An Improved CFD Tool to Simulate Adiabatic and Diabatic Two-Phase Flows

THÈSE N^{O} 4776 (2010)

PRÉSENTÉE LE 7 SEPTEMBRE 2010

À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRANSFERT DE CHALEUR ET DE MASSE
PROGRAMME DOCTORAL EN ENERGIE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Bogdan Alexandru NICHITA

acceptée sur proposition du jury:

Prof. M. Parlange, président du jury
Prof. J. R. Thome, directeur de thèse
Dr B. Smith, rapporteur
Prof. A. Tomiyama, rapporteur
Prof. I. Zun, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2010

to my son Tiberiu Alexandru

Acknowledgments

This work has been carried out at Heat and Mass Transfer Laboratory (LTCM) at Ecole Polytechnique Fédérale de Lausanne under the supervision of Prof. John R. Thome.

I would like to thank Prof. John R. Thome for his support, encouragement and guidance during the whole research work. I would also like to thank to Prof. Zun for his support and fruitful discussions we had. Special thanks go to Prof. Marc Parlange, Prof. Akio Tomiyama and Dr. Brian Smith for their time in reading this manuscript, together with their constructive suggestions and remarks.

I would also like to thank all of my colleagues here at LTCM for the wonderful time we spent together. I would like to thank Dr. Stefano Nebuloni for the fruitful and motivating discussions we had during this time. I would like to thank Dr. Chin Lee Ong (Jeff), Gustavo Rabello dos Anjos, Dr. Jung Eung Park, Dr. Jackson Braz Marcinichen, Ricardo José da Silva Lima and Dr. Mathieu Habert for their friendship and support. I would also like to thank Monique Barraud Plummer, Lise von Gross, Salomé Liebenberg, Cecile Tavernay and Nathalie Matthey-de-l'Endroit for their help in solving quickly the administrative problems.

This research involved writing of about 20000 lines of C code. Since I am an autodidact in this area, I am heartily thankful to my brother-in-law Andrei Niculescu for the discussions we had during this time.

Last but not least, I would like to thank my family for their support and motivation and above all to my son Tiberiu Alexandru to whom this work is dedicated.

Trust in the Lord with all your heart, and do not lean on your own understanding. In all your ways acknowledge Him, and He will make your paths straight.

Proverbs 3, 5–6

Abstract

With increasing computer capabilities, numerical modeling of two-phase flows has developed significantly over the last few years. Although there are two main categories, namely ‘one’ fluid and ‘two’ fluid methods, the ‘one’ fluid methods are more commonly used for tracking or capturing the interface between two fluids. Level set (LS), volume-of-fluid (VOF), front tracking, marker-and-cell (MAC) and lattice Boltzmann (LB) methods are all ‘one’ fluid methods. It is clear that there is no perfect method; each method has advantages and disadvantages which make it more appropriate for one kind of problem than for others. For instance, a LS method will accurately compute the curvature and the normal to the interface, but tends to loss mass which is physically incorrect. On the other hand, a VOF method will conserve mass up to machine precision, but the computation of the curvature and normal to the interface is not as accurate. In order to minimize the disadvantages of these methods, several authors have used two or more methods together to model two-phase flows. This is the case for the CLSVOF (Couple Level Set Volume Of Fluid) method, where LS and VOF are coupled together in order to better capture the interface. In CLSVOF, the level set function is used to compute the interface curvature and normal to the interface, while the volume of fluid function is used to capture the interface.

For two-phase flows in microchannels, surface tension forces play an important role in determining the dynamics of bubbles whereas gravitational forces are generally negligible. Also it is very important to consider the interaction between the boundaries and the fluids by prescribing or computing the correct contact angle between them. The commercial CFD code FLUENT allows the use of static constant angles, or the use of User Defined Functions (UDF) to compute the dynamic contact angles. It is inappropriate to use a static contact angle to model cases involving moving contact lines. For such cases a dynamic contact angle scheme should be implemented.

In this study, FLUENT was used to model adiabatic and diabatic, time dependent two-phase flows. Since FLUENT already contains a VOF method, a LS method was implemented and coupled with VOF into FLUENT via UDFs. Furthermore, since the LS function, used to compute the surface tension force, ceases to be a signed distance to the interface even after one time step, a re-initialization equation was solved after each time step. This involved using a fifth order WENO (Weighted Essentially Non Oscillatory) scheme to discretize the space derivatives (otherwise oscillations of the interface occurred), and a first order Euler method for the time integration. In another part of the study, a 3D dynamic contact angle model based on volume fraction, interface reconstruction, and experimentally available advancing and receding static contact angles was also developed and implemented into FLUENT via UDFs.

Several validations for the developed CLSVOF method and dynamic contact angle model are presented in this thesis, these includes a static bubble, a bubble rising in a stagnant liquid for Morton numbers ranging from 10^2 to 10^{-11}, droplet deformation due to a vortex flow field, droplets spreading over a wall under the gravity effect and droplets sliding over a wall due to gravity. These validations demonstrated the high accuracy and the stability of our methods for modeling these phenomena.

A heat and mass transfer model was also implemented into the commercial CFD code FLUENT for simulating of boiling (and condesation) heat transfer. Several simulations were presented with water and R134a as working fluids. The influence of the contact angle and the wall superheat was also studied.

Key words: level set, volume-of-fluid, WENO, FLUENT, dynamic contact angle, heat transfer

Version abrégée

Grâce à l'augmentation des capacités informatiques, la modélisation numérique des écoulements diphasiques s'est développée de manière significative au cours des dernières années. Bien qu'il existe deux catégories principales, à savoir les méthodes “un” fluide et “deux” fluides, les méthodes “un” fluide sont plus communément utilisés pour le suivi ou la capture de l'interface entre deux fluides. Les méthodes dites Level Set (LS), Volume-Of-Fluid (VOF), Front Tracking, Marker-And-Cell (MAC) et Lattice Boltzmann (LB) sont toutes des méthodes “un” fluide. Il apparaît clairement qu'il n'existe pas de méthode parfaite, chaque méthode ayant ses avantages et ses inconvénients qui la rendent plus appropriée pour un type de problème que pour un autre. Par exemple, si une méthode LS permet de calculer avec précision la courbure et la normale à l'interface, elle a tendance à perdre de la masse ce qui la rend physiquement inexacte. Par contre, une méthode VOF va conserver la masse avec précision, mais le calcul de la courbure et de la normale à l'interface n'est pas aussi précis. Afin de réduire au minimum les inconvénients de ces méthodes, plusieurs auteurs ont utilisé simultanément deux ou plusieurs méthodes pour modéliser les écoulements diphasiques. C'est le cas de la méthode CLSVOF (Couple Level Set Volume Of Fluid), qui couple les méthodes LS et VOF afin de mieux capturer l'interface. Dans la méthode CLSVOF, le Level Set est utilisé pour le calcul de la courbure et de la normale à l'interface, tandis que la fonction Volume of Fluid fonction est utilisée pour capturer l'interface.

