incompressible, density can be treated as constant. Hence, the final equation that needs to be discretized is:

$$\iiint_V \frac{\partial U}{\partial t} + U \cdot \nabla U dV = \iiint_V \frac{-1}{\rho} \nabla p + \nabla \cdot (\nu \nabla U) dV \quad (2.4)$$

where  $\nu = \mu/\rho$  is the kinematic viscosity of the fluid.

For ease of understanding, equation 2.4 can be split into four terms based on their contribution to flow:

1. Temporal term
2. Convective term
3. Diffusion term
4. Pressure term

The nomenclature used for the discretization is as follows: capital letters (N,E,W,S) stand for cell center values while, small letters (n,e,w,s) stand for face center values. P stands for current cell under consideration while all other cells labeled N,E,W,S (north, east, west and south respectively) are labeled with respect to this cell. n and n+1 stand for current and next time step respectively. For further understanding the reader is referred to [27]

The terms are discretized as follows:

**Temporal term:**

$$\iiint_V \frac{\partial U}{\partial t} dV = V_P \frac{V_P^{n+1} - V_P^n}{\Delta t} \quad (2.5)$$

here  $V_P$  is the volume of cell under consideration. The fundamental assumption here is that the volume of cell remains constant through out the time step  $\Delta t$ . For all the remaining terms **Gauss divergence theorem** has been used to convert volume integral into surface integral. The Gauss divergence theorem can be mathematically written as:

$$\iiint_V \nabla \cdot \phi dV = \iint_A \phi \cdot \bar{n} dV \quad (2.6)$$

In pure physical sense, Gauss Divergence theorem can be thought off as a result that relates flow of a vector field through a surface to the behavior of a vector field inside the surface. Applying this theorem, the finite volume discretization of the remaining terms is explained below:

**Convective term:**

$$\iiint_V \nabla(UU)dV = \sum_f U_f F_f \quad (2.7)$$

here  $U_f$  is velocity at face f under consideration. F can take values from n,e,w,s,t,b (north, west, west, south, top and bottom respectively.)  $F_f$  is the volume flux given by  $U_f.F_f$ .

**Diffusion term:**

$$\iiint_V \nabla.(\mu \nabla U)dV = \sum_f \mu \nabla U_f . dS_f \quad (2.8)$$

$dS_f$  is the area vector normal to any face f. An area vector directed away from the face is assigned positive value.

**Pressure term:**

$$\iiint_V \nabla P dV = \sum_f P_f S_{fn} \quad (2.9)$$

$P_f$  is the value of pressure at face center and  $S_{fn}$  is the surface area vector normal to the face.

## 2.5 The SMAC algorithm:

The final discretized equations for the SMAC algorithm are:

$$V_P \frac{U_p^{n+1} - U_p^n}{\Delta t} + \sum_f U_f^n F_f^n + \sum_f F_{fdu}^n = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fx} \quad (2.10)$$

$$V_P \frac{V_p^{n+1} - V_p^n}{\Delta t} + \sum_f V_f^n F_f^n + \sum_f F_{fdu}^n = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fy} \quad (2.11)$$



$$V_P \frac{W_p^{n+1} - W_p^n}{\Delta t} + \sum_f W_f^n F_f^n + \sum_f F_{fdu}^n = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fz} \quad (2.12)$$

These are the momentum equations along x,y and z directions respectively.

$$\sum F_f = 0 \quad (2.13)$$

This is the continuity equation.

The step by step operation of SMAC algorithm is summarized below:

1. Predictor Step: Obtain predicted velocities (or mass velocities) by dropping off pressure terms.
2. Pressure poission: Imposing continuity using the predicted velocities gives the pressure poission equation.
3. Corrector Step: Using the newly found pressure values in conjunction with continuity, final corrected velocities are obtained.

The detailed procedure is as follows:

**Predictor Step:** The pressure terms on the rhs of equations 2.10, 2.11, 2.12, are dropped to give predicted or mass velocities as shown in equations 2.14, 2.15, 2.16

$$V_P \frac{U_p^* - U_p^n}{\Delta t} + \sum_f U_f^n F_f^n + \sum_f F_{fdu}^n = 0 \quad (2.14)$$

$$V_P \frac{V_p^* - V_p^n}{\Delta t} + \sum_f V_f^n F_f^n + \sum_f F_{fdu}^n = 0 \quad (2.15)$$

$$V_P \frac{W_p^* - W_p^n}{\Delta t} + \sum_f W_f^n F_f^n + \sum_f F_{fdu}^n = 0 \quad (2.16)$$

$U_p^*, V_p^*$  and  $W_p^*$  represent the predicted velocities.

Subtracting equations 2.14 2.15 2.16 from equations 2.10 2.11 2.12 respectively yields:

$$V_P \frac{U_p^{n+1} - U_p^*}{\Delta t} + = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fx} \quad (2.17)$$

$$V_P \frac{V_p^{n+1} - V_p^*}{\Delta t} + = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fy} \quad (2.18)$$

$$V_P \frac{W_p^{n+1} - W_p^*}{\Delta t} + = -\frac{1}{\rho} \sum_f P_f^{n+1} S_{fz} \quad (2.19)$$

Using continuity equation and substituting the values of  $U_P^{n+1}$ ,  $V_P^{n+1}$  and  $W_P^{n+1}$  obtained from equations 2.17, 2.18 and 2.19 the Pressure poission's equation is obtained which is given by:

$$\sum_f (\nabla P_f^{n+1}) \cdot S_f = \frac{\rho}{\Delta t} \sum_f F_f^* \quad (2.20)$$

where  $F_f^* = U_f^* \cdot S_F$  and  $U_f$  is the interpolated value at face center. Note that here  $F_f$  deals with velocities along all 3 principle directions

The solution to equation 2.20 gives the new pressure values which can then be substituted in equations 2.17, 2.18 and 2.19 to yield final velocity field.

Once the velocity field is known, the newly found values can be substituted in the scalar transport equation 2.3. The discretized form of equation 2.3 is as follows:

$$\Delta V_P \frac{\alpha_P^{n+1} - \alpha_P^n}{\Delta t} + \sum_f F_f^n \alpha_f^{n+1} = 0 \quad (2.21)$$

The value of  $\alpha^n$  is already known due to either initialization or from previous iteration. Hence, equation 2.21 can be iteratively solved to get values of  $\alpha^{n+1}$ .

This concludes one time step. For the next time step the  $n + 1$  level values are transferred to  $n^{th}$  level and the algorithm repeats.

# Chapter 3

## Validations and Test Cases

As mentioned previously, the aim of this thesis is to rigorously test vof solver for its capabilities in various domains, and identify areas of improvement. OpenFoam itself, in all its standard releases, has a vof based multiphase solver called interFoam. It works on PIMPLE algorithm. It is a tailor made algorithm aimed at harnessing the benefits of both SIMPLE and PISO algorithm. But it is implicit in nature. Yet, this thesis utilizes an explicit solver. The reasoning behind that has been already done in Chapter 2

Initially, the implicit solver was tested against some standard rigorous tests and later same set of tests were again performed on rbsFoam, the freshly compiled explicit solver.

All the test cases, that were conducted and the reasons behind conducting them are mentioned below. Any multiphase solver is judged based on following domains:

- **Navier Stokes solving capabilities:** These tests estimate how well the Navier Stokes equation is solved. The standard test cases used include 3D lid driven cavity, Flow over a square cylinder.
- **Surface tension modelling:** In any multiphase flow problem, surface tension appears as a source term in the governing equations. Its calculation involves estimation of the curvature. The tests carried out include: 3D cube to bubble test, spurious currents estimation.
- **vof modelling:** The tests carried out to test the solution of the vof equation

include: vortex in a box test, Dual vortex stretching test, etc.

- Finally, one needs to check if all the above modules put together, work well. For that a standard bubble rise test is done in 2D, for both high and low density ratios. To do the same in 3D, a droplet splashing simulation is attempted.

All above test cases will be described at length below. The description shall include the geometry used for the test case, the different meshes used, the time stepping method, the physical properties used and the boundary conditions. The nomenclature of the boundary conditions used below is derived from OpenFoam to facilitate replication and further use.

### **3.1 Testing the Navier Stoke's equation solving capabilities of the solver:**

#### **3D Lid driven cavity:**

A fluid is kept in a cubical domain. The top lid of the domain is moving at a constant velocity. The results are reported for flows with Reynolds number of 100 and 1000. After steady state is achieved the x velocity is plotted against Y center line and the v velocity is plotted against X center line of the domain and the plots are compared with standard experimental results by Ku et al. [48]. This test gives a closer look at the Navier Stoke's solution capabilities of the solver.

Different meshes are used to do a mesh independence study. 10x10x10; 20x20x20 ; 30x30x30; 40x40x40. Here the numbers indicate the number of cells along x, y and z directions respectively as mentioned in the constant/blockMeshDict file in OpenFoam case directory. Dirichlet boundary condition is used on all the faces for velocity. The top wall is assigned a velocity of 1m/s along x direction while all other faces have 0 m/s velocity assigned to them. A zero gradient pressure boundary condition is imposed on all faces except the bottom where the BC is Dirichlet having a value of 0 which is needed to begin the solution iteration. (guess value)

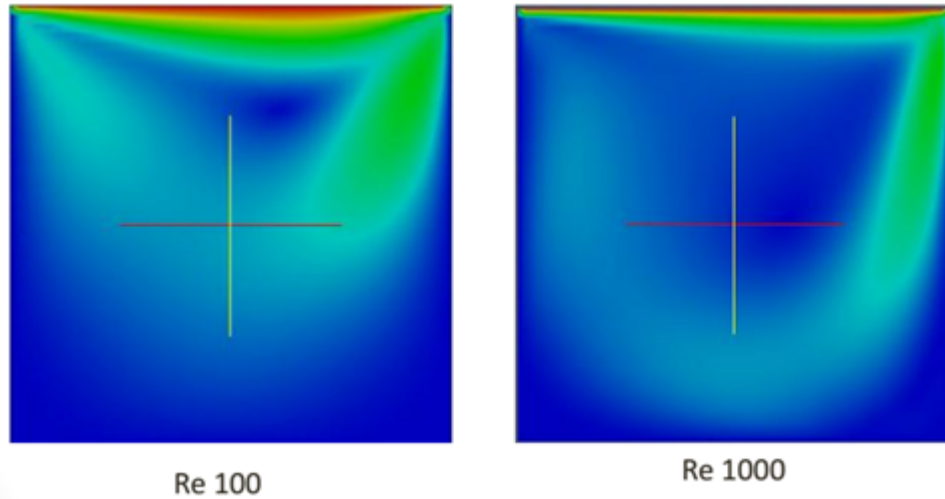


Figure 3.1: Contours of Umag for Re 100 and Re 1000

Discussion:

From the above plots it can be concluded that the results obtained from the simulations are consistent with those obtained from experiments performed by Ku et al. Since a lid driven cavity case is a single phase incompressible flow, it demands the solution to Navier Stokes solution only. So, satisfactory results for this test conclude that the Navier Stokes solution algorithm implemented in rbsFoam is fairly accurate. Secondly, from Figures 3.3 and 3.2, it is seen that the peaks are not captured to sufficient accuracy. This is expected since the flow is very transient in those regions due to recirculation vortices. For all above cases, the time step was 1ms. A finer mesh is needed to capture the vortices accurately to numerical precision. A grid convergence study was also done to check what quality mesh would be good enough to best capture the phenomenon. The results are reported in figure 3.4. The Re 1000 case was chosen for grid convergence study since it offers more adverse test conditions involving strong recirculation zones and vortices.

From above plots, it is observed that a finer mesh provides better results by capturing the velocity profiles in the recirculation zone more accurately. Also when compared against a commercial cfd solver like fluent, the results from rbsFoam for the same test case of re 1000 give a 100 complete match.

### 3.2 Testing the surface tension model implementation of the solver:21

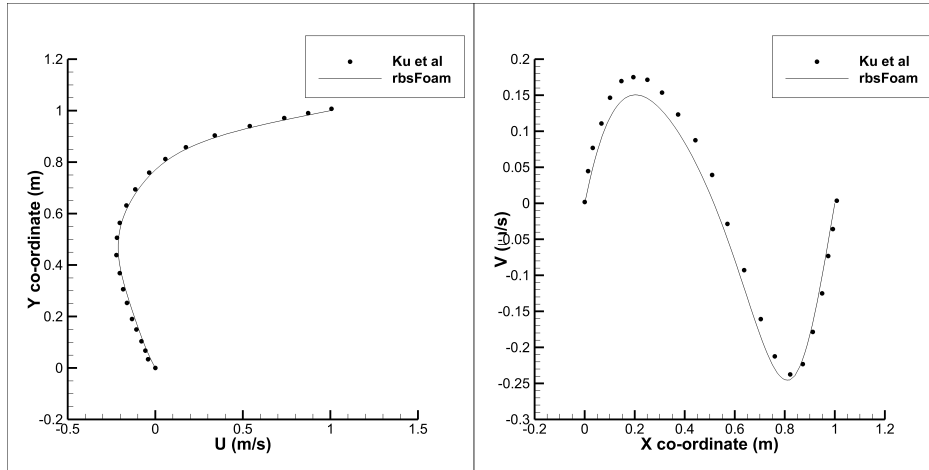


Figure 3.2: Steady state plots for Re100

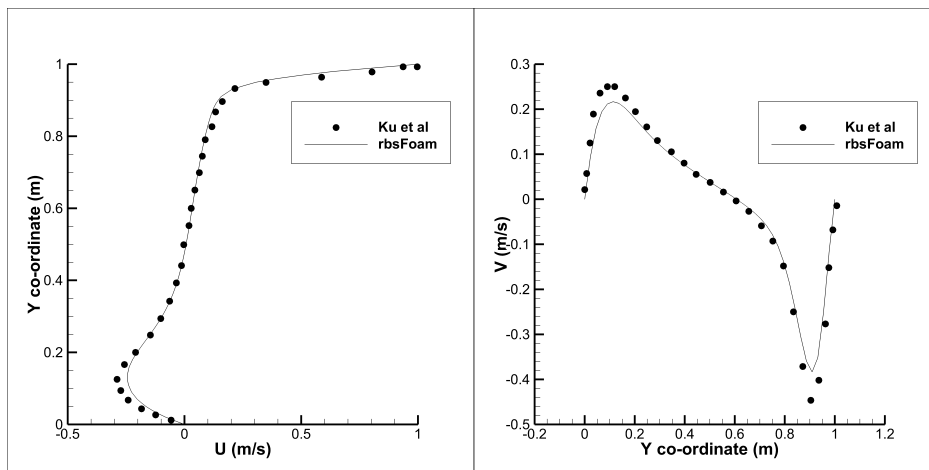


Figure 3.3: Steady state plots for Re 1000

### 3.2 Testing the surface tension model implementation of the solver:

Depending upon the initial configuration, this test can be either 2D or 3D. If the initial configuration is a circle, its a 2D test and if it is a sphere, its a 3D test.

## 223.2 Testing the surface tension model implementation of the solver:

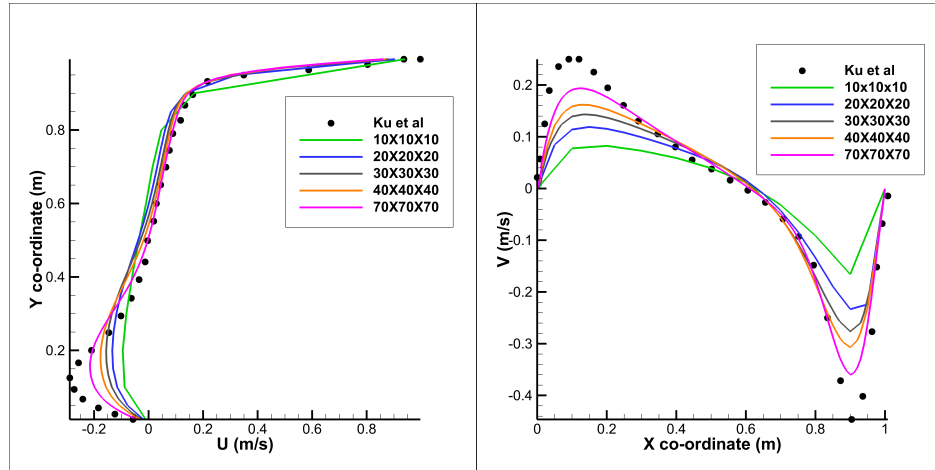


Figure 3.4: Grid convergence study for Re1000

### 3.2.1 3D: Cube to sphere test:

While solving for any multiphase flow problem, surface tension appears as a source term in the Navier Stokes equation. rbsFoam used continuum surface force model for dealing with surface tension. The standard tests for checking the accuracy of the surface tension model include a square or cubical bubble test (in 2D and 3D respectively). The details for the 3D test are mentioned below. The same setup was used for the 2D case.

Due to surface tension, liquids have an inherent tendency to obtain a shape that has the least surface energy associated with it. Hence, any square or cubical liquid drop will have a tendency to take the shape of a circle or sphere respectively. So in this test, a cubical liquid drop is placed in a domain with zero gravity. The simulations are run until a steady state is reached. Once steady state is reached, the circularity of the sphere is tested which also serves as a measure of accuracy of the surface tension model. The surface area of the resultant sphere serves as a mathematical measure for this test

This test was done on both; the explicit solver rbsFoam and the implicit solver interFoam. Interesting results were obtained when implicit as well as explicit multiphase solvers were subjected to this test.

A cubical domain of size 2m is chosen for the test and a bubble of unit diameter is placed at its center. To test the mesh dependency, a 25x25x25, 50x50x50, 75x75x75, 100x100x100, 125x125x125 and 150x150x150 meshes are chosen. The

### 3.2 Testing the surface tension model implementation of the solver:23

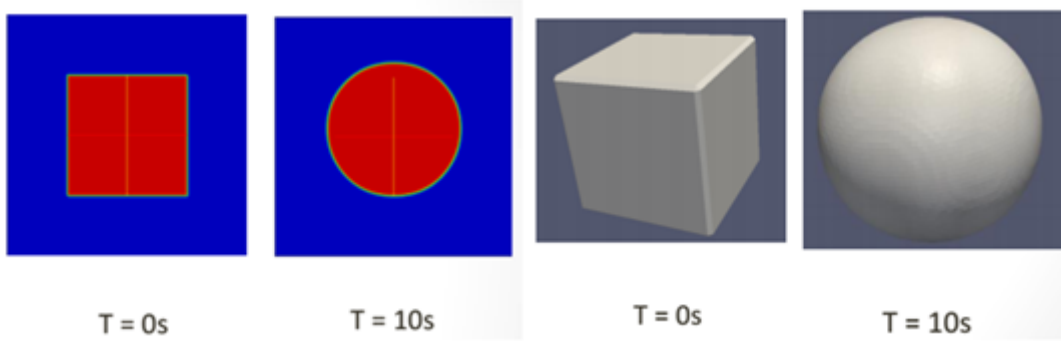


Figure 3.5: Surface tension test

numbers here indicate the number of cells along x, y and z directions respectively as mentioned in the *constantpolyMesh/blockMeshDict* file. ZeroGradient boundary conditions (Newman) are imposed on all faces for the volume fraction  $\alpha$ .water as well as velocity. For pressure, all faces except the top, has a zeroGradient BC attributed to it. The top face has a Dirichlet boundary condition assigning a value of 0 for pressure.

Grid Size	25	50	75	100	125	150
Actual SA ( $m^2$ )	4.836	4.836	4.836	4.836	4.836	4.836
SA after 10 sec ( $m^2$ )	5.144	5.098	4.700	4.834	5.002	4.952
error after 10 sec	0.064	0.054	0.028	0.0004	0.0343	0.024
SA after 20 sec ( $m^2$ )	5.143	5.097	4.700	4.832	4.923	4.892
error after 20 sec	0.063	0.054	0.028	0.0008	0.018	0.012
difference in error	-0.0003	-0.0003	0	0.0003	-0.0162	-0.0123

Table 3.1: Surface area of sphere and corresponding error

The same result has been plotted below for better understanding:

**Discussion:** It can be clearly seen from Figure 3.6 that area after 10 sec and 20 sec is different for finer grids. This can be reasoned out as follows. When the bubble starts changing its shape from cube to sphere it keeps on oscillating until it reaches a steady state. The magnitude of these oscillations keep on reducing. On a coarser grid the magnitude is small as compared to grid size. This is so because



### 243.2 Testing the surface tension model implementation of the solver:

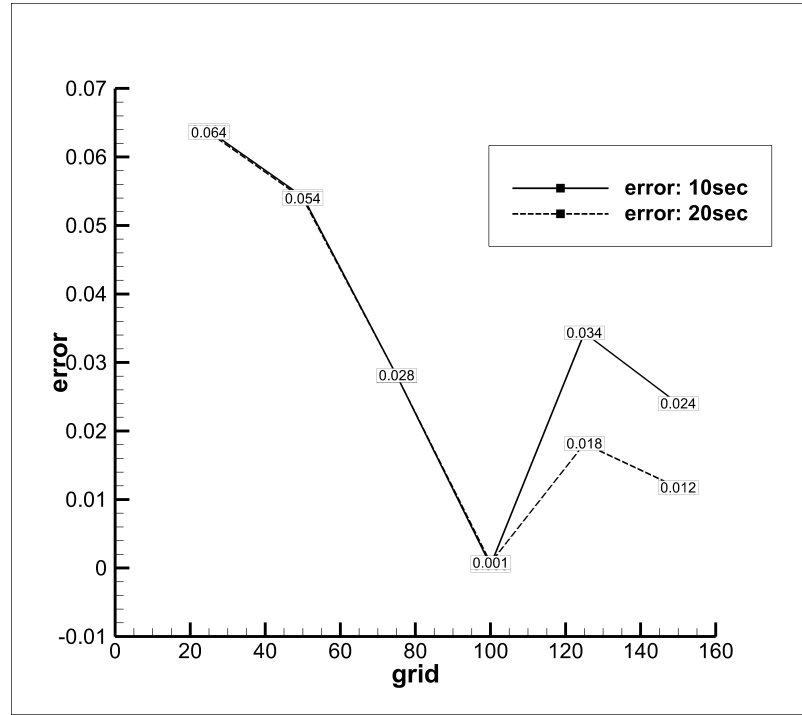


Figure 3.6: Errors using implicit solver : interFoam

the curvature estimation technique also depends on grid resolution. Hence the effect of time is not so pronounced till  $100 \times 100 \times 100$  grid is used. As we go to the finer grids, the amplitude of oscillation becomes comparable to grid size and hence the difference in area is observed.

#### 3.2.2 2D: Square to circle test

While simulation the 2D as well as 3D test, it was also observed that the bubble was displaced along +y direction. Ideally, in the absence of any external force and zero gravity, the bubble should remain stationary. To investigate further, the bubble centroid was plotted vs time for all grid sizes. The results are reported in the Figure 3.7 below:

**Discussion:** A general observation that can be made from the plot in Figure 3.7 is that as the grid becomes finer the bubble tends to stay stationary. Since the bubble height is affected by the grid size we conclude that the external force causing the bubble to rise is associated with the grid. The only force that comes into picture is the surface tension force. So, to understand the anomalous

### 3.2 Testing the surface tension model implementation of the solver:25

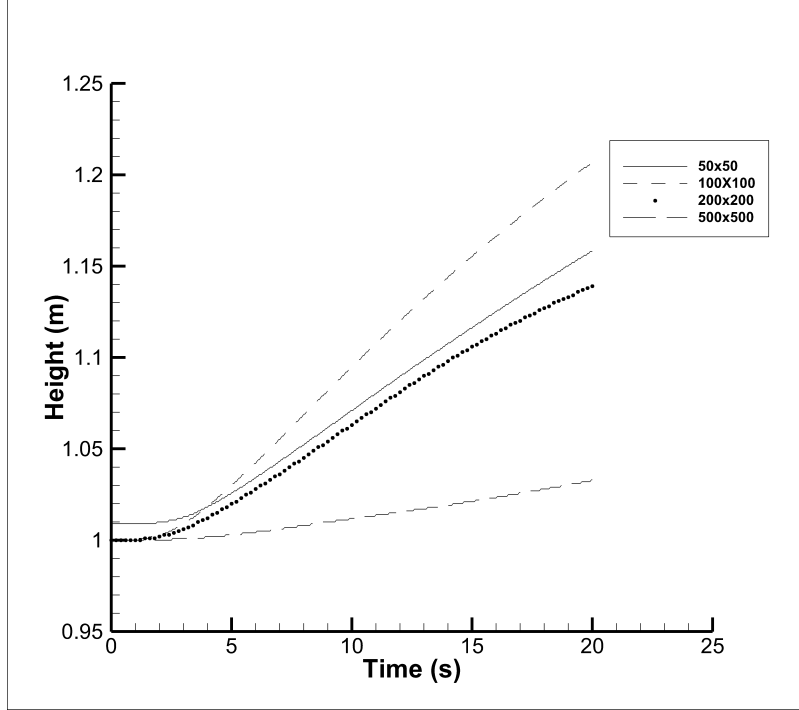


Figure 3.7: Bubble centroid vs time for different grids: interFoam

behavior, csf formulation for surface tension modelling is re-visited.

The csf formulation for surface tension model reads:

$$F_{st} = \sigma k \nabla k \quad (3.1)$$

where  $k = -\nabla \cdot \frac{\nabla \alpha}{|\nabla \alpha|}$  is an estimate of curvature using the volume fraction  $\alpha$ .

From above equation it is clear that surface tension force calculation depends on calculation of gradient of which in turn depends upon the grid used. Hence we get different results for different grid sizes.

Also, another most prominent reason for the anomalous behavior is the spurious currents. Spurious current is a small amplitude artificial velocity field which arises from an imbalance between discretized forces in multiphase or multi-component flows. An attempt has also been made to report the value of spurious currents using rbsFoam which will be reported later.

The same analysis is presented for rbsFoam: First the results for cube to sphere test. The data for surface area is tabulated below in table 3.2. The same data is presented graphically in figure 3.8 for better understanding.

### 263.2 Testing the surface tension model implementation of the solver:

Grid Size	25	50	75	100	125	150
Actual SA ( $m^2$ )	4.836	4.836	4.836	4.836	4.836	4.836
SA after 10 sec ( $m^2$ )	5.150	5.173	4.802	4.802	5.117	4.952
error after 10 sec	0.065	0.070	0.070	0.070	0.058	0.024
SA after 20 sec ( $m^2$ )	5.144	5.106	4.722	4.894	5.015	4.893
error after 20 sec	0.0637	0.0558	0.024	0.012	0.037	0.012
difference in error	0.0013	0.0139	0.0166	0.0050	0.0210	0.0112

Table 3.2: Surface area of sphere and corresponding error: rbsFoam

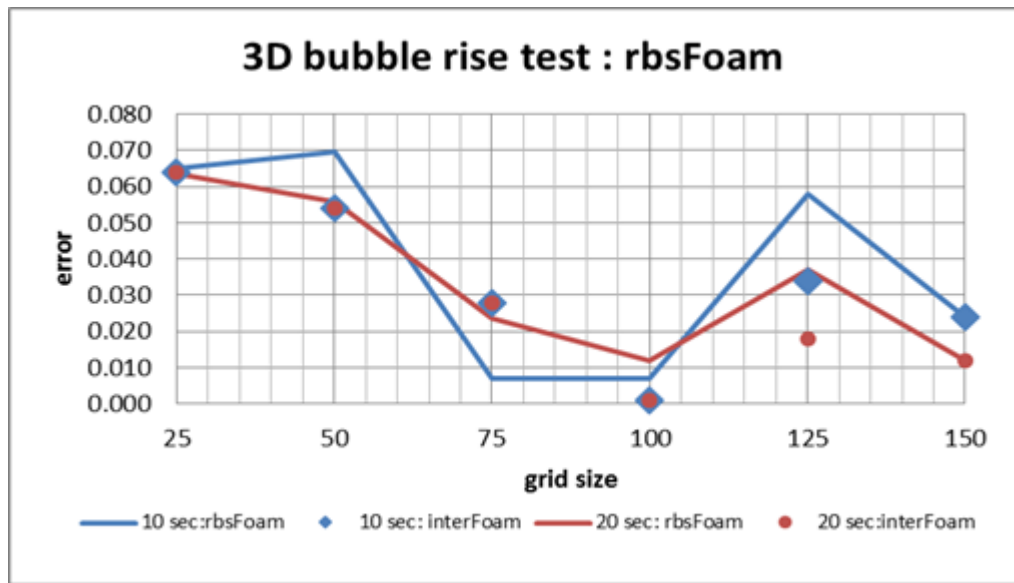


Figure 3.8: Errors using explicit solver : rbsFoam

Also, a 2D square to circle test was performed to study the behavior of bubble when subjected to an explicit algorithm and compare the same with the results from explicit algorithm. The results are shown in figure 3.9.

It can be clearly observed from the above plot that the behavior of the bubble when operated upon by the explicit algorithm is chaotic. A general trend cannot be seen in the plot. For finer meshes, the bubble is highly unstable. From the simulations it was also observed that the bubble stays bounded inside the computational domain only if we maintain a max courant number of 0.01. Else, due

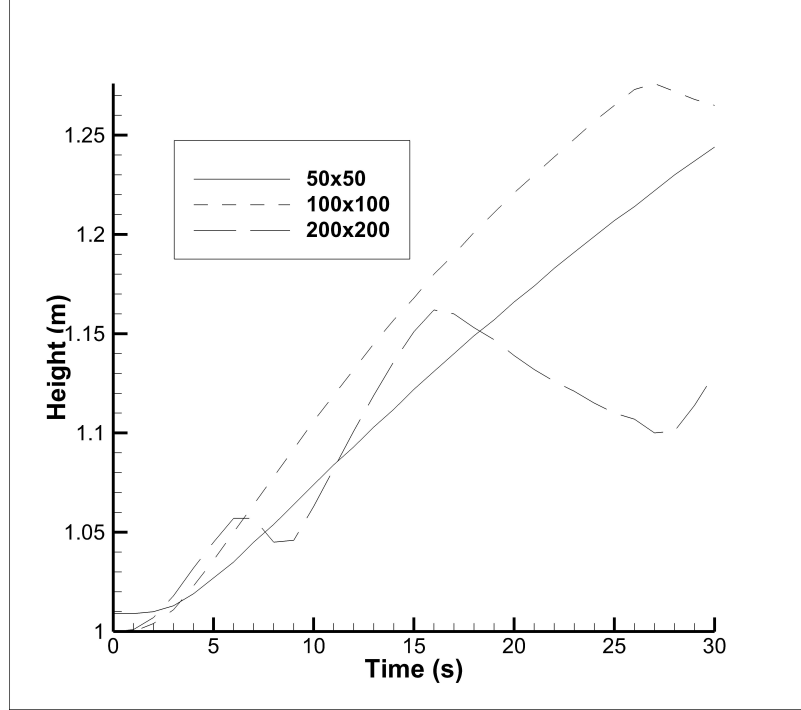


Figure 3.9: Bubble centroid vs time for different grids: rbsFoam

to the spurious currents the bubble suddenly disappears from the computational domain. A detailed analysis of the spurious currents is reported below:

### 3.3 Static droplet test: a measure of Spurious currents

A static spherical droplet is suspended in a cubical domain of side  $L = 0.1\text{m}$  in zero gravity. A droplet of diameter  $0.5L$  is placed at the center of the domain. The density ratio is unitary, i.e.  $\rho_l = \rho_g = 1000$  and so is the viscosity ratio i.e.  $\mu_l = \mu_g = 0.03$  and a surface tension value of  $\sigma = 0.045$ . Ideally one would expect no motion of the interface since the system is in equilibrium. However, the curvature calculation scheme might result in non-symmetric pressure distribution, and since  $U$  calculation is linked to pressure calculation, a fluctuating property called spurious currents comes into play. Four meshes are used:  $16^3$ ,  $32^3$ ,  $64^3$  and  $128^3$ . The time step is fixed to  $0.79 \times 10^{-3}$  s. The solver is run for 51 iterations (50 time steps).

Grid Size	L1		L2		Umax	
	Menard et al	rbsFoam	Menard et al	rbsFoam	Menard et al	rbsFoam
16x16x16	1.2e-3	2.11e-1	4.86e-3	2.26e-1	2.93e-5	5.37e-3
32x32x32	1.16e-3	2.36e-1	4.78e-3	6e-2	7.61e-5	1.64e-2
64x64x64	1.23e-3	1.95e-1	4.91e-3	4.12e-2	6.8e-3	1.14e-2
128x128x128	1.19e-3	1.95e-1	4.82e-3	3.93e-2	8.26e-6	1.15e-2

Table 3.3: Surface tension test results: spurious currents

Hence the end time for simulation is  $0.408 \times 10^{-1}$  s. The errors are calculated as follows:

$$L_1 = \left| \frac{\sum_{ijk}^{N_d} P_{ijk} - P_d}{N_d P_d} \right| \quad (3.2)$$

$$L_2 = \frac{\sum_{ijk}^{N_d} (P_{ijk} - P_d)^2}{N_d P_d^2} \quad (3.3)$$

here  $N_d$  is the number of cells in the interior of the droplet defined by  $\alpha \leq 0.99$ .  $P_{ijk}$  is the pressure inside a cell.  $P_d$  is the theoretical pressure difference denoted by:  $P_d = 2\sigma/R$ . The error is reported in the table below. The results are compared with the work of Menard et al.[49]. and reported in table 3.3

The maximum magnitude of velocity after 51 iterations is also reported. From the table it can be clearly seen that the magnitude of errors  $L_1$  and  $L_2$  as well as magnitude of maximum velocity in rbsFoam is large as compared to the work of Menard et al. [49]. This indicates poor implementation of the surface tension model in OpenFoam.

### 3.4 Testing vof model:

To test the vof model implementation, only the transport equation for volume fraction should be solved. Rest all other modules like the Navier Stokes part, etc should not come into play. Keeping in mind the above objective, the vortex in a box test and dual vortex stretching tests are designed. Here, a non-uniform position and time dependent velocity field is imposed at every grid point so that the Navier Stokes part of the code need not be solved for velocity. With this analytical velocity field as input, the volume fraction transport equation is solved.

**Description:**

For the 2D case, we choose a periodic domain of unit side. A circle with center at  $(0.5, 0.75)$  units is placed inside the box. The streamlines inside the box at time  $t=0$  are as shown below:

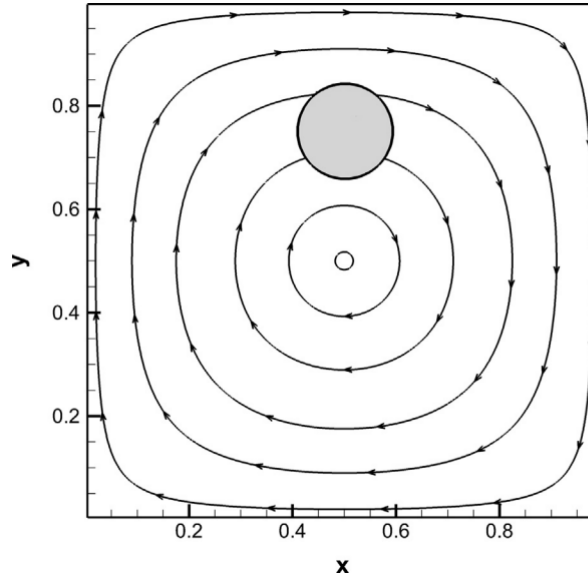


Figure 3.10: Streamlines at  $t = 0$

The velocity field is calculated based upon the stream function given by:

$$\psi(x, y, z, t) = \frac{1}{\pi} \cos\left(\frac{\pi t}{T}\right) \sin^2(\pi x) \sin^2(\pi y) \quad (3.4)$$

The velocity is calculated from the stream function as:  $u = \frac{\partial \psi}{\partial y}$  and  $v = -\frac{\partial \psi}{\partial x}$

The sinusoidal temporal term makes the function periodic. Hence, after a time period of  $T/2$ , the flow reverses and the system starts coming back to its initial configuration. Due to the initial vortex, the circle is stretched along the streamlines until a maximum is reached at a flow time of  $T/2$  after which the flow reverses. The simulations are run for a total flow time of 2 sec and 8 sec. Three grids are used for the test: 64x64, 128x128, 256x256. Under ideal hypothetical conditions after the cycle time the code should be able to recover the circle.