Pour les écoulements diphasiques en micro-canaux, les forces de tension superficielles jouent un rôle prépondérant dans la détermination de la dynamique des bulles alors que les forces gravitationnelles sont généralement négligeables. Il est alors très important de tenir compte de l'interaction entre les limites et les fluides en calculant de manière correcte l'angle de contact entre les deux. Le code CFD commercial FLUENT permet l'utilisation d'angles constants statiques, ou encore le recours à des fonctions définies par l'utilisateur (UDF) pour calculer les angles de contact dynamique. Il est inapproprié d'utiliser un angle de contact statique pour modéliser les cas impliquant le déplacement des lignes de contact. Pour de tels cas, un modéle d'angle de contact dynamique doit être implémenté.

Dans cette étude, FLUENT a été utilisé pour modéliser des écoulements diphasiques adiabatiques et diabatiques en régime non-stationnaire. Comme FLUENT inclut déjà une méthode VOF, une méthode LS a été implémentée et couplée dans FLUENT avec VOF via des fonctions UDF. De plus, comme la fonction LS, utilisée pour le calcul de la tension de surface, cesse d'être une distance assignée à l'interface même après un pas de temps, une ré-initialisation de l'équation a été résolue après chaque pas de temps. Cela est réalisé à l'aide d'un schéma WENO (Weighted Essentially Non Oscillatory) du cinquième ordre pour discrétiser l'espace dérivés (autrement oscillations de l'interface s'est produite), et une méthode d'Euler au premier ordre pour l'intégration temporelle. Dans une autre partie de l'étude, un modèle 3D dynamique d'angle de contact basé sur une fraction de volume, une reconstruction d'interface, et des angles statiques obtenus expérimentalement a également été élaboré et mis en œuvre dans FLUENT via des fonctions UDF.

Plusieurs validations pour la méthode CLSVOF développée et pour le modèle dynamique d'angle de contact sont présentés dans cette thèse, comprenant une bulle statique, une bulle montant dans un liquide stagnant pour des nombres de Morton allant de 10^2 à 10^{-11}, la déformation d'une goutte par un champ tourbillonnaire, la diffusion de gouttes sur un mur avec l'effet de la gravité et le ruissellement de gouttes sur un mur par gravité. Ces validations démontrent la haute précision et la stabilité de nos méthodes de modélisation lors de la modélisation de ces phénomènes.

Un modéle de transfert de chaleur et de masse a été également mis en œvre dans le code CFD commercial FLUENT pour la simulation de vaporization (et condensation). Plusieurs simulations ont été présentées avec l'eau et le R134a comme fluides de travail. L'influence de l'angle de contact et de surchauffe des murs a été étudiée.

Mots-clés: Level Set, Volume-Of-Fluid, WENO, FLUENT, angle de contact dynamique, transfert de chaleur.

Contents

Acknowledgments **v**

Abstract **vii**

Version abrégée **ix**

List of Figures **xv**

List of Tables **xix**

Nomenclature **xxi**

1 Introduction **1**

2 State-of-the-art-review **5**

2.1 'One fluid' methods 5

2.1.1 Level set methods 5

2.1.2 Volume of fluid and interface capturing methods 13

2.1.3 Coupled level set / volume of fluid method 23

2.2 Surface tension models 25

2.3 Heat transfer with phase change 27

2.4 Contact angle 30

2.5 Conclusions 33

3 Theoretical models **37**

3.1 The level set method 37
  3.1.1 Generalities 37
  3.1.2 The conservation equations 37
  3.1.3 Re-initialization 39
3.2 The volume of fluid method 40
3.3 Dynamic contact angle 43
  3.3.1 Mathematical formulation 43
  3.3.2 The contact angle model 44
3.4 Heat transfer with phase change 46
3.5 Summary 47

4 FLUENT discretization procedure and code development 49

4.1 Fluent discretization description and choices 49
  4.1.1 Pressure-based algorithm 50
    Segregated algorithm 50
    Coupled algorithm 50
  4.1.2 Density-based algorithm 51
  4.1.3 Discretization procedure 51
    First order upwind scheme 52
    Power-law scheme 53
    Second-order upwind scheme 53
    QUICK scheme 54
    MUSCL scheme 54
  4.1.4 Multigrid solver 55
  4.1.5 Convergence 55
  4.1.6 Under-relaxation 56
4.2 Code development 56
  4.2.1 LS implementation 56

4.2.2 Dynamic contact angle implementation 58

4.2.3 Heat transfer model implementation 59

4.3 Novel aspects of the new code 59

5 Numerical results 61

5.1 Numerical validations of the CLSVOF method 61

5.1.1 Verification cases 61

5.1.2 Static bubble 66

5.1.3 Droplet deformation due to a vortex velocity field 68

5.1.4 Inviscid gas bubble 73

5.1.5 Rayleigh-Taylor instability 73

5.1.6 Bubble rising in a stagnant liquid 79

Air water-sugar simulations. 79

Air-oil simulations 88

Air-water simulations. 88

5.2 Numerical validations with dynamic contact angle model 103

5.2.1 2D examples 103

5.2.2 3D examples 111

5.3 Heat transfer simulations 113

5.4 Film thickness simulations 121

6 Conclusions 125

6.1 Future work 127

Appendices 129

A The WENO scheme 129

B The computation of the plane (line) constant α 133

B.1 Three-dimensional case 133

B.2 Two-dimensional case 135

C Pressure-velocity coupling algorithms 137

C.1 The SIMPLE algorithm 137

C.2 The SIMPLER algorithm 139

C.3 The SIMPLEC algorithm 140

C.4 The PISO algorithm 140

Predictor step 140

Corrector step 1 140

Corrector step 2 141

D List of scientific publications, presentations and conferences 143

E Bibliography 145

Curriculum Vitae 157

List of Figures

2.1 Micro and macro regions [1] 28

3.1 The exact VOF color function for a circular arc over a square grid. 41

3.2 Interface reconstruction schemes. 41

3.3 Lagrangian advection of the interface along the x-direction: (a) interface reconstruction in the central cell before advection; (b) contributions along the x-direction of the central cell after advection. From [2] 43

3.4 Equilibrium contact line and contact angle. 44

4.1 Pressure-based solution methods: (left) segregated algorithm and (right) coupled algorithm. 51

4.2 Density-based solution algorithm 52

4.3 A typical control volume. 53

4.4 One-dimensional control volume 54

4.5 A typical non-orthogonal control volume from [3]. 57

5.1 LS-VOF comparison. Dashed contour–LS function. Continuous contour–VOF function. 62

5.2 The level set function after solving the re-initialization equation. 63

5.3 The level set function without solving the re-initialization equation. 64

5.4 Level set contour (red) and volume-of-fluid contour (green) after solving the coupling equation between LS and VOF. 65