The error is calculated as:

Time = 2sec								
Grid	E(i)	T(i)	E(r)	T(r)	Eref	E(i)/Eref	E(r)/Eref	T(r)/T(i)
64x64	0.004	614	0.007	363	-	-	-	0.59
128x128	0.003	3713	0.003	1333	0.001	3	3	0.36
256x256	0.001	12945	0.001	6001	-	-	-	0.46

Table 3.4: Error for T=2 sec using interFoam and rbsFoam

Time = 8sec								
Grid	E(i)	T(i)	E(r)	T(r)	Eref	E(i)/Eref	E(r)/Eref	T(r)/T(i)
64x64	0.023	3504	0.002	2321	-	-	-	0.66
128x128	0.008	13025	0.008	7145	0.003	2.75	2.75	0.36
256x256	0.004	49710	0.004	23762	0.002	2.005	2.005	0.48

Table 3.5: Error for T=8 sec using interFoam and rbsFoam

$$E = \frac{\sum_{i,j} |\alpha_{i,j}^f - \alpha_{i,j}^i|}{N_x N_y} \quad (3.5)$$

where f and i stand for final and initial time steps respectively. Note that here  $N_x$  and  $N_y$  stand for number of cells along x and y directions and not points. The error for both time periods is reported below. Also a comparison has been made between the implicit solver interFoam vs. the explicit solver rbsFoam on grounds of accuracy and computation time.

In the above tables 3.4 and 3.5 E(i) stands for error calculated using Equation 3.4 using the implicit solver, interFoam whereas E(r) stands for explicit solver, rbsFoam. Similarly, Eref stands for error from reference [51]. T(i) and T(r) stands for cpu time taken by implicit and explicit solvers respectively. Figures 3.11 and 3.12 show the contours of the droplet at maximum deformation (corresponding to half of flow time) and final position for times 2 sec and 8 sec respectively.

#### Discussion:

An important observation that can be made from the above table is that implicit as well as explicit solver have the same error which is quite expected as none of the solver actually solves for the velocity field. And yet we are able to obtain same set of results in almost half time as compared to the implicit solver. The reference chosen here is native code developed by Dr. Rajesh Reddy which is an in-house

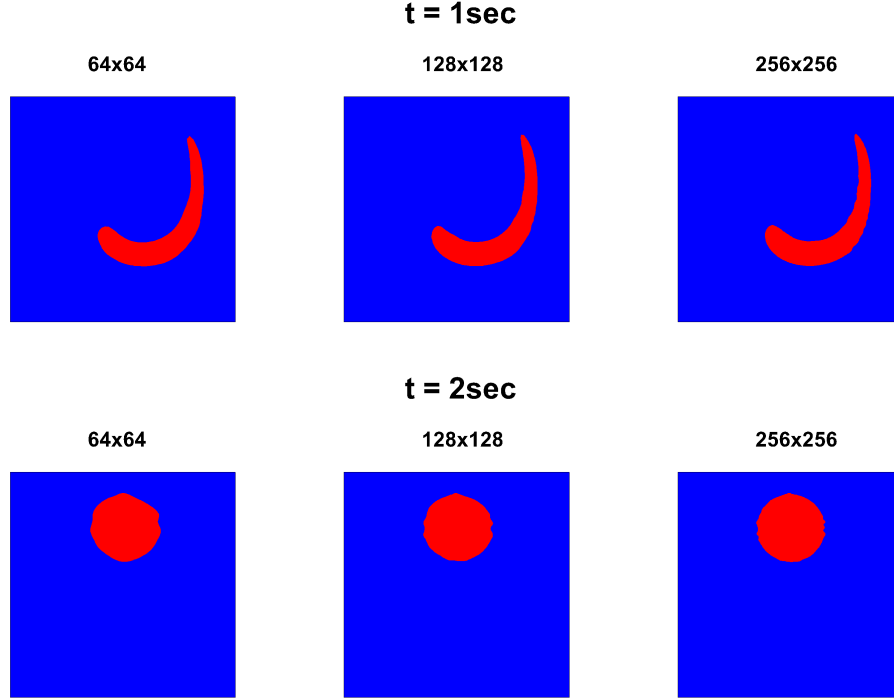


Figure 3.11: Droplet shapes for different grids at times 1s and 2s

CLSVOF multiphase code [51] . The error reported from rbsFoam is significantly large as compared to the reference indicating a poor implementation of vof model.

For 3D case a dual vortex stretching problem is chosen. A spherical droplet of radius 0.2 is placed at the point (0.35,0.35,0.35) inside a periodic cubical domain of unit dimension. The velocity field inside the domain is given by:

$$u(x, y, z, t) = 2\sin^2(\pi x)\sin(\pi y)\sin(\pi z)\cos(\pi t) \quad (3.6)$$

$$v(x, y, z, t) = -\sin(\pi x)\sin^2(\pi y)\sin(\pi z)\cos(\pi t) \quad (3.7)$$

$$w(x, y, z, t) = -\sin(\pi x)\sin(\pi y)\sin^2(\pi z)\cos(\pi t) \quad (3.8)$$

The simulations are run for a cycle time of 1 sec. The contours at time  $t=0.5$  s and  $t=1$  sec are shown in Figure 3.13. It can be clearly observed that as the grid becomes finer, the initial spherical configuration can be recovered better.



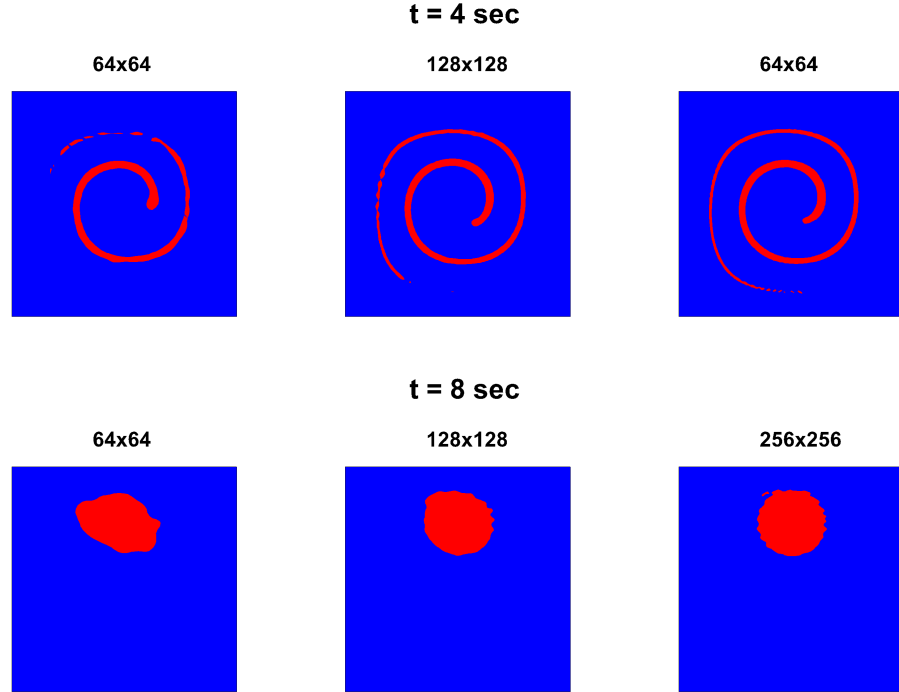


Figure 3.12: Droplet shapes for different grids at times 4s and 8s

If the simulations are allowed to run for 8 sec, the resulting data reveals mass loss at various time instances. The following plot reports the mass loss at various time instances.

As one would expect from a vof model, excellent mass conservation is reflected from the plot in Figure 3.14. During the first half of the cycle, the sphere stretches under the influence of counter-rotating vortices. During this half cycle, mass loss is seen from the plot. But during the second half, the mass loss is recovered. This mass is conserved throughout the cycles (8 in total). The error is calculated as follows:

$$L_1 = \frac{1}{N^3} \left( \sum_{i,j,k=1}^N |\alpha_{i,j,k}^f - \alpha_{i,j,k}^i| \right) \quad (3.9)$$

N indicates number of cells along either x, y or z direction. (Since same number of cells are used along all directions.)

The following table reports the errors obtained from rbsFoam and also those

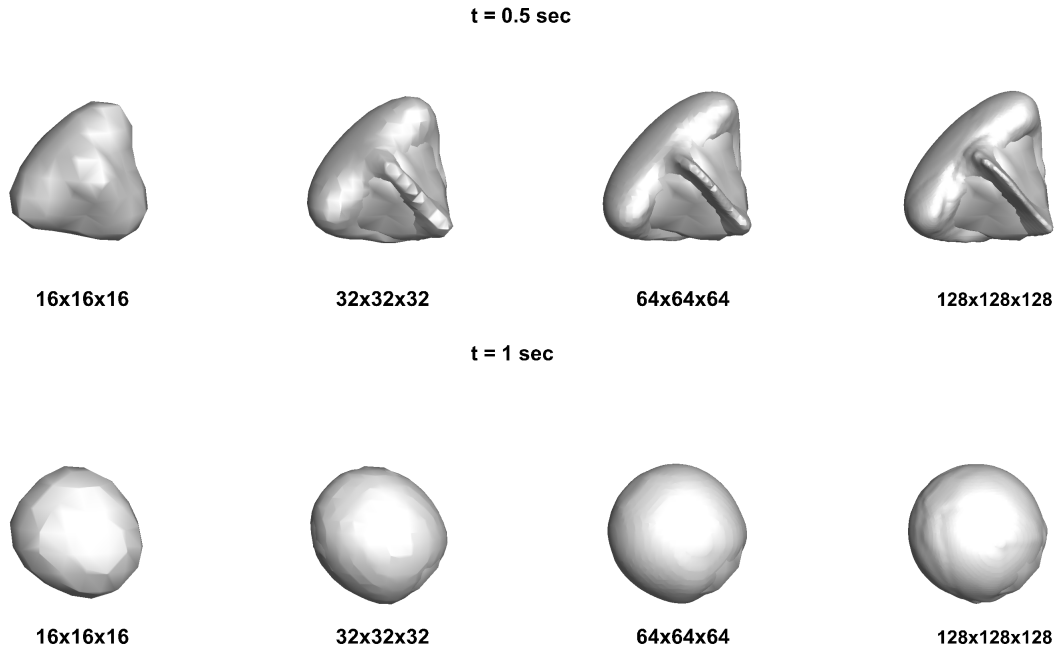


Figure 3.13: Mass conservation for 8 cycles

Grid	$L_1(\text{rbsFoam})$	$L_1(\text{Menard et al})$
16x16x16	1e-2	1.83e-2
32x32x32	5.8e-3	8.93e-3
64x64x64	2.65e-3	6.77e-3
128x128x128	1.4e-3	3.26e-4

Table 3.6: Error for dual vortex stretching test

reported in the Reference Menard et al. [49].

The findings from table 3.6 reveal that error obtained from rbsFoam is slightly less as compared to the standard reference code by Menard et al. That being said, for the simulations performed using rbsFoam, a max courant number of 0.01 was maintained indicating very small time steps. Whereas, no comments on time stepping were made in the reference. Hence, these findings are inconclusive.

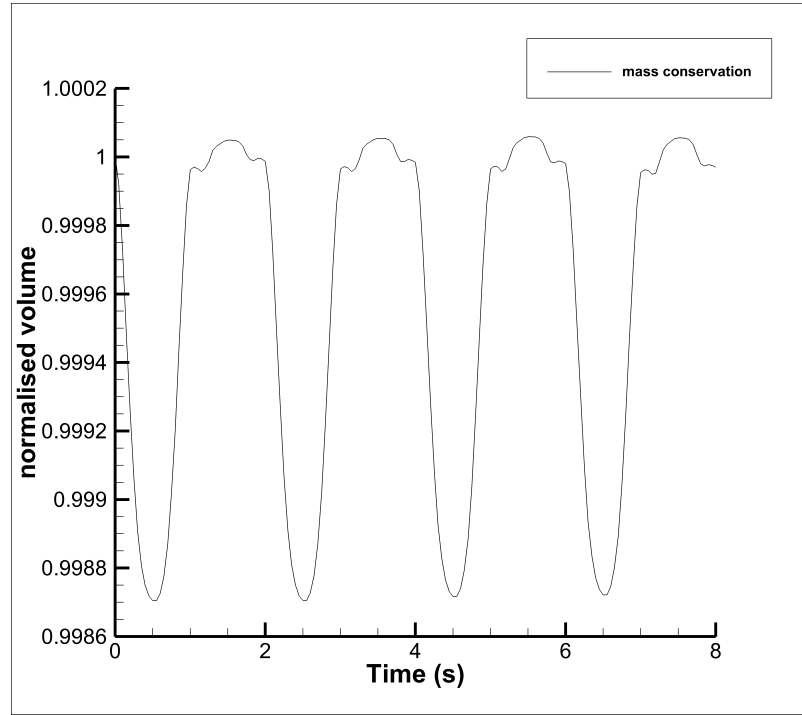


Figure 3.14: Mass conservation for 8 cycles

Test case	$\rho_1$	$\rho_2$	$\mu_1$	$\mu_2$	$g$	$\sigma$	Re	We	$\frac{\rho_1}{\rho_2}$	$\frac{\mu_1}{\mu_2}$
1	1000	100	10	1	0.98	24.5	35	10	10	10
2	1000	1	10	0.1	0.98	1.96	35	125	1000	100

Table 3.7: Case parameters for 2D bubble rise

## 3.5 Testing NS, vof and ST modelling combined:

### 3.5.1 2D bubble rise

One of the most standard and stringent tests done to validate the coupling between the above mentioned modules is the bubble rise test. A 2D domain of dimensions 1m x 2m domain is chosen. A bubble of radius 0.25m is placed at (0.5,0.5). The simulations are performed for different density and viscosity ratios. The details of the physical parameters chosen for the task are mentioned in the table below:

Re = Reynold's Number and We = Weber Number

The grids chosen for the task are 40x80, 80x160, and 160x320. The y centroid of the bubble and rise velocity is plotted against the flow time. The y centroid and rise velocity are calculated as follows:

$$y_c = \frac{\sum_{i=1}^N \alpha_i y_i}{\sum_{i=1}^N \alpha_i} \quad (3.10)$$

$$y_c = \frac{\sum_{i=1}^N \alpha_i v_i}{\sum_{i=1}^N \alpha_i} \quad (3.11)$$

The important point to be noted here is that N stands for number of points which is different from  $40 \times 80 = 3200$ , since 40x80 or 80x160 indicate the number of cells along each direction.

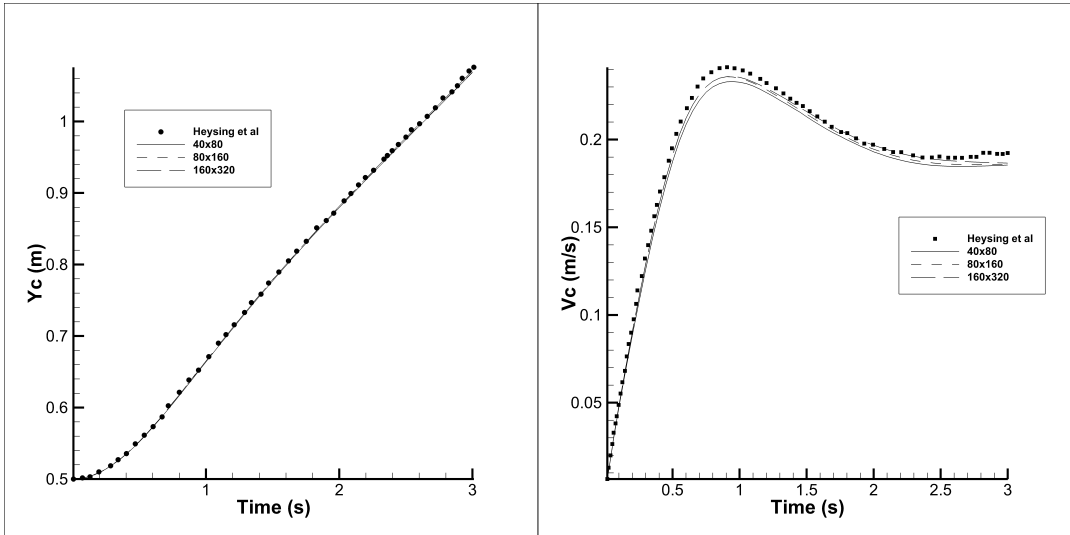


Figure 3.15: 2D bubble rise results for density ratio = 10

As it can be clearly seen from plots in Figure 3.15, highly grid convergent and accurate results are obtained for a density ratio of 10 an compared to those of Heysing et al. [50]

Again from Figures 3.15 and 3.16 is can be clearly seen that the simulation results obtained from rbsFoam match decently with the experimental results reported by Heysing et al. Another important point to be considered is the time

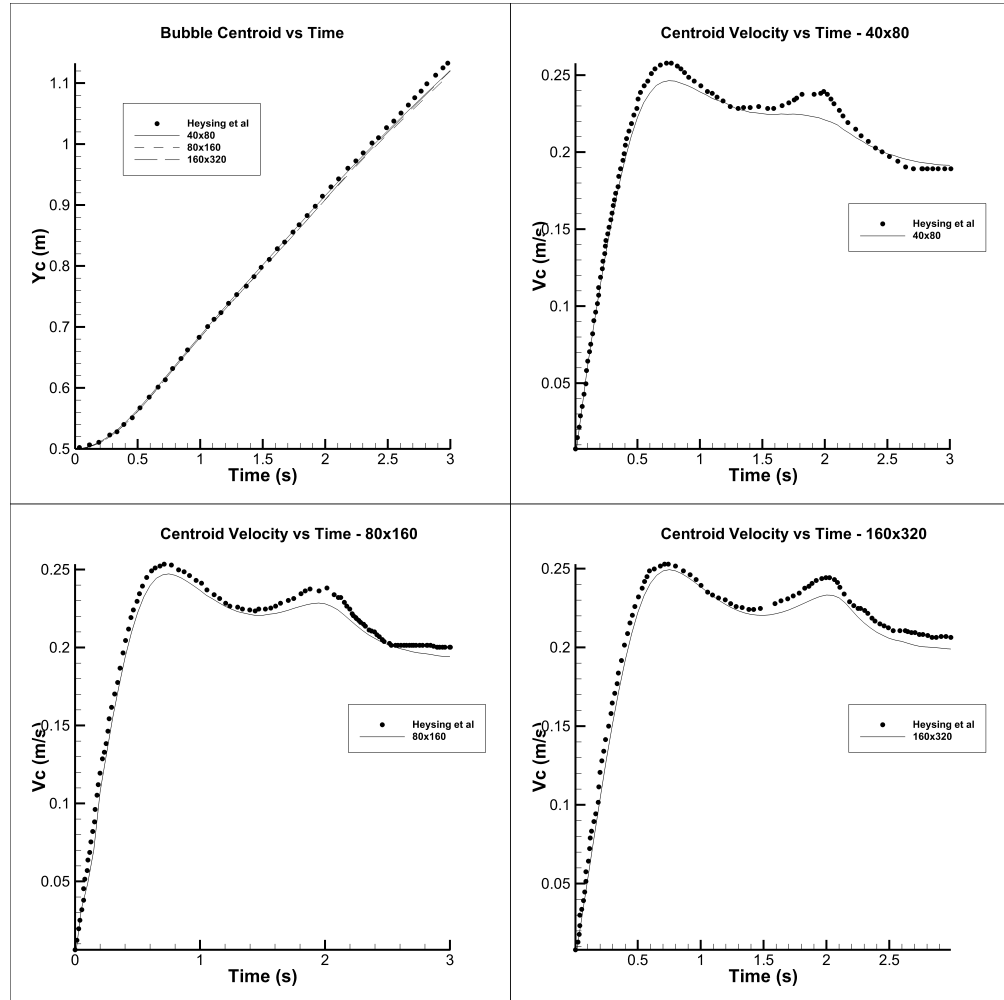


Figure 3.16: 2D bubble rise results for density ratio = 1000

required for simulations using the explicit method vs the implicit method. The same has been reported in the table 3.8

Hence, from above results it can be concluded that the solver is decently accurate and with the optimum case setup obtained from the above test cases, a real life problem can be successfully addressed.

### 3.5.2 Droplet splashing

The bubble rise case mentioned above tests the coupling between NS, vof and surface tension modelling in 2D. But simply validating a 2D case isn't good enough.

Density ratio	solver	40x80	80x160	160x320
10	clocktime for interFoam (sec)	108	909	8649
	clocktime for rbsinterFoam (sec)	67	471	3629
<b>Speed-up</b>	<b>interFoam/rbsinterFoam</b>	<b>1.61</b>	<b>1.93</b>	<b>2.38</b>
1000	clocktime for interFoam (sec)	143	1003	11168
	clocktime for rbsinterFoam (sec)	90	724	10260
<b>Speed-up</b>	<b>interFoam/rbsinterFoam</b>	<b>1.59</b>	<b>1.39</b>	<b>1.09</b>

Table 3.8: Clocktime comparison: explicit vs implicit solvers for density ratio 1000

2D and 3D behavior of a solver is significantly different. For example, for curvature estimation, to evaluate the surface tension and interface location, in 2D, a line needs to be reconstructed whereas in 3D a plane needs to be reconstructed. Hence, a droplet splashing experiment was chosen for 3D validation.

A water drop of diameter  $D_o = 5.1mm$  is released inside a cubical domain of dimensions  $6.5D_o \times 1.75D_o \times 6.5D_o$  with a velocity of 2.14 m/s along -y direction. The value of acceleration due to gravity,  $g$ , is set to  $9.81 \text{ m/s}^2$  along -y direction. The domain is filled with water upto a height of 0.5mm.

As the droplet hits the water film, a crater is formed which grows in diameter with respect to time. Also, due to splashing the water surface ejects in the form of ligaments which rise along the +y direction further giving rise to satellite droplets.

For mathematical validation the non dimensional crown diameter defined as  $D^* = \frac{D_c}{D_i}$  is plotted against non-dimensional time defined by  $t^* = \frac{tU}{D_i}$ . The results are compared against experimental results reported by Cosalli et al. The calculation of crown diameter is a challenging task. In the experimental results reported, the diameter was calculated as an average of diameters measured at various points. The definitions of all the diameters can be found in Cosalli et al. The plots can be seen in Figure 3.19

Though at first sight, the results in figure 3.19 might appear to be bad, the uncertainty in estimation of the various diameters experimentally justifies the mis-match. So. the results fall well within the tolerance zone.

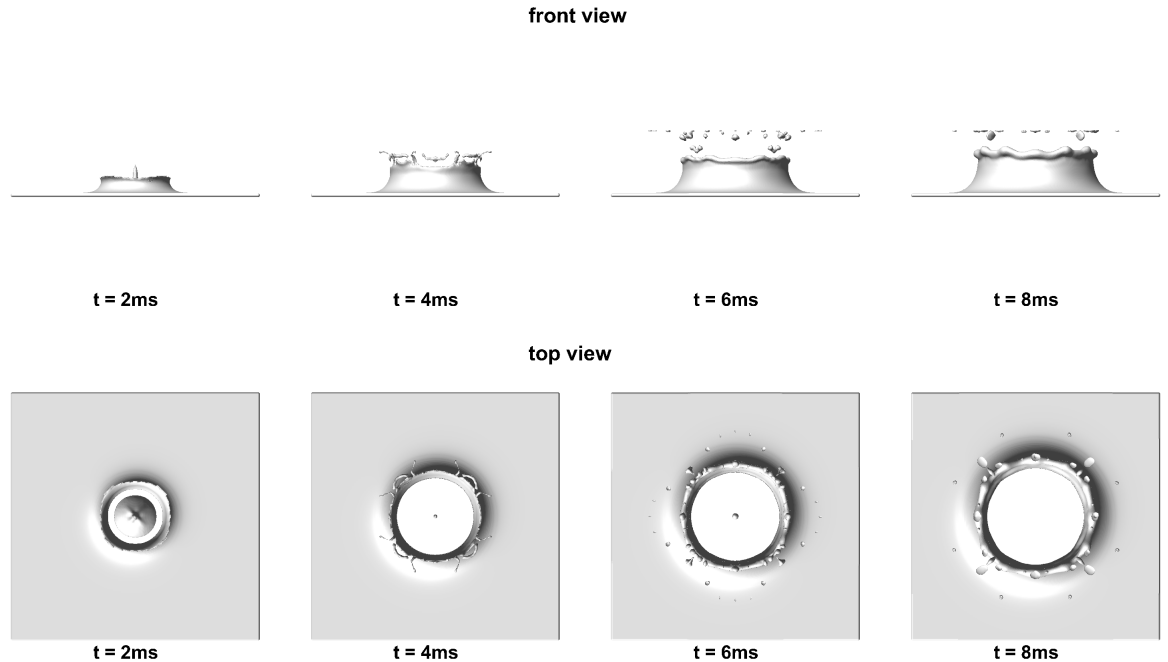


Figure 3.17: Results for droplet splashing - A

### 3.6 Closure

This section summarizes the conclusions that can be made from the validations performed above.

The results from the lid driven cavity test indicate excellent Navier Stoke's solution implementation. Another study was carried out to compare OpenFOAM's results with FLuent, which is a commercial CFD package. The results gave a complete match. The square to circle and cube to sphere tests indicated a descent, though not good Surface tension model. Similar finding were also found from the static droplet test which indicated larger spurious current amplitude as compared to literature. The comparison between explicit and implicit methods confirmed the known facts that implicit solvers are more stable as compared to explicit. Also, a minimum cfl number of 0.01 needs to be maintained to get accurate results. This setting needs to be ensured for both courant numbers, one based on velocity of the flow field, and second, based on the interface velocity. 2D vortex stretching and 3D vortex stretching test cases reported less accurate vof model implementation in OpenFOAM. The interface capturing technique needs improvement. Or an altogether different interface capturing approach like Level

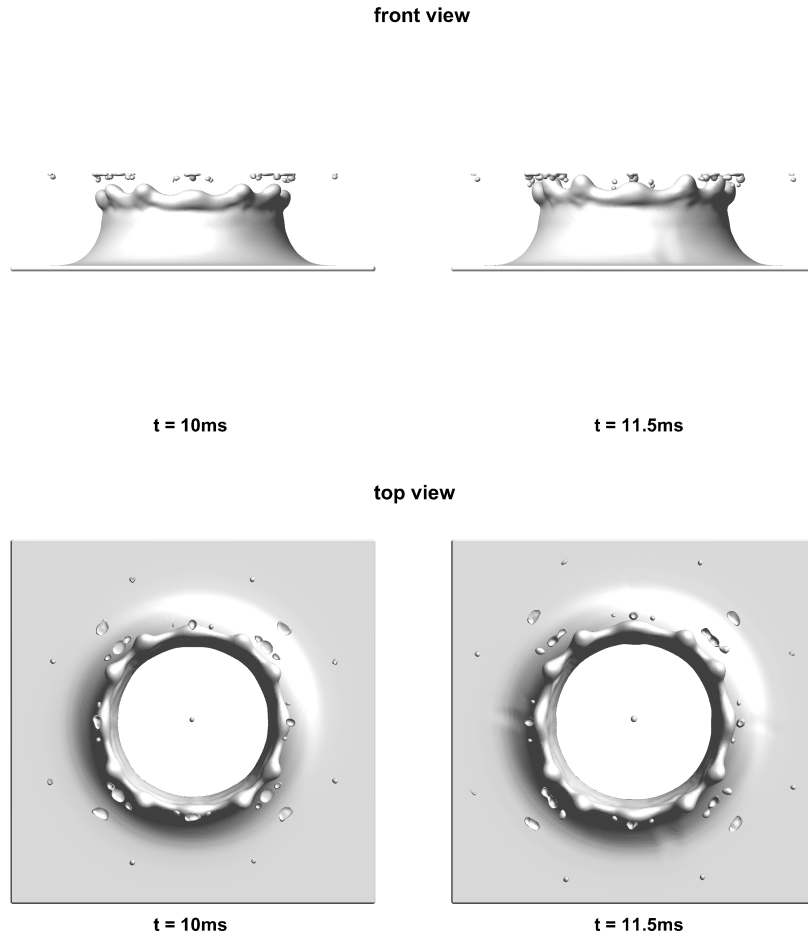


Figure 3.18: Results for droplet splashing - B

Set or a Coupled Level set vof (CLSVOF) approach needs to be adopted for further improvement. 2D bubble rise and droplet splashing tests gave satisfactory results indicating good communication between all 3 modules namely, NS, surface tension and vof model implementation. With the above knowledge of optimum test case setup and scope of errors, a droplet collision study is presented in the next chapter.



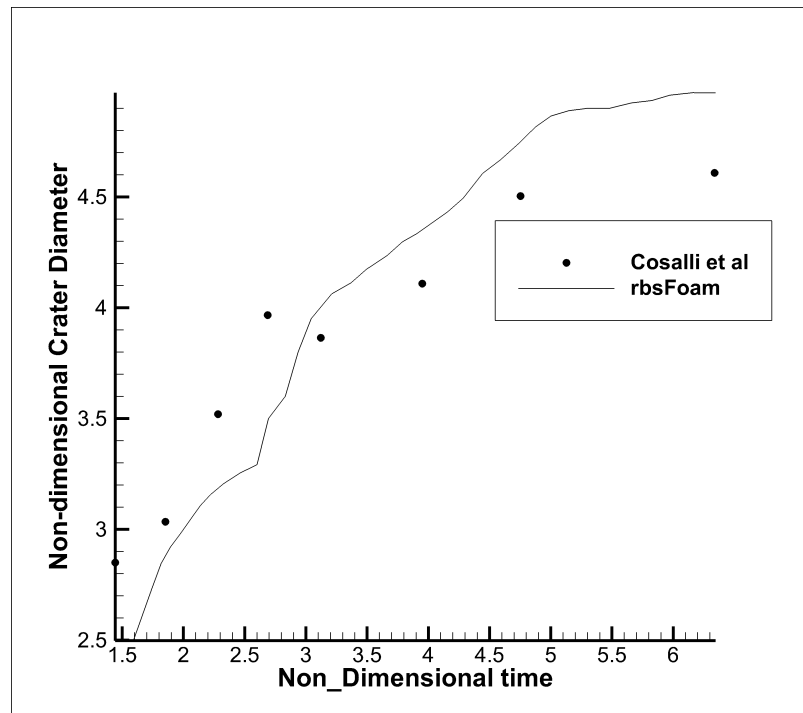


Figure 3.19: Results for droplet splashing - B

# Chapter 4

## Droplet Breakup

In this chapter, droplet dynamics is characterized. The various parameters that come into play along with their significance is explained in detail.

While dealing with bubble and droplet dynamics, the forces that come into picture are: inertial force, surface tension force and aerodynamic force. For example, when a water drop is subjected to a blast of air, the water droplet has cohesive forces which manifest themselves in the form of surface tension, trying to maintain the spherical shape of the droplet keeping it together. Opposing this force is the inertial force exerted by the incoming blast of air trying to deform the water droplet. This inertial force experienced by the droplet usually depends upon its size characterized by droplet diameter  $D_0$ . When the inertial forces dominate over the surface tension, droplet breakup occurs. To address all these forces at once, one needs a non-dimensional number. Hence, Weber number is introduced. Weber number (We) is defined as the ratio of inertial force to surface tension force. Mathematically :

$$We = \frac{\rho U^2 D}{\sigma}$$

where  $\rho$  stands for density of droplet material,  $U$  is the magnitude of relative velocity of drop w.r.t. surrounding fluid,  $D$  is characteristic droplet diameter and  $\sigma$  is the surface tension of the pair.

Most of the droplet breakup regimes are characterized based on  $We$  of the flow. For example, when a drop is subjected to an air blast, for a  $We$  of 3.4 the droplet undergoes oscillatory deformation but no breakup is observed implying that the deforming inertial forces are weak as compared to the surface tension force. If the

We is increased to 13.5 bag breakup is observed. Here the droplet first deforms into a liquid disc which later on takes the shape of a bag with a disintegrating liquid disc at its center. When the We is further increased to a value of 100, the droplet undergoes sheet thinning breakup. In this breakup mode, the droplet first deforms to form a thin sheet of fluid which then further disintegrates into smaller droplets. [56]

An important point to be noted here is that, though We is an important parameter, the breakup mode also depends on system configuration. In case of off-center droplet collision, for same We number, totally different breakup regime can be expected with changes in impact parameter.

Before simulating droplet breakup, one needs to decide upon the resolution needed to successfully capture the tiny droplets formed in the process. To validate the same, the case presented by [53] was replicated to test the performance of the new solver. The reference stated that dynamic mesh refinement was used to simulate this case. But, as discussed in appendix, OpenFOAM's dynamic mesh refinement utility is not accurate or fast enough for the task. Hence a uniform mesh having a resolution equal to the finest level in reference was used. That resolution was reported to be 50 cells per droplet diameter. A decent match with the volume fraction contour confirmed that this resolution was decent enough to capture the droplets.

In this thesis, the results for off-center droplet collision are presented. A lot of possibilities can be thought off when two droplets are allowed to collide. They are listed below:

- Droplets of different fluids can collide in same environment.
- Droplets of different diameters can collide in a head on fashion.
- Droplets of different diameters can collide in an off-center fashion with varying levels of offset.
- Any possible permutation and combination of the above.

Since all of these are high fidelity simulations, simulating all cases is computationally very expensive and time consuming. Hence, two possibilities are explored. For one set of cases, a diameter ratio  $\frac{D_{large}}{D_{small}} = 1.5$  was chosen and analysis was done by varying the impact parameter from 0,0.25,0.5,0.75. For the

other set, an impact parameter of  $x = 0.5$  was fixed and the cases were simulated for different diameter ratios varying from 1,1.25,1.5 to 1.75. The details of these parameters are given in the next section.

## 4.1 Parameters affecting droplet collision:

Droplet collision has been extensively studied for decades. Various experiments have been performed to study the effects of different physical and geometrical parameters on droplet collision. Offset ratio also known as impact parameter has a very pronounced effect on droplet collision. Depending upon impact parameter, the result of the collision may be a stable coalescence, transient coalescence finally resulting into droplet breakup. Also, changing the material of the droplet changes the way droplet interacts with another droplet. For instance, it has been experimentally reported that for head on collision of water droplets no bouncing is observed, where as for hydrocarbon droplets bounce under similar conditions. Another important area of exploration is the effect of weber number. With increasing Weber number following collision regimes have been reported in literature: droplet bouncing, stable droplet coalescence, droplet stretching finally leading to separation. Also, during separation depending upon the We and impact parameter satellite droplets may or may not be generated. A quantification of these satellite droplets and their relation to the causative parameters is another important area of study. To study the role of viscosity alone, droplet collision experiments have also been performed in vacuum thus removing aerodynamic forces out of picture.