5.5 Level set contour (red) and volume-of-fluid contour (green) without coupling between LS and VOF (with large loss of mass). 65

5.6 Static bubble parasitic currents for 2D CLSVOF and VOF. 67

5.7 Interface location for a 2D droplet deformed by a vortex, with T=6s and a grid of 128x128. 70

5.8 Interface location for a 2D droplet deformed by a vortex with T=6s and a grid of 256x256. 71

5.9 Interface location for a 2D droplet deformed by a vortex with T=6s and a grid of 512x512. 72

5.10 Interface position for an inviscid, axisymmetric gas bubble rising in liquid; 128x256 grid. 74

5.11 Interface position for an inviscid, axisymmetric gas bubble rising in liquid from Sussman [4]. 75

5.12 Interface evolution for Rayleigh-Taylor instability at different times on a grid of 64x256 mesh points. 76

5.13 Interface evolution for Rayleigh-Taylor instability at different times on a grid of 128x512 mesh points. 77

5.14 Interface evolution for Rayleigh-Taylor instability at different times on a grid of 256x1024 mesh points. 78

5.15 Bubble rise velocity as a function of time with CLSVOF and VOF. 82

5.16 Interface position after t=0s, 0.6s and 1.2s, with CLSVOF and VOF for Morton number Mo=850. 83

5.17 Interface position after t=0s, 0.4s and 0.8s, with CLSVOF and VOF for Morton number Mo=41.1. 84

5.18 Interface position after t=0s, 0.4s and 0.75s, with CLSVOF and VOF for Morton number Mo=1.31. 85

5.19 Dimensionless height and width of the bubble. 86

5.20 Drag coefficient versus Reynolds number. 86

5.21 Bubble contour and velocity profile for Morton number 5.31. 87

5.22 Predicted bubble shape by CLSVOF for air bubble rising in clear mineral oil. . . . 89

5.23 Characteristic bubble path with 3D CLSVOF and VOF. 90

5.24 Interface position after t=0s, 0.05s, 0.1s, 0.15s, 0.2s, 0.25s, 0.30s, 0.35s, 0.40s, 0.45s and 0.5s, for an air bubble rising in water with CLSVOF and VOF. 92

5.25 Pressure contour for 5mm bubble with CLSVOF, time=0.35s, 0.37s. 94

5.26 Pressure contour for 5mm bubble with VOF, time=0.35s, 0.37s. 95

5.27 Pressure contour for 5mm bubble with CLSVOF, time=0.455s, 0.5s. 96

5.28 Pressure contour for 5mm bubble with VOF, time=0.455s, 0.5s. 97

5.29 Interface position and velocity field with 3D CLSVOF for 5mm bubble. 98

5.29 Interface position and velocity field with 3D CLSVOF for 5mm bubble, cont. . . 99

5.30 Interface position and velocity field with 3D VOF for 5mm bubble. 100

5.30 Interface position and velocity field with 3D VOF for 5mm bubble, cont. 101

5.31 A comparison between bubble rise velocity obtained experimentally by Zun and Groselj [5] and bubble rise velocity obtained with CLSVOF and VOF. 102

5.32 A comparison between bubble rise velocity obtained with CLSVOF, VOF and bubble rise velocity predicted by Wallis [6]. 102

5.33 Snapshots of the 2D water drop for a grid of 80x25 and gravity vector (0.0,-9.81). 104

5.34 Snapshots of the 2D water drop for a grid of 160x50 and gravity vector (0.0,-9.81).105

5.35 Snapshots of the 2D water drop for a grid of 320x100 and gravity vector (0.0,-9.81).106

5.36 Snapshots of the 2D bubble contour for a grid of 80x25 and gravity vector (-9.81,0).107

5.37 Snapshots of the 2D bubble contour for a grid of 160x50 and gravity vector (-9.81,0).108

5.38 Snapshots of the 2D bubble contour for a grid of 320x100 and gravity vector (-9.81,0). 109

5.39 Velocity field near the contact line. 110

5.40 Snapshots of the 3D bubble contour for a grid of 160x50x160 and gravity vector (0,-9.81,0.0) in plane z=0.04m. 111

5.41 Snapshots of the 3D bubble contour for a grid of 160x50x160 and gravity vector (-9.81,0.0,0.0) in plane z=0.04m. 112

5.42 Initial condition for boiling inside microchannels. 113

5.43 Bubble contour for boiling of water in microchannels. 114

5.44 3D bubble contour with dry patches at the lateral walls. 114

5.45 Bubble contour for boiling of R134a in microchannels 115

5.46 Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 40°. 116

5.47 Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 54°. 117

5.48 Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 80°. 118

5.49 Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with 15K wall superheat with a contact angle of 54°. 119

5.50 Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with 20K wall superheat with a contact angle of 54°. 120

5.51 Bubble contour for the air bubble flowing inside 1mm tube filled with ethanol. . 123

5.52 Velocity field near the rear of the middle air bubble. 123

5.53 Film thickness along the middle air bubble. 124

A.1 The three substencils: (a) the left-biased stencil; (b) the right-biased stencil. . . . 129

B.1 The "cut volume" of a 3D rectangular cell. 133

B.2 The "cut volume" of a 2D rectangular cell. 135

C.1 The SIMPLE algorithm from [7]. 138

C.2 The SIMPLER algorithm of Patankar [8] form [7]. 139

C.3 The PISO algorithm from [7]. 142

List of Tables

2.1 Advantages and disadvantages of ‘one fluid’ methods 35

5.1 Physical properties of the fluids. 66

5.2 Spurious currents for static bubble after 1.0s. 66

5.3 Droplet coordinates after one period of shear flow with CLSVOF. 69

5.4 CLSVOF, VOF and experimental data for an air bubble rising in a water-sugar solution. 80

5.5 2D-axisymmetric bubble rise velocity with CLSVOF compared to Bhaga and Weber [9] data. 81

5.6 Physical properties of mineral oil and air. 88

5.7 2D-axisymmetric bubble rise velocity with CLSVOF compared to Hnat and Buckmaster [10] data. 88

5.8 Physical properties of the fluids. 89

5.9 Bubble rise velocity and bubble mean lateral displacement for different surface tensions. 91

5.10 Physical properties of the fluids for different diameters. 93

5.11 Bubble rise velocities vs. data of Zun [5]. 98

5.12 Bubble rise velocities vs. data of Peebles and Garber [11]. 98

5.13 Bubble rise velocities vs. theory of Wallis [6]. 100

5.14 Error between successive mesh refinements for 2D air-water simulations with gravity vector (0,-9.81) and (-9.81,0). 103