Hence, to conclude, following parameters need to be noted while addressing droplet breakup:

- Colliding droplets' diameter ratio
- Impact parameter or offset ratio
- Composition of colliding droplets
- Weber number for the flow
- Reynold's number for the flow

## 4.2 Case setup parameters:

In all the reported simulation, the grid resolution has been maintained in between 40 to 50 cells per droplet at  $t=0$ . The size of the smallest colliding droplet at  $t=0$  has been set to  $100\mu\text{m}$  and remains fixed through out. The diameter of larger droplet is obtained from the diameter ratio defined by:  $D_r = \frac{D_{large}}{D_{small}}$ . This ensures that the mesh always has a minimum resolution of  $D_0/50$ . Also, it takes some small amount of time for the flow field to develop. (Around 2 time steps). Hence, an initial separation distance of about 4 grid cells is maintained along the direction of relative velocity. (Here, the relative velocity vector is along x direction). As it can be clearly seen from the contour plots, the surface of the bubble after around 2 time steps is pretty smooth as compared to that at  $t=0$ . Reynolds number and Weber number are defined with respect to smaller droplet diameter. The values of viscosity and surface tension are assigned to be those for a water air pair at atmospheric conditions. The values are as follows:

<b>Fluid/Property</b>	<b>Water</b>	<b>Air</b>
<b>Density (Kg/m<sup>3</sup>)</b>	1000	1
<b>Viscosity (Pa.s)</b>	1e-3	2e-5
<b>Surface Tension (N/m)</b>	0.001	

Table 4.1: Physical Properties

Using above values the  $Re$  and  $We$  for all the simulations are 100 each respectively. From previous experience of test cases and validations, it was observed that for the explicit solver, a cfl number of 0.01 is needed to get the desired results. Hence, dynamic time stepping is used with a cfl number of 0.01. In case, a higher courant number is used, unstable flow field is obtained which causes the bubbles to flush out of the domain. But, an interesting observation was that a high courant number was used for a coarser mesh, the results were stable and the bubble was not flushed out. Hence, as finer meshed are used, the cfl criteria has to be made more stringent.

Offset ratio or impact parameter (denoted as  $x$ ) is defined as the ratio of distance between droplet centers along a direction normal to their relative velocity (denoted as  $y$ ) and smaller droplet diameter (denoted as  $D_0$ ). Mathematically,  $x = \frac{y}{D_0}$ . In this thesis, the effect of offset ratio and diameter ratio has been

studied at length. For the first case, a diameter ratio of 1.5 is fixed and the offset ratio is varied from 0, 0.25, 0.5 to 0.75. In the second case, offset ratio of 0.5 is fixed and the diameter ratio is varied from 1,1.25 to 1.75. The results are discussed at length in the following section. The general layout is represented in figure 4.1.

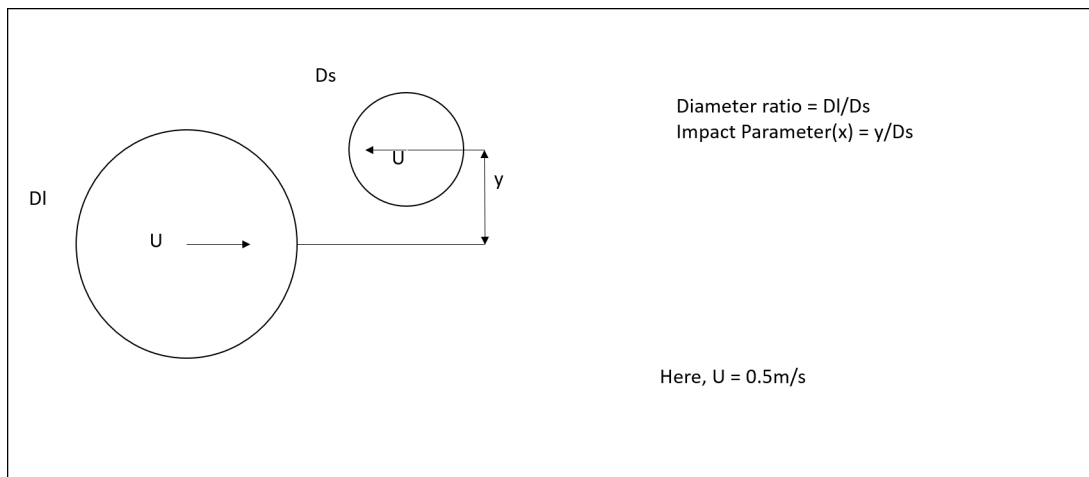


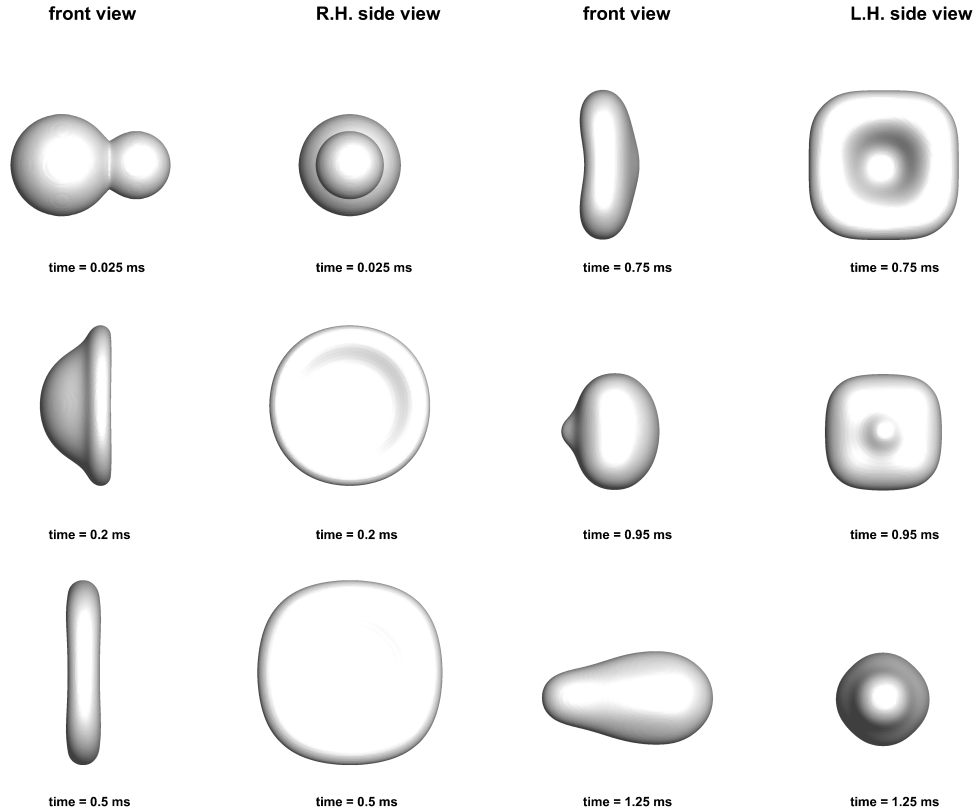
Figure 4.1: General Layout

### 4.3 Results for different offset ratios and diameter ratio = 1.5

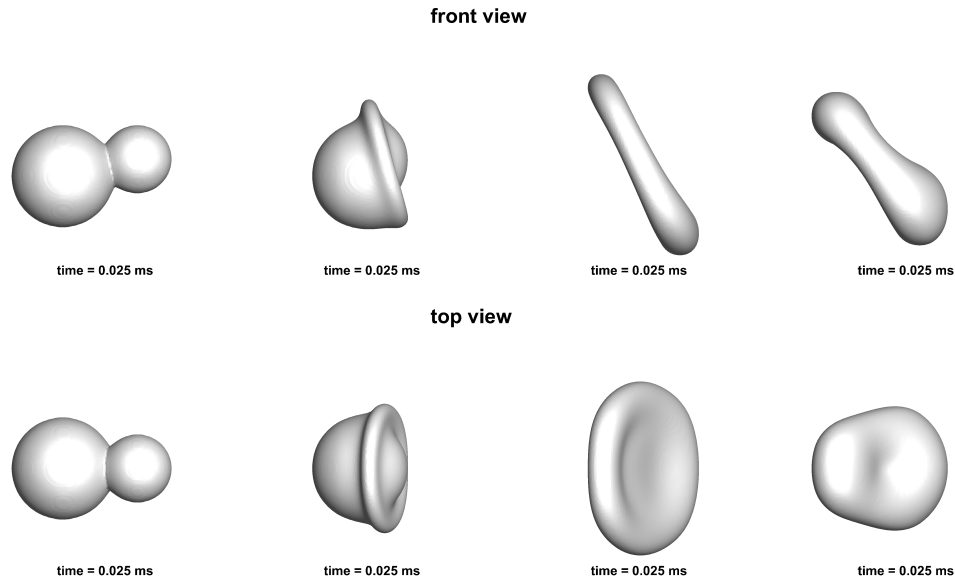
In the following simulation results, the diameter ratio was kept constant to 1.5 and the offset ratio was varied.

#### 4.3.1 Offset ratio $x = 0$

An offset ratio of 0 implies head on collision of droplets. The contours of volume fraction  $\alpha$  at different time steps are shown in figure 4.2. Since both the droplets are moving with the same velocity the larger droplet has more inertial force. Hence, upon collision, the larger droplet absorbs the impact of the smaller droplet and continues to move to the right but with reduced velocity. The smaller droplet completely coalesces with the larger droplet upon impact and transfers all its energy to the larger drop. Since the flow and the liquid is assumed to

Figure 4.2: Contour plots for  $x=0$ 

be incompressible, the entire energy supplied by the smaller drop is expended in lateral expansion of the larger droplet. But the surface tension forces oppose this occurrence and start pulling the liquid disc inside towards the core and hence the blob of liquid starts to take a distorted spherical shape. At this point a tail of liquid is seen protruding from the back. This is that fluid that came in contact with the smaller droplet and hence has zero velocity to equal and opposing inertial forces acting on it. But, as the blob progresses this volume of fluid is pulled along the direction of motion by the surface tension and shear forces and hence begins to drift along with the entire blob. The smaller droplet did not have enough energy to overcome the inertial and surface tension force of the larger droplet and break free to the other side. Hence, droplet breakup was not observed. For the definition of weber number, the larger diameter was considered. Hence,  $We$  for the flow is 100.

4.3.2 Offset ratio  $x = 0.25$ Figure 4.3: Contour plots for  $x=0.25$ 

As the offset ratio keeps on increasing, the interaction region between larger and smaller droplet also goes on decreasing. This implies lesser opposition force for both the droplets. Since, the larger droplet has a lot of inertia, the effect of this interaction region is not much pronounced on the larger drop. But, as it can be clearly seen from figure 4.3, the smaller droplet emerges with larger velocity after collision. The off-centered inertial forces cause a couple to act upon the system of droplets setting them into rotary motion. The smaller droplet easily coalesces with the larger droplet in its upper region. Since the collision is now localized in the upper region of the larger droplet, there occurs large lateral deformation in this region. This region starts to move with reduced velocities as compared to the lower region of the droplet causing elongation. But after maximum lateral expansion is achieved at the expense of the kinetic energy of the smaller droplet, the surface tension and viscous forces start to retract the deformed blob of mass. Due to the surface tension force and inertial force combined, the blob undergoes oscillatory motion along the lateral direction while it continues its rotation along the  $z$  axis. No droplet breakup is observed in this simulation as well. Surface waves can be seen at different instances in figure 4.3. It is difficult to deduce whether these waves are physical in nature or a result of numerical instability.



### 4.3.3 Offset ratio $x = 0.5$

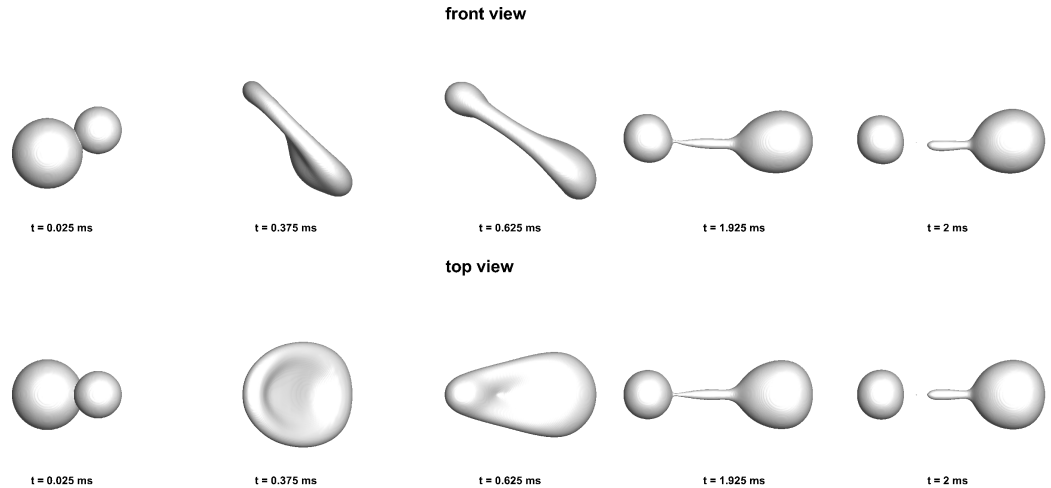


Figure 4.4: Contour plot for  $x=0.5$

As mentioned above, a further increase in offset ratio implies further reduction in the opposing force for the motion of both droplets. For an offset ratio of  $x = 0.5$ , the point of impact shifts further upwards. The smaller droplet coalesces with the upper region causing its lateral deformation and bringing it to a standstill. Due to this a very thin film is formed at the center which is very close to rupture. At this point the surface tension forces set in, causing retraction. Due to this, most of the fluid appears to be separated into two spherical drops. The blob of fluid is now acted upon by an even stronger couple resulting from inertial forces that causes its rotary motion. After collision the top region has velocity close to zero whereas the bottom region keeps moving in the original direction at velocities close to original. Due to this the ligament connecting the two drops is stretched. Upon reaching its limit, a pinch off is observed separating the smaller blob of liquid from the larger blob without the formation of any satellite droplets. The results are shown in figure 4.4

### 4.3.4 Offset ratio $x = 0.75$

Amongst all test cases studied, this case has the highest offset ratio of  $x = 0.75$ . This implies least resistance and maximum elongation of the bubbles. Upon collision, the smaller drop immediately merges with the larger one, but the point

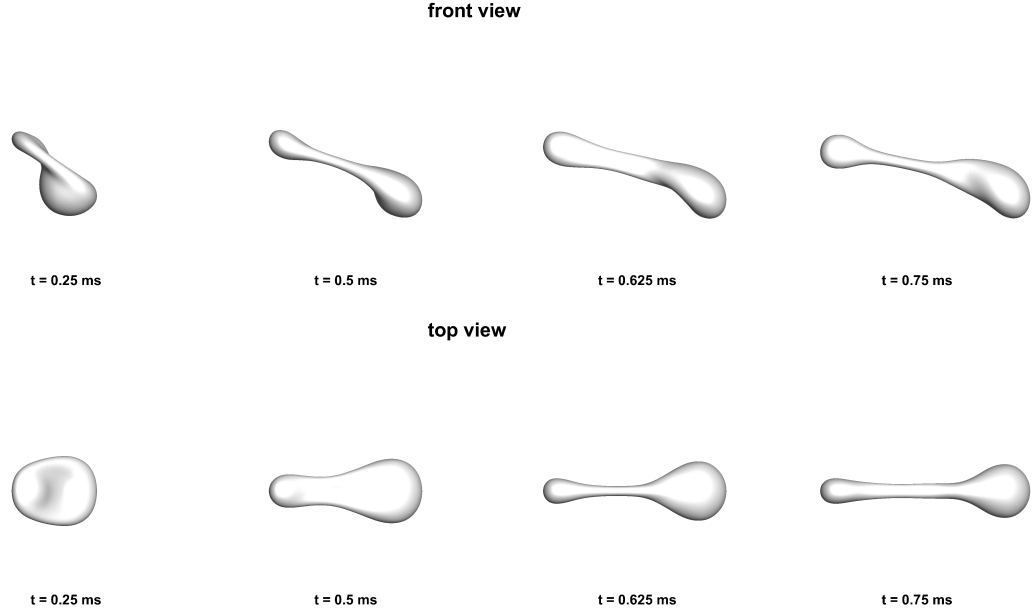


Figure 4.5: Contour plot for  $x=0.75$ : A

of impact or coalescence is shifted towards the upper end of the larger droplet because of the high offset ratio. As a result, after collision most part of the droplets retains its initial kinetic energy. Hence, lesser energy is available for lateral deformation. Hence, in this case, as seen in figure 4.4, the lateral expansion is less as compared to longitudinal elongation. As the time progresses, the droplets continue to move in opposite directions while still being connected by a small filament of the liquid. Since, both drops have enough energy to move on in opposite directions, the ligament is stretched on both sides due to the surface tension force from both the drops. At a certain point, the pinch off occurs causing the ligament to separate from both the drops. After pinching, the ligament of liquid, due to surface tension force, tries to achieve a configuration of minimum surface area. Due to this, the ligament tries to further break down into smaller droplets which are evident in the figure.