5.15 Physical properties of water and R134a 113

5.16 A comparison between film thickness predicted by correlations and film thickness obtained by simulations for air bubbles flowing in ethanol. 121

Nomenclature

Roman Letters

$\vec{A}$	face area vector	$[m^2]$
c_p	specific heat	[J/kgK]
Eo	Eötvös number	[-]
F_b,f_P	body force	$[N/m^3]$
F	volume fraction	[-]
$\mathbf{g}$	gravity vector	$[m/s^2]$
h_{ev}	heat transfer coefficient	$[W/m^2K]$
h_{fg}	latent heat	[J/kg]
H	Heaviside function	[-]
I	interpolation coefficient	
k	thermal conductivity	[W/mK]
k	mean curvature of the interface	[1/m]
LS	level set	
M	molecular mass	[kmol/kg]
Mo	Morton number	[-]
N_{faces}	number of faces enclosing the cell	
Nf	Viscosity dimensionless number	[-]
$\mathbf{n}$	unit normal vector to the interface	[-]
Pe	Peclet number	[-]
$\hat{R}$	universal gas constant	[J/kmolK]

$\vec{r}$	direction vector	[m]
Re	Reynolds number	[-]
S_ϕ	Source of ϕ per unit volume	
t	time	[s]
u	x velocity	[m/s]
$\mathbf{u}$, $\vec{v}$, $\vec{V}$	velocity vector	[m/s]
V	cell volume	[m^3]
v	y velocity	[m/s]
VOF	volume of fluid	
w	z velocity	[m/s]
W_k	total kinetic energy	[J]
$\mathbf{x}$	position vector	[m]
x	x Cartesian coordinate	[m]
y	y Cartesian coordinate	[m]
z	z Cartesian coordinate	[m]

Greek Letters

$\delta x \, \delta y$	components of the vector pointing between FV corner	
Δ	vector pointing between two adjacent cells	
δ	Dirac function	
δ	film thickness	[m]
ϵ	half of the interface thickness	[m]
ϵ	relative error	[-]
ϕ	the level set function	[m]
Γ_ϕ	diffusion coefficient of ϕ	
μ	dynamic viscosity	[Pas]
ρ	density	[kg/m^3]
σ	surface tension coefficient	[N/m]

θ	contact angle	[°]
θ	weighted factor	

Subscripts

b	bottom face
CD	central differences
E	east cell
e	east face
g	gas
l	liquid
n	north face
nb	neighboring cells
P	current cell
s	south face
SOU	second order upwind
st	surface tension
t	top face
W	west cell
w	west face

Chapter 1

Introduction

Over the last 20 years, various methods have been developed for the numerical simulation of two-phase flows. We can divide these methods into two main categories: ‘one’ and ‘two’ fluid methods. In ‘one’ fluid methods, a single set of conservation equations is solved and the interface between the two fluids is tracked or captured. On the other hand, in ‘two’ fluid methods, a set of conservation equations is solved for each phase and the interaction between phases is given by correlations. In the modeling of two phase flows, ‘one’ fluid methods are more widely used than the ‘two’ fluid methods. Focusing on ‘one’ fluid methods, two types of approach are used to compute interfacial motion: interface tracking and interface capturing methods. The main difference between these methods is that interface tracking is Lagrangian, while interface capturing is Eulerian.

In particular, interface tracking techniques explicitly track the interface with marker particles. Such techniques include MAC methods [12], arbitrary Lagrangian-Eulerian (ALE) methods [13–15], and front tracking methods [16, 17]. These methods are very accurate and efficient for flexible moving boundaries with small deformations. The main drawback is that they are difficult to use in cases where the interface breaks up or coalesces with another interface. Also additional re-meshing is needed when a large deformation of the interface occurs.

In interface capturing methods, instead, an auxiliary function is needed. These methods are very robust and have a wide range of applicability. However, they require a higher mesh resolution. Examples are volume of fluid (VOF) [18–21] and level set (LS) methods [22–25].

In VOF methods, the interface is given implicitly by a color function, which is defined as the volume fraction of one of the fluids within each cell. From this function, a reconstruction of the interface is made which is then propagated implicitly by updating the color function using the trasport equation. VOF methods are conservative and can deal with topological changes of the interface. However, VOF methods cannot accurately compute several important properties of the interface, namely the curvature and the normal to the interface. This is due to the discontinuous of the VOF function. Moreover, a high order of accuracy is hard to achieve because of discontinuities in the color function.

In LS methods, the interface is represented by the zero contour of a continuous signed distance function; this is known as the level set function. The movement of the interface is governed by a transport equation for the level set function. To keep the level set function as a signed distance function, a re-initialization process is needed but LS methods automatically deal with topological changes. It is generally easy to obtain a high order of accuracy just by using an ENO (Essentially Non-Oscillatory) or WENO (Weighted Essentially Non-Oscillatory) scheme. However, LS methods are not conservative, so that a physically incorrect loss or gain of mass occurs for incompressible two-phase flow. Several authors have overcome this problem by coupling the LS and VOF methods [4, 26–32]. In such coupled level set volume of fluid (CLSVOF) techniques, the level set function is used to accurately compute the curvature and the normal to the interface, while the volume of fluid function is used to capture the interface. Normally, a CLSVOF method is more accurate than both the standalone LS and VOF methods.

For two-phase flows in microchannels, surface tension forces play an important role in determining the dynamics of bubbles, whereas gravitational forces are generally negligible. It is also very important to consider the interaction between the boundaries and the fluids (wetting) by prescribing or computing the contact angle between the interface and the boundary. Imposing a static contact angle is inappropriate for modeling cases with moving contact lines. For such cases, a dynamic contact angle model should be implemented to more accurately represent the actual flow. So far, there is little experience in the literature on modeling of microscale two-phase flows using CLSVOF (or VOF) coupled with a dynamic contact angle model.

FLUENT is a general purpose commercial CFD code which has a VOF method (with its limitations) for modeling of two phase flows, as well as a two fluid model. FLUENT lets one specify a fixed value for the static contact angle between the solid boundary and the interface, or to specify a dynamic contact angle via UDFs. As mentioned before, a static contact angle is inappropriate for modeling cases with moving contact lines. In the case of the VOF method there is no mass transfer between liquid and gas phases which is required for modeling of heat transfer with phase change.

The objectives of this study are threefold: (i) to couple the LS and VOF methods within a commercial code (FLUENT) and then validate the performance of the resulting CFD scheme against several experimental examples; (ii) to implement a dynamic contact angle model into FLUENT; and (iii) to simulate evaporation inside microchannels. A variety of validations of the resulting model will be made then as test cases. Rather than proposing yet another new "in-house" numerical code, or the modification of published "in-house" codes, it was decided to work with UDF's to modify an important commercial code to achieve these goals.