To conclude, as the region of interaction between the two droplets decreases, the blob of fluid become more and more prone to pinching and breakup. As the offset ratio was increased from 0 to 0.75, the region of interaction, defined as the overlapping region between the droplets along flow direction, kept on reducing. Hence absolutely no pinching was observed for offset ratio of 0 and near pinching

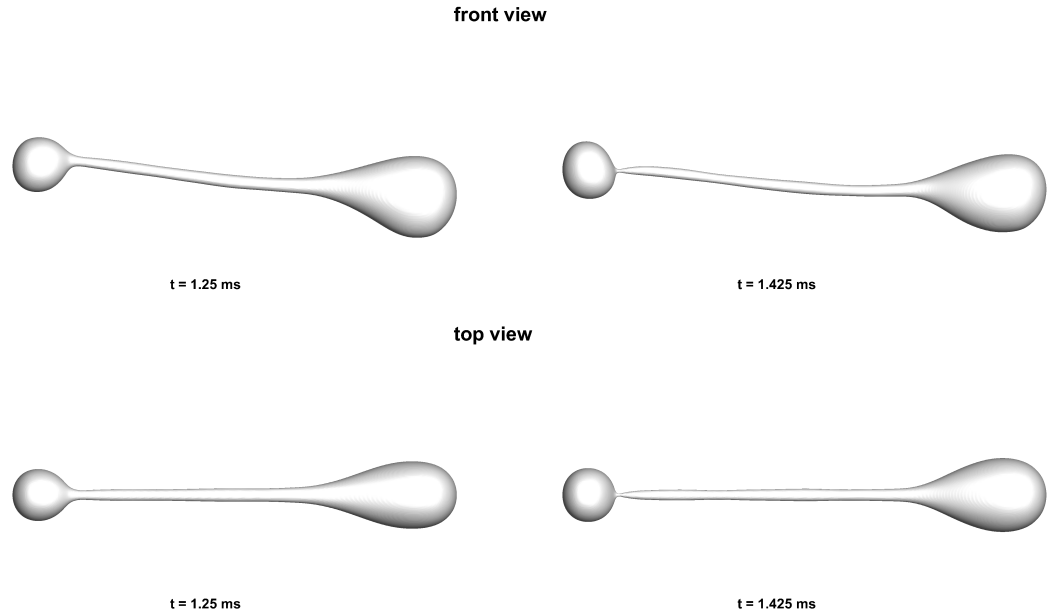


Figure 4.6: Contour plot for  $x=0.75$ : B

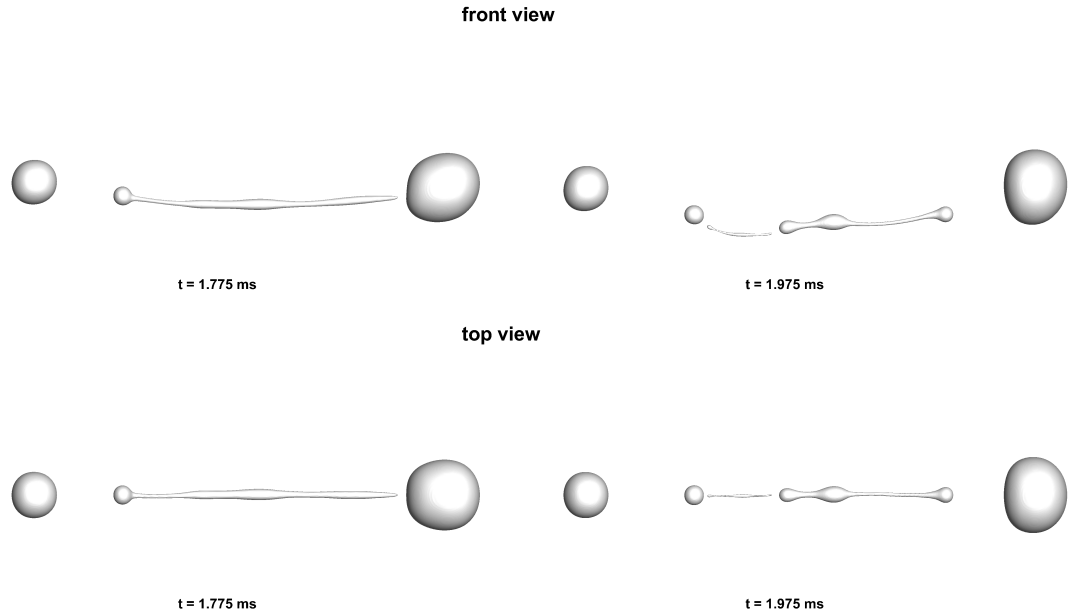
condition was obtained for offset ratio 0.25. For 0.5 offset ratio pinching with a single child droplet was observed and finally for the largest offset ratio of 0.75 pinching from both large droplets was observed resulting into ligament separation which further broke down to give child droplets.

## 4.4 Results for different diameter ratios and offset ratio = 0.5

Another important parameter in droplet collision studies is the diameters of colliding droplets. Hence, keeping the offset ratio fixed to 0.5, the diameter ratio of colliding droplets was varied from 1, 1.25 to 1.75. The effects are reported below:

### 4.4.1 Diameter ratio = 1

A diameter of 1 implies collision of two equal sized droplets. For a constant offset ratio of 0.5, a diameter ratio of 1 implies least interaction region for the droplets

Figure 4.7: Contour plot for  $x=0.75$ : C

as compared to diameter ratios of 1.25 , 1.5 and 1.75. Hence, relating to above observations one can predict a possibility of pinching and breakup. The same can be observed from figure 4.8. As soon as the droplets collide, immediate droplet coalescence is observed. Lateral expansion occurs to cope up with the kinetic energy of the impact. As the flattened disc shape is achieved, the surface tension forces start dominating and the disc starts to retract. Because of the energy gained due to retraction the droplet oscillates until a child droplet separates from the larger mass resulting into a long ligament and another huge blob of fluid. Since the ligament has insufficient surface area to hold the surface energy, it further breaks down into smaller droplets.

#### 4.4.2 Diameter Ratio = 1.25

The results for this off-center collision can be seen in figures 4.9,4.10 and 4.11. Upon collision, the droplets coalesce and resulting mass of fluid starts to expand along lateral direction. Since lot of kinetic energy is concentrated at the point of impact, the rapid shear deformation in this region as can be seen in 4.9. An oval void is created which immediately gets absorbed in the neighboring fluid due to surface tension. Because of off-center collision, both the masses of fluids continue

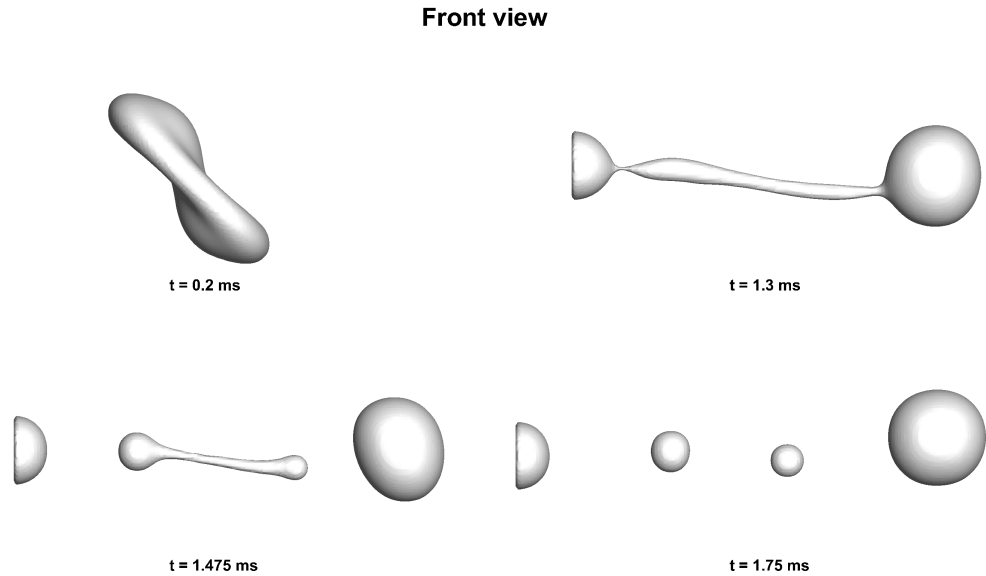


Figure 4.8: Contour plot for diameter ratio = 1

to move in opposite directions while still being connected by a thin ligament of fluid. As the two blobs move further apart, the ligament gets stretched to a point when pinch off takes place from the smaller blob. Further, the ligament also pinches off from the larger mass 4.10. The ligament now has larger surface area than the minimum required to hold that mass of fluid. Hence, the ligament further breaks down into smaller droplets as shown in figure 4.11.

#### 4.4.3 Diameter ratio = 1.75

This case has the largest region of interaction in between the droplets. Upon collision, the droplets coalesce and expand in lateral direction. Upon reaching a limit, the restoring surface tension forces retract the fluid mass. At this point, most of the energy is expended hence no pinching or droplet breakup is observed. The blob of fluid oscillates until the oscillations finally die out. Due to off-center collision an inertial couple sets the fluid in rotary motion. The results are shown in figure 4.12

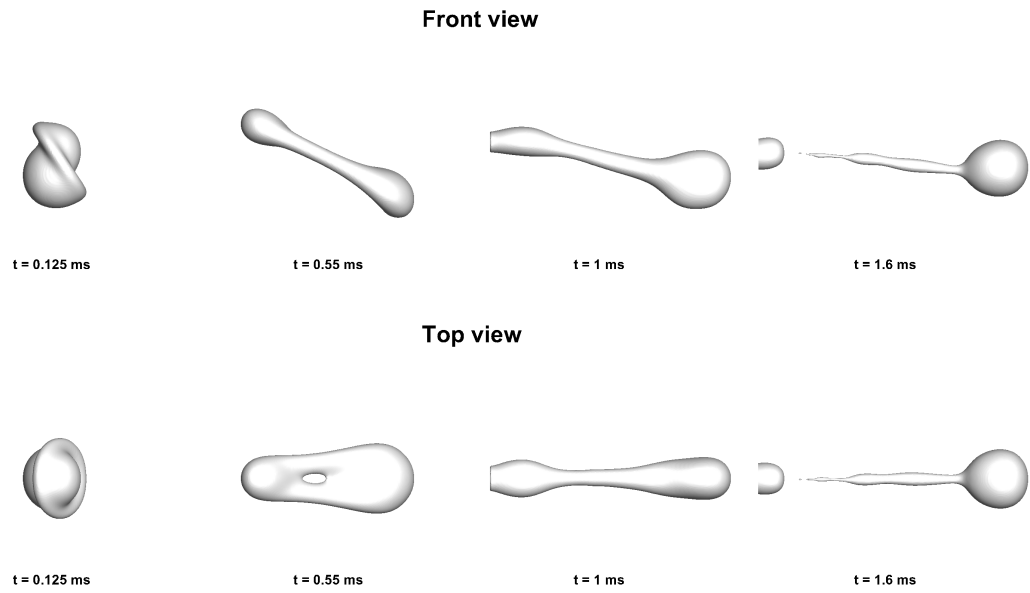


Figure 4.9: Contour plot for diameter ratio = 1.25

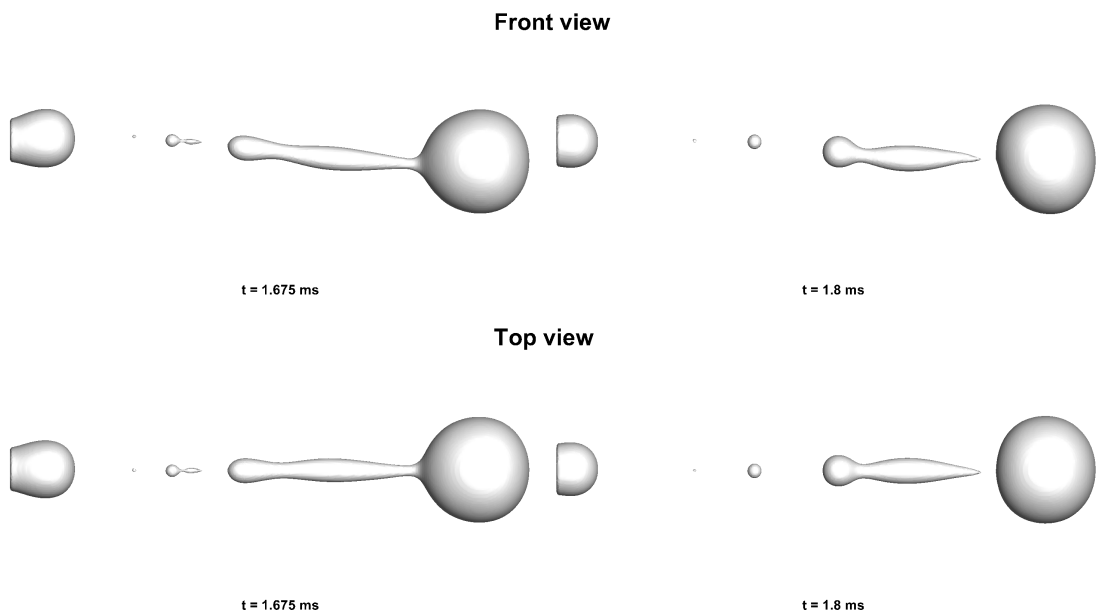


Figure 4.10: Contour plot for diameter ratio = 1.25

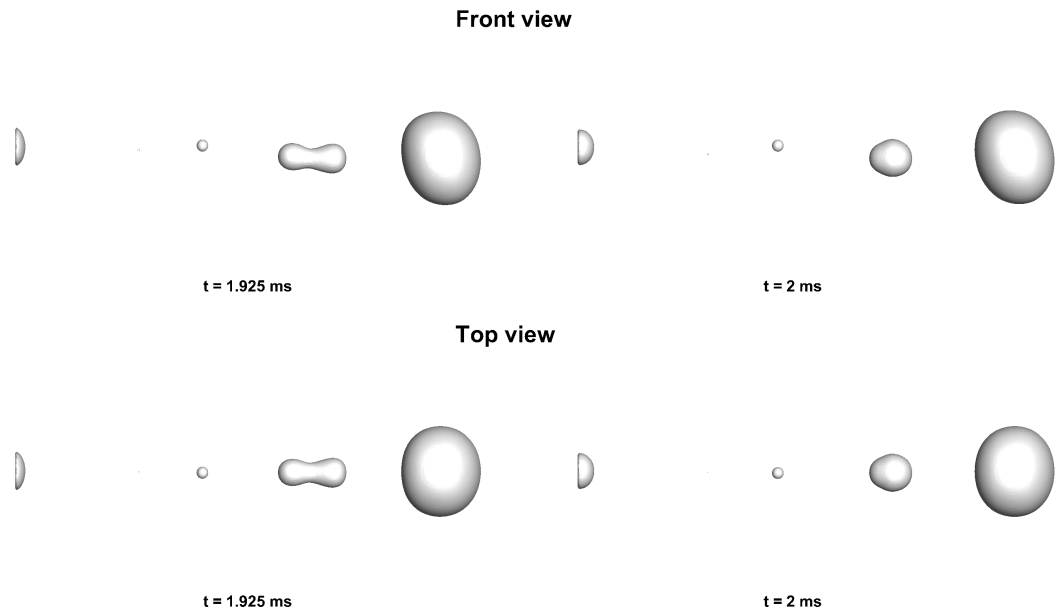


Figure 4.11: Contour plot for diameter ratio = 1.25

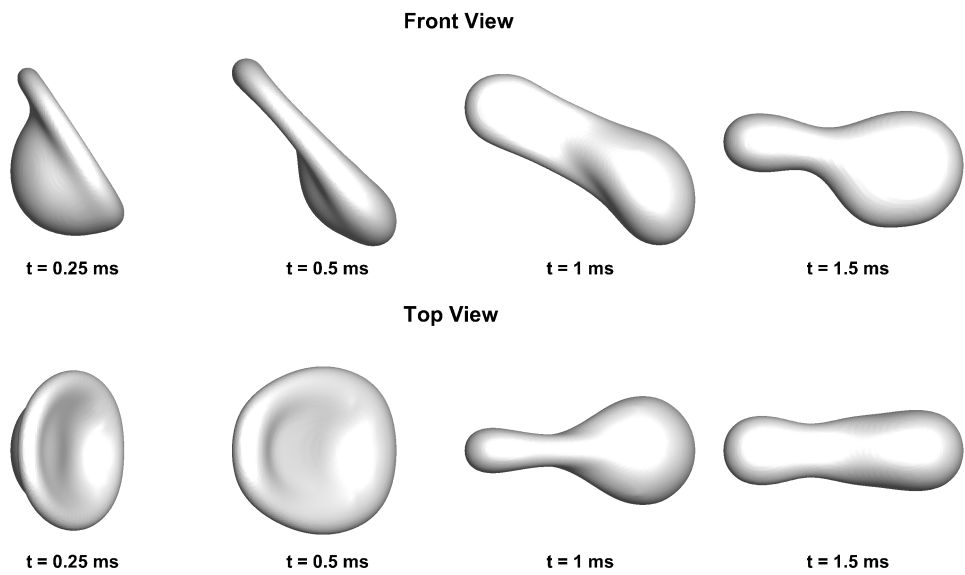


Figure 4.12: Contour plot for diameter ratio = 1.75

# Appendix A

## Exploring OpenFOAM

This section talks about the non-standard tests and some utilities native to OpenFoam that were explored during the course of this thesis. They include the following:

- Optimum choice of smoothers, preconditioners and solvers used to solve the U (velocity) and P (pressure) matrices.
- The time required to solve the computationally most expensive Pressure-Poissons equation for different combination of smoothers, preconditioners and smoothers.
- The speed up analysis when a large mesh is decomposed over different number of cores.
- Dynamic mesh refinement utility for multiphase flows.
- Refine mesh utility for multiphase flows.

### A.1 Smoothers, solvers and preconditioners:

The details about the choice of preconditioners, solvers and smoothers need to be specified in the system/fvSchemes file. A snapshot of a sample fvScheme file is given below:



```

P
{
  solver          PCG;
  preconditioner  DIC;
  tolerance       1e-6;
  relTol          0;
}
U
{
  solver          smoothSolver
  smooter         synGaussSeidel;
  tolerance       1e-6;
  relTol          0;
}

```

Figure A.1: Sample fvScheme dict

### A.1.1 Pre-conditioners:

The aim of any iterative solver is to solve the system of equation given by:

$$Ax = B$$

If M is any matrix such that the system:

$$M^{-1}Ax = M^{-1}B$$

has a faster convergence as compared to the original system, then M is known as a preconditioner. This leads to M (mostly) being an easily invertible approximation of A. All mathematical operations involving M, should be computationally cheap. The above system was an example of left preconditioning. Central and right preconditioning also exist. Given below is a list of the preconditioners available in OpenFOAM along with a brief explanation of their application. They are available in the *FOAM\_SRC/OpenFOAM/matrices/lduMatrix/preconditioners* directory:

diagonalPreconditioner - Diagonal preconditioner for both symmetric and asymmetric matrices. This preconditioner actually does not help with faster

propagation through the grid, but it is very easy and can be a good first step. Note: The reciprocal of the diagonal is calculated and stored for reuse because on most systems multiplications are faster than divisions.

DICPreconditioner - Simplified diagonal-based incomplete Cholesky preconditioner for symmetric matrices (symmetric equivalent of DILU). The reciprocal of the preconditioned diagonal is calculated and stored.

DILUPreconditioner - Simplified diagonal-based incomplete LU preconditioner for asymmetric matrices. The reciprocal of the preconditioned diagonal is calculated and stored.

FDICPreconditioner - Faster version of the DICPreconditioner diagonal-based incomplete Cholesky preconditioner for symmetric matrices (symmetric equivalent of DILU) in which the reciprocal of the preconditioned diagonal and the upper coefficients divided by the diagonal are calculated and stored.

GAMGPreconditioner - Geometric agglomerated algebraic multigrid preconditioner (also named Generalised geometric-algebraic multi-grid in the manual).  
noPreconditioner - Null preconditioner for both symmetric and asymmetric matrices.

### **A.1.2 Solvers:**

The word solver in OpenFOAM terminology is misleading. Solver, as specified in the system/fvSolution dictionary implies the linear solver used to solve the system of linear equations generated from the matrix equations 1 and/or 2. The list of solvers along with their brief application is given below:

BICCG - Diagonal incomplete LU preconditioned BiCG solver  
diagonalSolver - diagonal solver for both symmetric and asymmetric problems  
GAMG - Geometric agglomerated algebraic multigrid solver (also named Generalised geometricalgebraic multi-grid in the manual)  
ICC - Incomplete Cholesky preconditioned Conjugate Gradients solver

PBiCG - Preconditioned bi-conjugate gradient solver for asymmetric lduMatrices using a runtime selectable preconditioner

PCG - Preconditioned conjugate gradient solver for symmetric lduMatrices using a run-time selectable preconditioner

smoothSolver - Iterative solver using smoother for symmetric and asymmetric matrices which uses a run-time selected smoother

### A.1.3 Smoothers:

Although the preconditioners discussed before can considerably reduce the number of iterations, they do not normally reduce the mesh dependency of the numbers of iterations. OpenFOAM supplies the following smoothers to be used with the solvers in the smoothers/ directory:

DIC/ - Simplified diagonal-based incomplete Cholesky smoother for symmetric matrices.

DICGaussSeidel/ - Combined DIC/GaussSeidel smoother for symmetric matrices in which DIC smoothing is followed by GaussSeidel to ensure that any "spikes" created by the DIC sweeps are smoothed-out.

DILU/ - Simplified diagonal-based incomplete LU smoother for asymmetric matrices. ILU smoothers are good smoothers for linear multigrid methods.

DILUGaussSeidel/ - Combined DILU/GaussSeidel smoother for asymmetric matrices in which DILU smoothing is followed by GaussSeidel to ensure that any "spikes" created by the DILU sweeps are smoothed-out.

GaussSeidel/ - The GaussSeidel method is a technique used to solve a linear system of equations. The method is an improved version of the Jacobi method. It is defined on matrices with non-zero diagonals, but convergence is only guaranteed if the matrix is either diagonally dominant, or symmetric and positive definite.

Of all the solvers mentioned above, GAMG was of prime interest. Hence it was tested for a 0.1Million and 1 Million mesh for a lid-driven cavity case. The required to solve the pressure poisson equation was also noted. The sample parameters that need to be specified in the system/fvSolution while using GAMG as a solver are explained in Figure A.2

```

{
P
{
  solver      GAMG; % keyword used to call the solver
  tolerance   1e-06; % this tolerance is normalized for grid independence
  relTol      0.1; % ratio of residuals from two consecutive iterations
  smoother    GaussSeidel; % this actually operates on each level performing the task
  nPreSweeps  0; % sweeps to be performed while moving from finer to coarser mesh
  nPostSweeps 2; % sweeps to be performed while moving from coarser to finer mesh in V
cycle there is also nFinestSweeps for the finest layer
  cacheAgglomeration on;
  agglomerator faceAreaPair;
  nCellsInCoarsestLevel 10; % you can also specify the no of refinements (steps in V cycle)
  mergeLevels 1; %No of cells to merge during interpolation from finer to coarser or vice versa
}
U
{
  solver      smoothSolver;
  smoother    symGaussSeidel;
  tolerance   1e-05;
  relTol      0.1;
}
}

```

Figure A.2: Sample GAMG settings

Various combination of smoothers, solvers and preconditioners were analyzed to get the optimum case setup that ensures fast convergence. The analysis is presented below:

The mesh size used for above analysis was a 0.1 Million. The data in Table A.3 is presented in the form of a bar chart for better understanding in Fig 3. The problem attempted was a 3D lid driven cavity.

From the above analysis we conclude that when a finer mesh is chosen, the time required to solve the pressure Poisson equation goes up significantly as compared to the other operations. Also, the optimum case setup consists of DIC preconditioned GAMG solver with Gauss Siedel as smoother for pressure and a smooth solver with Gauss Siedel as smoother for Velocity.

1					3			
P		U			P		U	
solver	PCG	solver	BICCG		solver	GAMG (smoother = GaussSeidel)	solver	smoothSolver
preconditioner	none	preconditioner	None		preconditioner	none	smoother	GaussSeidel
total cpu time (s)		total pressure Poisson time (s)			total cpu time (s)		total pressure poisson time (s)	
1861.47		1396.6			1901.1		1470.77	
2					4			
P		U			P		U	
solver	PCG	solver	smoothSolver		solver	GAMG (smoother = GaussSeidel)	solver	smoothSolver
preconditioner	DIC	smoother	GaussSeidel		preconditioner	DIC	smoother	GaussSeidel
total cpu time (s)		total pressure Poisson time (s)			total cpu time (s)		total pressure poisson time (s)	
1557.03		1136.02			1462.9		1163.86	

Figure A.3: Different settings for *fvSchemes* file

The case setup for GAMG has lot of open ends that need to be addressed. They include, deciding upon `nCellsInCoarsestLevel` number of levels of coarsening defined by `nMaxLevels` in the `system/fvSolution` dictionary. The optimum value found for both was 100 and 1 respectively.

## A.2 Multicore speedup analysis:

The basic ideology behind parallel operation of any solver is that, a huge mesh is broken down into number of parts which are then solved simultaneously. This requires information to be transferred along the faces of the decomposed domain. The time saving obtained by operating the solver in parallel mode is limited by the time required for information transfer along the faces. Hence, a balanced has to be maintained. As a thumb rule, 1 lac cells per domain ensure a good speed up.

For the speed up analysis, same 3D lid driven cavity was chosen. The case setup parameters were kept constant for all the simulations. `decomposePar` utility in OpenFOAM was used for domain decomposition. Two types of decomposition methods are available in OpenFOAM, namely: `scotch` and `simple`. `Simple` is a geometric methods while `scotch` is more of an algebraic method. They have also been compared. The ideology behind `scotch` is same as the one mentioned above.

The above plot in figure A.6 concludes that `scotch` is a better decomposition

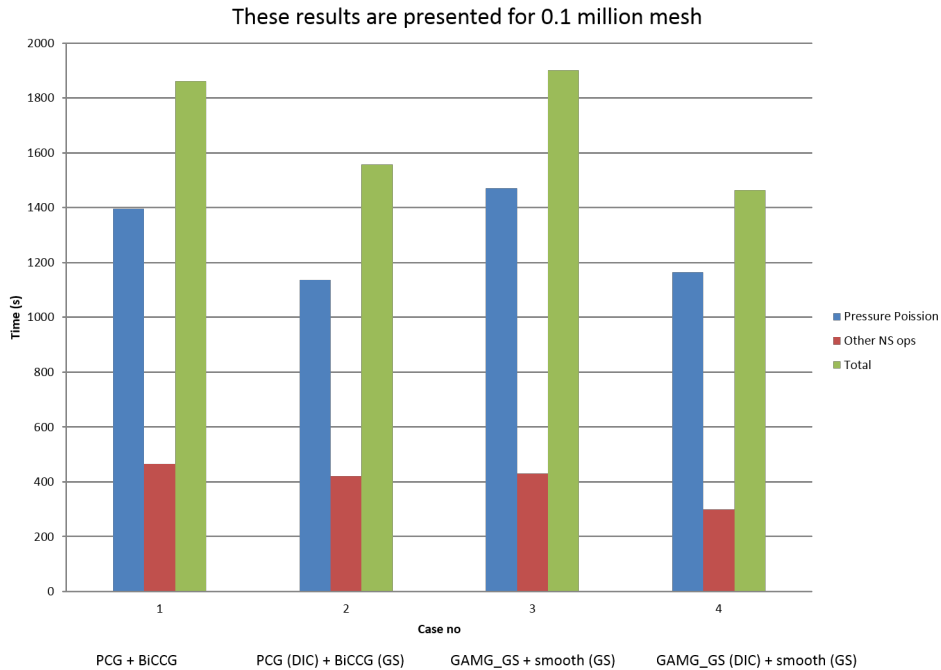


Figure A.4: Time consumption for 0.1 Million mesh

technique as compared to simple. Another important observation from the above plot is that if odd number of cores are chosen for decomposition we get a sudden speed down instead of a speed up. Yet using scotch along with even cores doesn't give satisfactory speedup.

A probable reason for loss of speedup was identified as lack of enough number of cells per core. Hence, with same case setup parameters, a 1 Million mesh was chosen. The results for the same are as reported below in figure A.7

### A.3 Dynamic mesh refinement:

Usually, in fluid flow problems, the actual phenomenon is localized in certain region while, most of the other part of domain is isolated. For such problems, usually a block mesh is chosen which is fine in the region that houses the actual phenomena while the other region still remains coarse.

This technique saves a lot of valuable computation time by reducing the overall mesh size. But, if the region of interest changes with time, then a statically refined block mesh doesn't solve the problem. The solution to such problems is

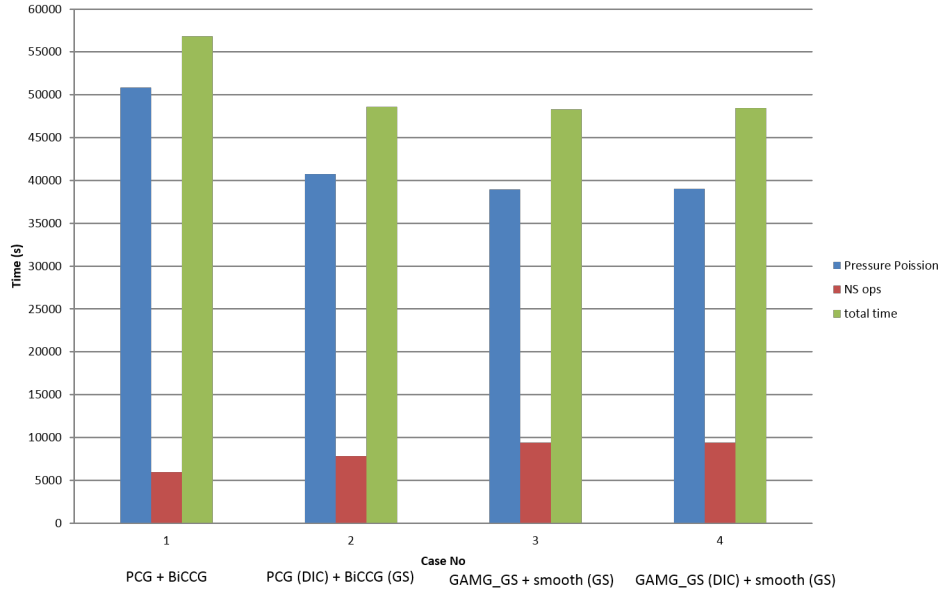


Figure A.5: Time consumption for 1 Million mesh

dynamic mesh refinement. This technique is especially useful while dealing with droplet and bubble dynamics.

In this technique, the mesh is refined based on a particular parameter. For most of the multiphase flow problems involving droplets and bubbles the volume fraction is the chosen parameter based on which the refinement is done. All the cells that have value lying between the specified limits are refined. Thus, the overall size of mesh is reduced drastically. However, there is an overhead in terms of interpolation of values from coarser to finer and vice versa. Hence the saving in time resulting from a smaller mesh may be counter balanced by the time required for refinement, coarsening and interpolation. Hence, optimum choice of parameters such as, number of levels of refinement, maximum number of refined cells and refinement interval is imperative to achieve maximum benefit from the method. An attempt was made to judge the dynamic mesh refinement capabilities of OpenFOAM. The 2D bubble rise problem for a density ratio of 10 and 1000 was chosen as a benchmark to validate the accuracy of dynamic meshing.

The contour plots and well as validation plots for bubble centroid vs. height are given below. The explicit solver rbsFoam was used to generate these results.

In the plot A.9, the legends rf1 and rf2 stand for mesh refinement level of 1

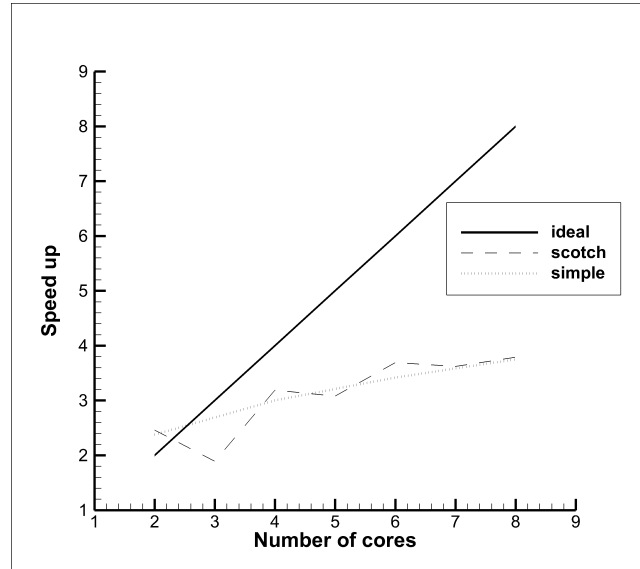


Figure A.6: Comparing decomposition techniques: scotch vs. simple

and 2 respectively. If any given cell is split in half along all the axes, then such refinement is attributed as 1 level of refinement. Accordingly, a single cell is split into 8 cells per level of refinement (for a 3D problem). Hence for a refinement level of 2, 1 cell from the base case will be split into 64 cells.

The plots in figure A.9 reveal that results for any level of refinement are almost same and match fairly well with the experimental results implying that the base mesh of 40x80 cells was good enough to obtain the desired results. Hence, for this particular case dynamic meshing is not advantageous since it increases the computational time with no gain in accuracy. Upon a closer look, it is observed that the results for refinement level of 2 have a wavy nature indicating instability that might have been introduced due to the interpolation errors. Though these errors are not so pronounced in the given case.

Now a more stringent and adverse test case of density ratio 1000 is considered.

From the above the plot in figure A.11, its evident that with dynamic mesh refinement there is a significant loss of accuracy. For rf2 with a courant number setting of 0.01 a wavy plot of significant amplitude is obtained confirming the reasoning provided for density ratio 10 case. The interpolation errors introduced



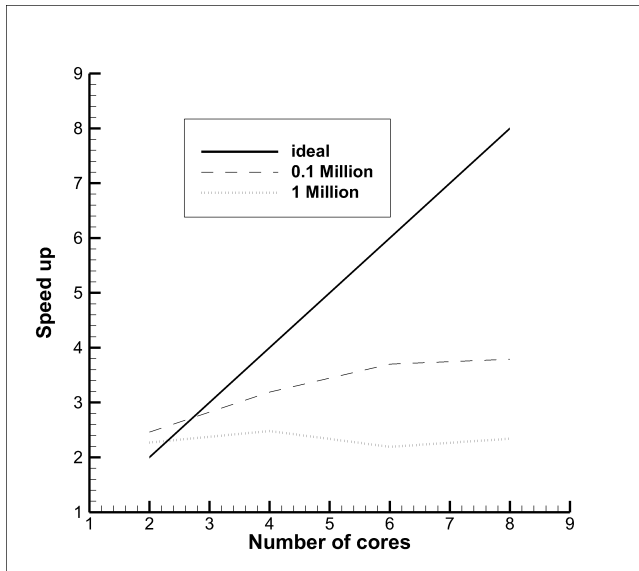


Figure A.7: Comparing decomposition techniques: scotch vs. simple

due to dynamic mesh refinement are more pronounced in the density ratio 1000 case. Though, if a lower courant number setting is used the solution appears to stabilize. Yet, there is a significant deviation from the reference results. To check the stability of implicit solver in such adverse test case, the above analysis was repeated using `interFoam`. The results are reported in figure A.12.

The plot A.12 confirms that implicit solver is more stable as compared to explicit one. Though, the results still differ significantly from the reference. With increasing level of mesh refinement, the deviation seems to increase as well. One of the reasons behind the deviation in results is the sudden interpolation that needs to be done from a coarse grid to fine grid near the interface at the initial time step. One way to avoid this is to initialize the volume fraction on a finer block as explained in the Figure A.13 below.

The finer block is so placed that during initialization, the entire geometry of phase 2 should lie within that box. This avoids the sudden numerical jerk that comes into play while interpolating from a very coarse grid to very fine grid and vice versa. Again the bubble rise test case for a density ratio of 1000 is considered to test the behavior of the bubbly numerically and compare the same with standard results.

The refinement level selected is 1. Hence during initial time steps, the inter-

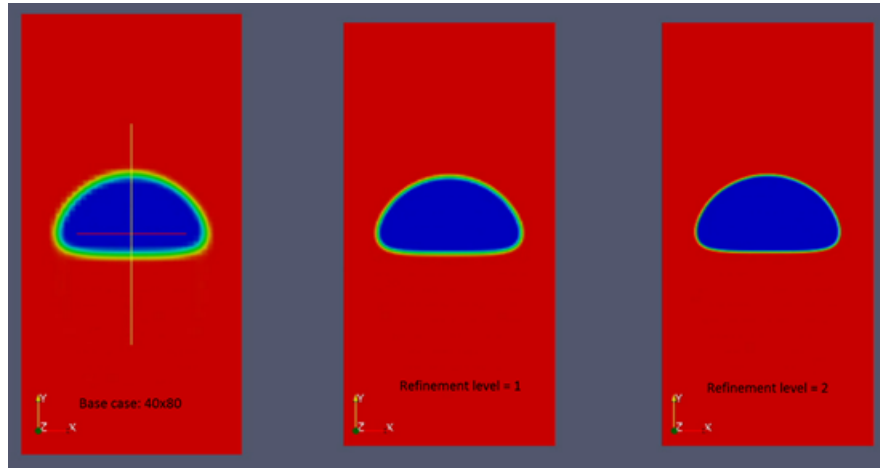


Figure A.8: Terminal Bubble positions for different levels of mesh refinement  $dr = 1000$

pulation is done only for a single refinement level. From the plots it is observed that the deviation begins at 0.5 sec. The contour plot reveals that the bubble leaves the finer domain at 0.5 sec. This is indicative of the fact that the source of error is interpolation from finer grid to coarser grid.