This thesis is organized as follows:

- Chapter 1: Introduction;
- Chapter 2: State of the art review;
- Chapter 3: Theoretical models;
- Chapter 4: Details of FLUENT discretization procedure and code development;
- Chapter 5: Numerical results;
- Chapter 6: Conclusions.

Chapter 2

State-of-the-art-review

In this chapter, the 'one fluid' methods (with an emphasis on the level set and volume of fluid methods), surface tension models, numerical heat transfer with phase change and contact angle models will be reviewed.

2.1 'One fluid' methods

In this section the 'one fluid' methods will be reviewed on a paper by paper basis. Emphasis will be placed on the innovative steps proposed in each in order to form an overview of the state-of-the-art.

2.1.1 Level set methods

Level set (LS) methods are widely used in a large variety of applications, such as compressible and incompressible two phase flows, image processing, flame propagation, detonation shock dynamics, solidification, crystal growth, etching and deposition, etc. A general description of LS methods can be found in [33, 34]. Here, the work done with the LS method in the field of two-phase flow modeling will be reviewed on a paper by paper basis. It should be noted that nearly all of the papers were published in the last 4 to 5 years, indicative of the growing importance of the LS scheme since the beginning of this thesis in 2005.

Sussman, Smereka and Osher [22] presented a LS approach for computing solutions to incompressible two-phase flow. They considered a second order projection method which implements a second order upwind procedure for discretizing the convection terms. They pointed out the importance of the re-initialization procedure, which must be done at every time step in order to obtain a good conservation of mass. Two cases were considered for the rising bubble problems: one with moderate *Re* numbers and small surface tension, and the other with low *Re* numbers and high surface tension. An analysis of the water drop problem was also included.

Chang et al. [35] proposed a LS formulation to compute an incompressible two-phase flow with a free surface. The effect of surface tension was also included. To discretize the spatial derivative in the LS equation, either a 4th order central difference scheme or a second order ENO scheme, was employed. They found that the former scheme was more accurate and less diffusive, while the latter one was more robust, especially for their case with a large density ratio. They simulated several problems, including: the merging of two bubbles with the same density, the interactions of bubbles of three-density interfaces (two bubbles with different densities and the liquid density), the vortex sheet roll-up in a periodic jet, and an area preserving re-initialization test.

Sussman and Smereka [36] presented several 3D axisymmetric free boundary problems for two immiscible fluids (air and water). They offered another approach for computing free surface problems. They solved the Navier-Stokes equation in both fluids on a fixed grid, and they captured the front by defining the interface to be the zero LS of a continuous function called the LS function. Sussman and Smerka also performed several numerical experiments. First, they studied drop oscillations in zero-gravity environments, for both linear and nonlinear motion; The grid size was 50x100 and 64x128, respectively. They compared their numerical results with data available from the literature and they found very good agreement. Second, they studied the problem of a gas bubble rising due to gravity through a liquid. A grid size of 64x256 was used and good agreement with literature was found. Two approaches were used: namely the LS computation and the boundary integral method. Good agreement was found until the topology of the bubble changed. Third, they studied a water drop impinging on a pool of water using a 63x126 grid. Due to computational limitations, they found a qualitative, but not a quantitative agreement with experimental data available from the literature.

Sussman et al. [23] presented an improved LS method for incompressible two-phase flows. The LS method was coupled with a variable density projection method. A third order ENO scheme was used for the discretization of the convective terms, while a central differencing scheme was used for viscous and curvature terms. A third order Runge-Kutta method was used for the time discretization. The surface tension force was interpreted as a body force localized on the interface. They did several numerical tests like Zalesak's problem and rising gas bubbles with infinite Re numbers. A good agreement with available numerical data in the literature was achieved.

Sussman et al. [24] presented a numerical method using the LS approach for solving incompressible two-phase flow with surface tension. They coupled the LS scheme to an adaptive projection method for the incompressible Navier-Stokes equations. Several numerical simulations were presented: axisymmetric evolution of an inviscid air bubble in water with surface tension for different Weber numbers, 2D axisymmetric zero gravity drop oscillations, 2D impact of a drop on a water surface, 2D axisymmetric collision of drops, and fully 3D simulations of a single rising gas bubble in a stagnant liquid. They validated their algorithm with experiments and data available from the literature.

Sussman and Fatemi [25] presented a "constraint" added to the classical re-distancing algorithm to improve the conservation of mass. To discretize the LS equation and the modified re-initialization equation, they used a second or third order ENO scheme for the spatial derivatives and a second order or third order Runge-Kutta scheme for time discretization. To demonstrate the advantage of using the "constraint", they presented several simulations: 1D simulations, Zalesak's 2D problem, the collision of two drops, and a bubble rising in a stagnant liquid. They pointed out that even with the "constraint" they could not get 100% mass conservation.

Quecedo and Pastor [37] presented a method to solve two-phase flows using the finite element method. The proposed method discretized the Navier-Stokes equations and the LS function along the characteristics, using a Galerkin method. Additionally, the solution of the Navier-Stokes equation used a fractional-step method that allows equal order of interpolation for both the velocity and pressure fields. They did several numerical applications: a color drop advected by a uniform field, the flow in a T-branch (to test the coalescence of two interfaces), the flow around a cylinder, and a step cavity.

Yue, Lin and Patel [38] presented a numerical method that coupled the incompressible Navier-Stokes equations with the LS method in a curvilinear coordinate system. To discretize the governing equations, they used a finite volume method on a collocated grid with a four-step fractional step method. They used different techniques to discretize the LS equation, namely a third order quadratic upwind interpolation (QUICK), together with second and third order essentially non-oscillatory (ENO) schemes. They did several benchmark tests, such as a stationary circle, a rotating slotted disk, the stretching of a circular fluid element, a traveling solitary wave and the broken dam problem. They found that their results were in excellent agreement with both theoretical predictions and experimental data available in the literature.

Kohno and Tanahashi [39] developed a novel numerical scheme by coupling the LS method with adaptive mesh refinement in order to analyze moving interfaces. The finite element method was used to discretize the governing equations and the cubic interpolated propagation (CIP) scheme was used for the re-initialization of the LS function. In order to prove the accuracy of their method, they did several simulations: the collapse of a water column (here they neglected the surface tension effect due to the large scale), oscillation of a drop under zero-gravity (with surface tension effect), and the movement of a droplet under zero gravity (they analyzed the Marangoni effect by introducing a temperature field). For all simulations they used the conventional LS method and their algorithm (LSM-AMR). They obtained good agreement with data available in the literature and, moreover, with their algorithm they demonstrated that the computational time could be reduced by at least 50% with respect to the computational time for the conventional LS method.

Olsson and Kreiss [40] presented a mass conservative LS method for computing two-phase flow. Instead of using a re-initialization equation, like in the standard LS method, they used an artificial compression step to make sure that the interface keeps its thickness and shape. In order to demonstrate the capability of their method, they ran several 2D simulations for the followings cases: rotating circle, vortex test, rising bubble and falling droplet. They found that conservation of mass was significantly improved compared to the standard LS method.

Spelt [41] extended the LS method to simulate a flow with multiple moving contact lines. In his method, he accounted for inertia, a relationship between contact-line speed and contact angle, slip and contact-line hysteresis. In his technique, the slip-length parameter was imposed explicitly instead of being calculated. He studied the deformation of a droplet in a creeping shear flow and he compared his results with the results available from the literature (numerical results and experimental observations). His method converged and agreed with his previous work.

Gómez, Hernández and López [42] presented a local LS algorithm for simulating interfacial flows described by 2D incompressible Navier-Stokes equations. In their method, the LS transport and re-initialization equations were solved in a narrow band around the interface using an adaptive refined grid, while the Navier-Stokes equations were solved on the original grid. The Navier-Stokes equations were solved with a second order approximate method, while the transport and re-initialization LS equations were solved with a third order WENO scheme for the spatial derivative and a forth order Runge-Kutta algorithm for time discretization. They presented several numerical results, like the time-reversing single vortex test, a deformation field test, Zalesak's disk test, standing gravity and capillarity waves for small amplitudes, and the Rayleigh-Taylor instability. While their method was more accurate than the standard LS methods, a comparable degree of accuracy with a recent *particle* LS method [43] was obtained.

Nagrath, Jansen and Lahey Jr. [44] presented a stabilized finite element method for the 3D computation of incompressible bubble dynamics using a LS method. The streamline-upwind/Petrov-Galerkin (SUPG) method was used to discretize the governing flow and LS equations, including re-distancing, while the continuum surface force (CSF) [45] was applied to account for the surface tension. In order to demonstrate the capability of their algorithm, several simulations were presented: a 2D simple advection of a cylindrical bubble, a 2D simulation of buoyant bubble motion at medium Reynolds numbers, a 3D simulation of buoyant bubble motion for low and high *Re* numbers, and with a 3D simulation of two bubble coalescence. For all cases they obtained results which agreed well with other numerical data reported in the literature.

Lin et al. [46] presented a numerical method for free surface flows that coupled the incompressible Navier-Stokes equations with the LS method in the finite-element framework. The implicit characteristic Galerkin approximation together with the fractional four-step algorithm was employed to discretize the governing equations. Their validation process started with a vortex decay that was examined to calculate the discretization errors and estimate the order of accuracy of the scheme for verification. They proved that their Navier-Stokes solver was of second-order accuracy. Then, a stationary circle, the rotation of a slotted disk, and the stretching of a circular fluid were used to test the accuracy of the LS method. After the validation tests, two problems were simulated, namely the traveling of a solitary wave, and 2D/3D broken dams without and with a square problem. Their results were in good agreement with other existing experimental and numerical results.

van der Pijl et al. [47] presented a mass-conserving level-set method to model bubbly flows. Their algorithm is a level-set method, where a volume of fluid function is used to conserve mass when the interface is advected. The flow is computed with a pressure correction method using the Marker-and-Cell scheme. They used the volume of fluid function only to conserve mass, so the difficult convection step (interface reconstruction) was eliminated. The continuum surface model of Brackbill et al. [45] was used to account for the surface tension force. Several numerical simulations were presented: linear advection of a circle, Zalesak's rotating disk, and air-water flow (bubble rising and falling droplet). They obtained good agreement with numerical data available in literature and the mass was conserved up to a specific tolerance.

Smolianski [48] presented a unified approach for the numerical simulation of unsteady two-phase flows with moving interfaces. His approach relied on the finite element spatial approximation, the LS interface representation and the operator-splitting temporal discretization. He showed that the use of finite elements permitted him to localize the interface precisely, without introducing any artificial parameters like the interface thickness. Besides the re-initialization for the LS function, his algorithm proposed also a correction step of the LS function in order to enforce conservation of mass. He did several numerical simulations like the static bubble test (to observe the spurious currents), the rising bubble in an initially quiescent liquid, the breaking bubble, the coalescence of two bubbles, the Rayleigh-Taylor instability, and the bifurcating jet. He obtained good agreement with numerical and experimental data available in the literature.

Gutiérez and Bermejo [49] formulated a LS method in the framework of finite element-semi-Lagrangian methods to compute the solution of the incompressible Navier-Stokes equations with a free surface. They used a quasi-monotone semi-Lagrangian scheme, which was both unconditionally stable and essentially non-oscillatory, to compute the advective terms in the Navier-Stokes equation, the transport equation, and the equation of the re-initialization stage for the LS function. They did several numerical tests: the steady advection test (the Zalesak's rotating disk), the bottom bump (they computed the profile of the wave generated by the flow past a bump located at a bottom of a shallow channel), and the submerged hydrofoil. They obtained good agreement with either the exact solution or with data available in the literature.

Zheng et al. [50] presented an implementation of an unstructured adaptive re-meshing algorithm using finite-element/LS methods in simulations of droplet and fluid-fluid interface interactions and coalescence in multiphase flows. They implemented an efficient MINI element Stokes solver, and a Navier-Stokes solver based on a new modified projection method that uses a few iterations in the projection step to handle the non-linear term and ensures divergence free flow. They adapted the computational mesh at every time step using a local length scale that was linearly proportional to the magnitude of the level-set function. They presented several numerical simulations: Stokes flow dynamics of 2D and 3D deformable drops in a steady shear flow, drop impact onto a fluid interface under gravity, coalescence events in a large system of drops, a 3D drop breakup, and a 2D numerical investigation of the impact and rebound of a drop onto a fluid interface with inertia. They found good agreement with other numerical and experimental data available in literature.

Ge and Fan [51] studied the impact of sub-cooled water and *n*-heptane droplets on a superheated flat surface based on 3D numerical simulation. They developed a three-dimensional model to describe the collision mechanism of the sub-cooled droplet with the superheated surface. The droplet deformation was tracked by the use of the LS method, while a vapor-layer model was utilized to account for the vapor pressure force due to the evaporation. They simulated three different conditions: 1.5mm *n*-heptane droplets impacting on a stainless steel surface, 3.8mm water droplets impacting on an inconel surface and 4.0mm water droplets impacting on a copper surface. The simulation results achieved good agreement with experimental measurements reported in the literature.