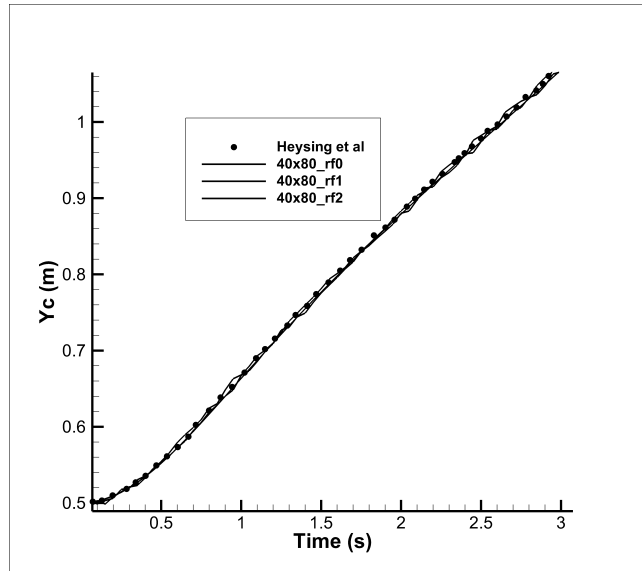


Figure A.9: Bubble centroid vs time for density ratio 10

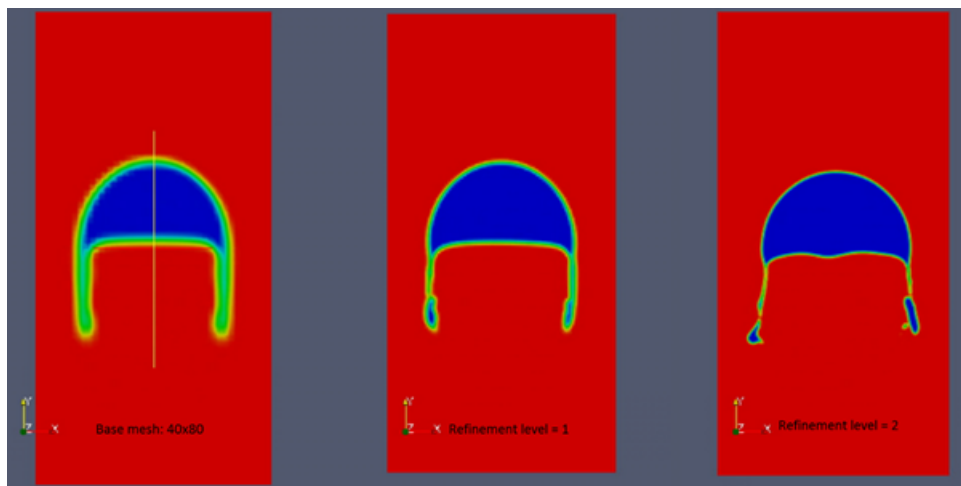


Figure A.10: Terminal Bubble positions for different levels of mesh refinement dr 1000

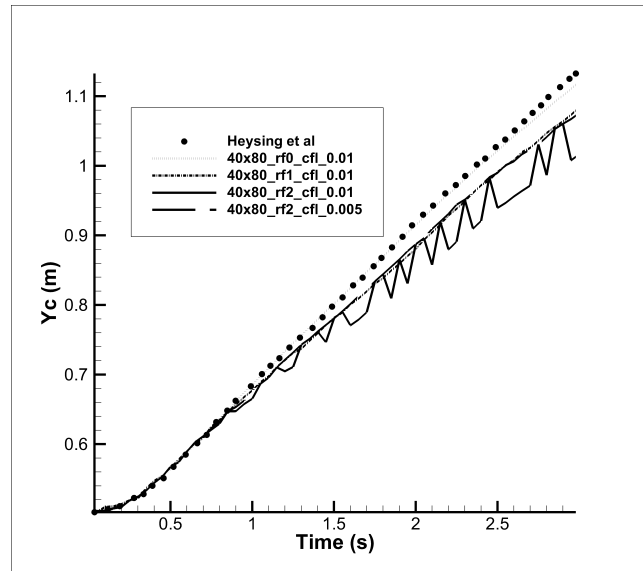


Figure A.11: Bubble centroid vs time for density ratio 1000

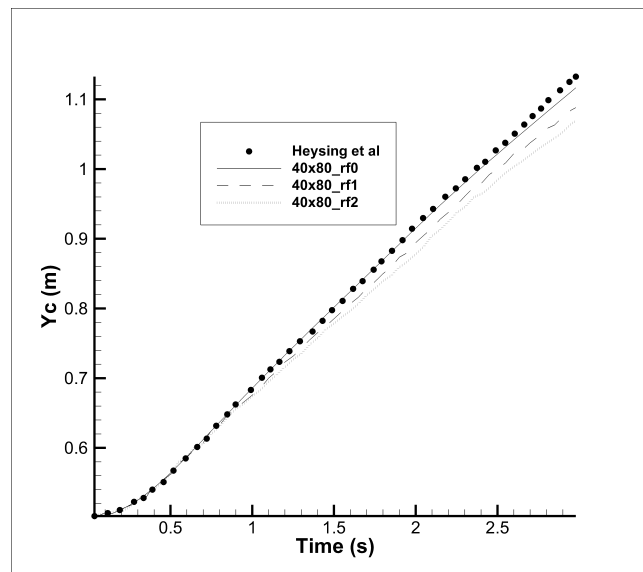


Figure A.12: Bubble centroid vs time for density ratio 1000

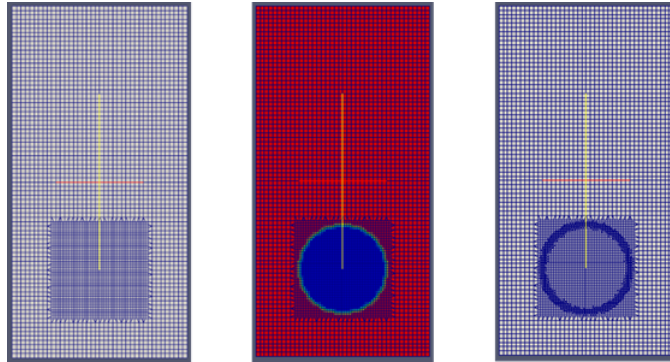


Figure A.13: Bubble centroid vs time for density ratio 1000

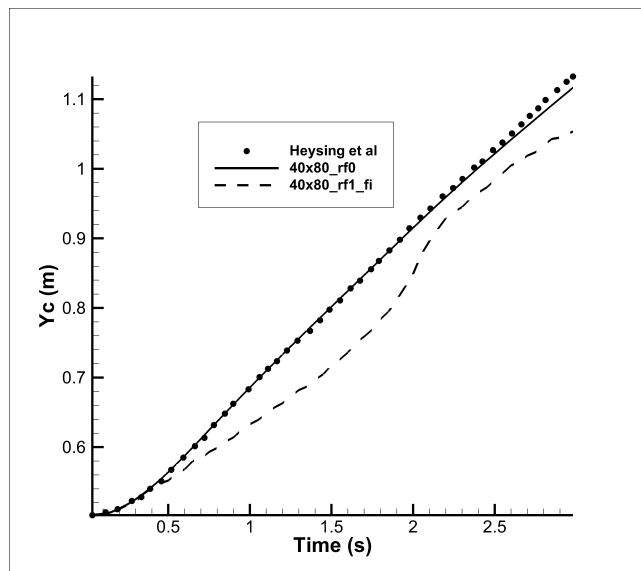


Figure A.14: Bubble centroid vs time for density ratio 1000

# References

- [1] BALDWIN, B.S. AND LOMAX H., “Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows”, *AIAA Paper 78-257.*, (1978)
- [2] J.R. Adam, N.R. Lindblad, C.D. Hendricks, *The collision, coalescence, and disruption of water droplets*, *J. Appl. Phys.* 39 (1968) 173.
- [3] R.W. Park, *Behavior of Water Drops Colliding in Humid Nitrogen*. PhD Thesis, Department of Chemical Engineering, The University of Wisconsin, 1970.
- [4] P.R. Brazier-Smith, S.G. Jennings, J. Latham, *The interaction of falling water drops: coalescence*, *Proc. R. Soc. Lond. A* 326 (1972) 393.
- [5] N. Ashgriz, J.Y. Poo, *Coalescence and separation in binary collision of liquid drops*, *J. Fluid. Mech.* 221 (1990) 183204
- [6] G.M. Faeth, *Current status of droplet and liquid combustion*, *Prog. Energy Combust. Sci.* 3 (1977) 191
- [7] P.J. ORourke, F.V. Bracco, *Modeling of droplet interactions in thick sprays and a comparison with experiments*, *Startified Charge Auto. Eng. Conf.*, 101115 *Inst. Mech. Eng. Pub.* ISMB 0-85298-4693
- [8] J. Qian, C.K. Law, *Regimes of coalescence and separation in droplet collision*, *J. Fluid Mech.* 331 (1997) 59.
- [9] Y.J. Jiang, A. Umemura, C.K. Law, *An experimental investigation on the collision behaviour of hydrocarbon droplets.*, *J. Fluid Mech.* 234 (1992) 171.

- [10] J.-P. Estrade, H. Carentz, G. Lavergne, Y. Biscos, *Experimental investigation of dynamic binary collision of ethanol droplets A model for droplet coalescence and bouncing*, Int. J. Heat Fluid Flow 20 (1999) 486
- [11] G. Brenn, D. Valkovska, K.D. Danov, *The formation of satellite droplets by unstable binary drop collisions*, Phys. Fluids 13 (2002) 24632477.
- [12] V.A. Arkhipov, I.M. Vasenin, V.F. Trofimov, *Stability of colliding drops of ideal liquid*, Zh. Prikl. Mekh. Tekh. Fiz. 3 (1983) 95
- [13] K.D. Willis, M.E. Orme, *Experiments on the dynamics of droplet collisions in a vacuum*, Exp. Fluids 29 (2000) 347358.
- [14] C. Arcoumanis and A. C. Enotiadis. *In-Cylinder Fuel Distribution in a Port-Injected Model Engine Using Rayleigh Scattering*. Experiments in Fluids 11, (1991) 375387
- [15] P. Miles and M. Dilligan. *Quantitative In-Cylinder Fluid Composition Measurements Using Broadband Spontaneous Raman Scattering*. In SAE Technical Paper. 1996
- [16] D. S. Choi, G. M. Choi, and D. J. Kim. *Spray Structures and vaporizing characteristics of a GDI fuel spray*. Journal of Mechanical Science and Technology 16, (2002) 9991008
- [17] D. L. Siebers. *Liquid-phase fuel penetration in diesel sprays*. Technical Report, SAE technical paper 1998.
- [18] J. Gao, D. Jiang, and Z. Huang. *Spray properties of alternative fuels: A comparative analysis of ethanol/gasoline blends and gasoline*. Fuel 86, (2007) 16451650
- [19] M. Bysveen, T. Almas, K. A. Ulvund, A. Jrgensen, and F. Kvinge. *Development of a shadowgraph image technique describing the fuel spray behavior in a rapid compression machine*. Technical Report, SAE Technical Paper 2004
- [20] J. K. Kim, K. Nishida, and H. Hiroyasu. *Characteristics of the internal flow in a diesel injection nozzle*. In ICALSS. 1997 175182.

- 
- [21] N. Tamaki, M. Shimizu, K. Nishida, and H. Hiroyasu. *Effects of cavitation and internal flow on atomization of a liquid jet*. *Atomization and sprays* 8, (1998) 179197.
- [22] D. C. *On the experimental investigation on primary atomization of liquid streams*. *Experiments in Fluids* 45, (2008) 371422
- [23] A. Vallet, A. Burluka, and R. Borghi. *Development of a Eulerian model for the atomization of a liquid jet*. *Atomization and sprays* 11, (2001) 619642.
- [24] Q. Xue, M. Battistoni, S. Som, S. Quan, P. Senecal, E. Pomraning, and D. Schmidt. *Eulerian CFD Modeling of Coupled Nozzle Flow and Spray with Validation Against X-Ray Radiography Data*. *SAE International Journal of Engines* 7, (2014) 10611072.
- [25] G. Blokkeel, X. Silvani, F. Demoulin, and R. Borghi. *An Eulerian model to improve the primary breakup modeling of atomizing jet*. *Zaragoza* 9, (2002) 11.
- [26] M. L. Corradini. *Fundamentals of multiphase flow*. University of Wisconsin, Madison, WI
- [27] N. Ashgriz. In *Handbook of atomization and sprays: theory and applications*. Springer
- [28] J. Fukai, Z. Zhao, D. Poulikakos, C. M. Megaridis, and O. Miyatake. *Modeling of the deformation of a liquid droplet impinging upon a flat surface*. *Physics of Fluids A: Fluid Dynamics (1989-1993)* 5, (1993) 25882599.
- [29] F. Raymond and J.-M. Rosant. *A numerical and experimental study of the terminal velocity and shape of bubbles in viscous liquids*. *Chemical Engineering Science* 55, (2000) 943955.
- [30] S. Heister. *Boundary element methods for two-fluid free surface flows*. *Engineering analysis with boundary elements* 19, (1997) 309317.



- [31] J. Glimm, J. W. Grove, X. L. Li, K.-m. Shyue, Y. Zeng, and Q. Zhang. *Threedimensional front tracking*. SIAM Journal on Scientific Computing 19, (1998) 703727.
- [32] J. Glimm, D. Marchesin, and O. McBryan. *A numerical method for two phase flow with an unstable interface*. Journal of Computational Physics 39, (1981) 179–200.
- [33] F. H. Harlow and J. E. Welch. *Numerical calculation of time dependent viscous incompressible flow of fluid with a free surface*. Physics of Fluids 8, (1965) 21822189.
- [34] C. W. Hirt and B. D. Nichols. *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries*. Journal of Computational Physics 39, (1981) 201225.
- [35] W. J. Rider and D. B. Kothe. *Reconstructing volume tracking*. Journal of computational physics 141, (1998) 112152.
- [36] M. Sussman, P. Smereka, and S. Osher. *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*. Journal of Computational Physics 114, (1994) 146–159.
- [37] S. Osher and R. P. Fedkiw. *Level Set Methods: An Overview and Some Recent Results*. Journal of Computational Physics 169, (2001) 463–502.
- [38] D. Jacqmin. *Calculation of Two-Phase NavierStokes Flows Using Phase-Field Modeling*. Journal of Computational Physics 155, (1999) 96–127.
- [39] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. *Diffuse interface methods in Fluid Mechanics*. Annu. Rev. Fluid Mech. 30, (1998) 139165.
- [40] S. O. Unverdi and G. Tryggvason. *front-tracking method for viscous, incompressible, multi-fluid flows*. Journal of Computational Physics 100, (1992) 25–37.
- [41] H. Udaykumar, R. Mittal, and W. Shyy. *Computation of solidliquid phase fronts in the sharp interface limit on fixed grids*. Journal of computational physics 153, (1999) 535574.

- [42] G. Tryggvason, R. Scardovelli, and S. Zaleski. *In Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge university press, 2011.
- [43] G. Tryggvason, A. Esmaeeli, J. Lu, and S. Biswas. *Direct numerical simulations of gas/liquid multiphase flows*. *Fluid Dynamics Research* 38, (2006) 660–681. Recent Topics in Computational Fluid Dynamics-Recent Topics in Computational Fluid Dynamics.
- [44] D. Fuster, A. Bague, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski. *Simulation of primary atomization with an octree adaptive mesh refinement and VOF method*. *International Journal of Multiphase Flow* 35, (2009) 550565.
- [45] D. L. Youngs. *Time-dependent multi-material flow with large fluid distortion*. *Numerical methods for fluid dynamics* 24, (1982) 273285.
- [46] N. Ashgriz and J. Poo. *FLAIR: Flux line-segment model for advection and interface reconstruction*. *Journal of Computational Physics* 93, (1991) 449468.
- [47] R. Scardovelli and S. Zaleski. *Interface reconstruction with least-square fit and split EulerianLagrangian advection*. *International Journal for Numerical Methods in Fluids* 41, (2003) 251274.
- [48] H. C. Ku, R. S. Hirsh, and T. D. Taylor. *A pseudospectral method for solution of the threedimensional incompressible Navier-Stokes equations*. *Journal of Computational Physics* 70, (1987) 439462.
- [49] W. Aniszewski , T. Mnard , M. Marek *Volume of Fluid (VOF) type advection methods in two-phase flow: A comparative study* *Computers and Fluids* 97 (2014) 5273
- [50] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan and L. Tobiska , *Quantitative benchmark computations of two-dimensional bubble dynamics* *Int. J. Numer. Meth. Fluids* 2009; 60:12591288
- [51] Rajesh Reddy, R. Banerjee *GPU accelerated VOF based multiphase flow solver and its application to sprays* *Computers and Fluids* 117 (2015) 287303.

- 
- [52] G.E. Cossali, G. Brunello, A. Coghe, M. Marengo *Impact of a single drop on a liquid film: experimental analysis and comparison with empirical models* Italian Congress of Thermofluid Dynamics UIT, Ferrara, 30 June-2 July 1999.
- [53] N. Nikolopoulos, A. Theodorakakos, G. Bergeles *Off-centre binary collision of droplets: A numerical investigation* International Journal of Heat and Mass Transfer 52 (2009) 41604174
- [54] Vatsalya Sharma *Development of Parallel CFD solver for Three Dimensional Unstructured Grid* Masters thesis, IIT Hyderabad, JULY 2014
- [55] Rajesh Reddy *Numerical Study of Liquid Sheet Atomization using Two-Phase Flow Solver on GPU Architecture* Doctoral thesis, IIT Hyderabad, JULY 2014
- [56] F. Xiao, M. Dianat, and J. J. McGuirk. *Large Eddy Simulation of Single Droplet and Liquid Jet Primary Breakup Using a Coupled Level Set/Volume of Fluid Method*. Atomization and Sprays 24, (2014) 281302