Marchandise, Remacle and Chevaugeon [52] developed a quadrature-free Runge-Kutta discontinuous Galerkin method to solve the conservative form of the LS equation on both 2D and 3D unstructured grids. In their method, they did not need to reinitialize the LS function. They also showed that this formulation achieved a higher accuracy than ENO and WENO schemes for smaller meshes and reduced computational time. To test their algorithm, they did classical tests like bubbles in a uniform flow, Zalesak's rotating disk, a vortex in a box, and a 3D deformation field. The results obtained confirmed the accuracy of their method regarding the shape and, most important, the conservation of mass, since it is well known that the LS method has mass conservation problems.

Marchandise and Remacle [53] next presented a numerical method for the simulation of 3D incompressible two-phase flows, in an algorithm which combined an implicit pressure stabilized finite element method for the solution of incompressible two-phase flow problems with a level set method implemented with a quadrature-free discontinuous Galerkin method. Using this algorithm, they could compute the discontinuous integrals without introducing an interface thickness or re-initializing the LS function. For the time integration, a second order three-point backward difference scheme was employed. The overall scheme was second order accurate in space and time. They presented several numerical results for 2D Poiseuille two-phase flow, 2D sloshing in a rectangular tank, 2D and 3D dam break problem, and for Rayleigh-Taylor instability. Their results compared well with other numerical data available in the literature.

Deshpande and Zimmerman [54] proposed a unique approach to simulate mass transfer across a moving droplet. In the first part of their paper, several buoyancy-driven droplets rising through a quiescent medium were simulated using the LS method in order to study the influence of various dimensionless numbers on the deformation of a moving droplet and interfacial transport. Then, they proposed a new algorithm for computing the mass transfer across a moving droplet by adopting a two stage approach where the convection-diffusion equations for mass transfer were decoupled from the governing equations of the LS methodology. With this formulation, they could accurately infer the mass transfer coefficients without using any empirical correlations. Using this approach, the mass transfer coefficients obtained were found to be of the same order of magnitude of those obtained using empirical correlations.

Hu et al. [55] presented a conservative interface method for both multi-fluid and complex boundary problems for compressible flows. While the discretized governing equations are updated conservatively, the method treats the topological changes naturally by combining interface description and geometric operations with a LS technique. They did several 1D and 2D test cases: a gas-gas interaction, the well-known air–helium shock tube problem, a stiff air–helium shock tube problem, a 2D air–helium interaction, a gas–water shock tube problem, a shock impacting on an air–water interface, an underwater explosion, the collapse of 2D air bubbles in water, a shock diffraction on an airfoil and the liftoff of a cylinder by a shock wave. For the 1D simulations a good agreement with analytical solutions was obtained, while for the 2D simulations a reasonably good agreement with data available in the literature was achieved.

Nagrath et al. [56] presented a novel stabilized finite element method employing a combination of the ghost fluid and the LS method for studying the hydrodynamics of the implosion and rebound of a small air bubble in water. In their method, both fluids were treated as compressible fluids. The continuum surface force (CSF) model proposed by Brackbill et al. [45] was implemented to incorporate surface-tension-induced stresses. Also, an efficient re-distancing algorithm was adopted to ensure the volume of each phase was conserved during re-distancing. To demonstrate the accuracy of their method, first they did several tests: the simple advection of a cylindrical bubble, the two-phase shock tube problem, and an imploding air bubble in water. The results were then compared with the Rayleigh-Plesset analysis and found to be in good agreement for most of the implosion process.

Price and Chen [57] proposed a LS formulation in a generalized curvilinear coordinate to simulate the free surface waves generated by moving bodies or the sloshing of fluid in a container. To solve the re-initialization equation of the LS function, a least square technique was adopted. The surface tension effect was not included in their model. They did several numerical examples like oscillating a flow in a two-dimensional tank, the broken dam problem, the sloshing tank problem, and the flow around a Wigley ship hull. A good agreement with other numerical data reported in the literature was found.

Losasso, Fedkiw and Osher [58] discussed both the historical developments and the most recent work on improving the computational accuracy of the LS method, focusing in part on applications related to incompressible flows due to both of its popularity and stringent accuracy requirements. They reviewed higher order accuracy methods such as that of Hamilton-Jacobi WENO, the methods for maintaining a signed distance function, the hybrid methods such as the particle LS method and the coupled LS volume of fluid method, and the adaptive grid technique such as the octree approach to free surface flow.

Fukagata et al. [59] performed a numerical simulation of an air-water two-phase flow in a 20μm tube using a LS method with heat transfer. In their simulations, the mean pressure gradient and the wall heat flux were kept constant. Several cases were considered with *Re* numbers for gas and liquid phases ranging between 0.34–13 and 16–490, respectively, and the capillary number between 0.0087–0.27. They found the gas velocity to be 1.2 times higher than the liquid velocity, which agreed with the Armand void fraction correlation for macro tubes. Their computed two-phase flow local Nusselt number was found to be considerably higher than that for single phase flow. The frictional pressure drop was found to be higher than that predicted by the Chisholm correlation for the laminar-laminar case.

Gibou et al. [60] proposed a LS sharp interface method for two-phase flows with phase change. They used a LS method to capture the interface and a ghost fluid method to impose the jumping conditions at the interface. Only the viscosity was treated with a delta formulation mechanism for simplicity. To solve the Navier-Stokes equations, a standard projection method was adopted, where both the LS equation and the re-initialization equation were discretized with a fifth order HJ-WENO scheme. They started their numerical experiments by modeling 1D constant and a variable velocity two phase flows from which they obtained very good agreement with the theoretical results. They concluded their numerical simulations by modeling the 2D vaporization of a drop and the 2D film boiling problem with excellent qualitative results.

Tangui Ménard and Berlemont [61] presented a method devoted to reactive interface simulations with emphasis on vaporization. They used the LS method for interface tracking associated with the ghost fluid method to impose accurate jump conditions across the interface. The Navier-Stokes equations were solved with a first order projection method, while the convective terms were discretized with a fifth order WENO scheme, and the diffusive terms were computed with a second order central scheme. They did several 2D axisymmetric tests like vaporization of an axisymmetric static droplet (radius is 150μm), 2D droplet vaporization with a constant vaporization speed without any coupling with heat and mass transfer (radius 200μm), and vaporization of a moving droplet. For all cases, they obtained physically meaningful results.

Carrica, Wilson and Stern [62] proposed a single-phase LS method for unsteady free surface flows. In contrast with the standard level-set method for incompressible flows, the single phase LS method is concerned with the solution of the flow field in the denser phase only. This approach has several advantages: the computation takes place only in the liquid phase with a saving for the computational time, and by solving the equations only in one phase the problems with large density ratios disappears. As a drawback, the method does not solve the fluid equation in the gas phase (air). Furthermore, the stresses on the liquid caused by the air must be negligible or somehow specified. The re-initialization procedure of the LS equation was split into two steps. The first step was a close point re-initialization for those grid points that were neighbors to the interface, while in the second step a transport equation for the rest of the grid point was solved. They presented three numerical examples: a linear progressive wave, sloshing in a fixed rectangular tank, and the forward speed diffraction in a surface ship. They compared their numerical results with both analytical and numerical results available in the literature with good agreement.

In summary, it can be said that LS methods proposed so far represent a viable solution to use in modeling of two-phase flows. They are easy to implement in 3D and can precisely compute the curvature and normal to the interface. However, their main drawback as standalone methods comes from the fact that they are not conservative and physically incorrect loss or gain of mass occurs. Moreover, they are quite expensive in term of CPU usage due to the need to re-initialize the level set function after each time step. At the end of this chapter, a table which summarize the pros and cons of several ‘one fluid’ methods will be presented.

2.1.2 Volume of fluid and interface capturing methods

The volume of fluid method (VOF) is widely used in the computation of two phase-flows due to its advantage in conserving mass. Here, the innovative aspects of the VOF method applied to two-phase flows are reviewed on a paper by paper basis. Some papers addressing interface tracking methods are also included in this overview.

Hirt and Nichols [18] presented the VOF method and its implementation in a computer code SOLA-VOF. In order to locate the interface, the method first defines a function F in every mesh cell with values from 0 (cells with only fluid 2) to 1 (cells with only fluid 1). Therefore, a cell which has the value of the function between 0 and 1 contains the free surface. In order to test their method, several tests were simulated: a broken dam problem, an undular bore, a breaking bore, Rayleigh–Taylor instability, and a reactor safety application. The obtained results showed the capability of the method to provide good results for a wide range of complicated problems.

Youngs [19] briefly presented in his paper a 2D time dependent compressible finite difference Eulerian method. For interface tracking he used a simple method in which the portion of each interface in a cell was approximated by a straight line. For testing his method, he ran several test cases such as: Rayleigh-Taylor instability, a shock tube problem, an expanding pipe, and explosively driven flows. The results obtained with his method were in good agreement with both experiments and other Lagrangian methods.

Unverdi and Tryggvason [17] in their paper presented a method to simulate unsteady multi-fluid flows. The interface was represented explicitly, by a separate unstructured grid that moved through the stationary grid (where the flow field is discretized). They presented a front tracking method for viscous, incompressible multi-fluids flows, a method which they implemented for both 2D and 3D simulations. The Navier-Stokes equations were solved in a rectangular domain with a finite difference method, while the advection of the density and the viscosity fields was obtained by explicit tracking of the interface between the two fluids. They ran several 2D cases on a grid of 65x129 mesh points for different Eötvös and Morton numbers (Eötvös numbers ranged from 1 to 104 and Morton numbers from 10^{-7} to 102). They compared their calculations with ones available in the literature and, in spite of some differences between their setup and that in the literature (3D simulation, axisymmetric), they found a very good qualitative agreement. Also, they simulated the coalescence of two bubbles for different conditions. Furthermore, they carried out 3D simulations on a grid of 32x32x64 mesh points for different conditions.

Tomiyama, Zun, Sou and Sakaguchi [63] presented a numerical simulation of a 2D single bubble both in a stagnant liquid and in a linear shear flow using a VOF method. They investigated the effect of Eötvös and Morton numbers on the fluctuating bubble motion, as well as the effect of liquid velocity distribution. Their numerical results agreed qualitatively well with available experimental results. Also, they found that Eötvös and Morton numbers played the dominant role on the nature of the migration of the bubble rather than the bubble diameter.

Li [20] presented a piecewise linear interface calculation (PLIC) method. Tests confirmed the ability of this method to track the interface accurately in a two-phase flow with a free surface.

Kothe et al. [64] presented algorithms for tracking interfaces with PLIC–VOF methods in 2D and 3D. The CSF model was used to account for the surface tension forces. To illustrate the properties of PLIC–VOF and CSF model, they did several numerical simulations, namely: simple advection and rotation tests, and static inviscid spherical drop with surface tension acting along its interface. The numerical results they obtained confirmed the accuracy of their algorithm.

Rudmann [65] in his paper presented a new algorithm for volume tracking, based on the concept of flux corrected transport (FCT). The unique advantage of this method is that an interface reconstruction in not required. The “color” function C is first calculated, as an intermediate value using a diffusive advection scheme, and then an anti-diffusive flux is defined which is applied to the intermediate value of the “color” function to obtain its value at next time step. After he reviewed other volume-tracking methods (simplified line interface calculation (SLIC), volume of fluid (VOF), and method of Youngs), he ran several 2D simulations both for only the advection of C and for the case when C is coupled with the momentum equation (constant unidirectional velocity field, solid boundary rotations, shearing flow and for coupled color function and momentum equation Rayleigh-Taylor instability). For all cases the Youngs' method produced the most accurate results, but his method FCT-VOF, gave better results than the SLIC and VOF methods.

Tomiyana, Zun, Higaki, Makino and Sakaguchi [66] presented a numerical method for solving 3D bubbly flows. Their method was based on a two way particle tracking method, which took into account the effect of the bubbles on the liquid phase and vice-versa. To test their new method, first they simulated laminar bubbly upflow in a vertical duct using the Eötvös number and bubbly size distribution as parameters (and keeping constant the Morton number and dimensionless volumetric flux of the liquid phase). As results they obtained wall-peak, core-peak and intermediate bubble distributions. Significantly, their method has the capability to predict the transition between bubbly flow and slug flow. Then, to establish if their new algorithm could provide better predictions for bubble-induced liquid velocity fields, a single bubble rising through stagnant liquids was calculated. Both particle tracking and interface tracking methods were used. If the bubble size was greater than the computational cell, the liquid velocity fields predicted by the particle tracking method agreed well with those predicted by the interface tracking method.

Jeong and Yang [67] have adapted the VOF method for the finite element analysis of transient fluid flow with a free surface. For a higher resolution capture of the free surface configuration, Jeong and Yang incorporated an adaptation technique for generating an adaptive grid. Also, they developed three techniques based on the VOF model to increase the accuracy of the analysis: a filling pattern technique to predict the free surface, an advection treatment technique to calculate the fluid advection iteratively and forcedly, and a smoothing technique to improve the non-smoothness of the predicted free surface. In order to test their developed techniques, they analyzed radial flow with one point source, and the collapse of a dam. Then they compared their 2D numerical values with theoretical or experimental ones. For the first case studied, they employed two grids, one fixed and one with refinements. Obtained results were compared with theoretical ones and good agreement was found. Moreover, the results obtained with grid refinement were closer to the theoretical ones than those obtained with a fixed grid. For the second case they employed three grid systems, two fixed and one with refinement. Comparison with the available experimental data showed a good agreement, and again the conclusion was that grid refinement produces better results than a fixed grid.

Chen and Li [68] presented a numerical model for two-phase flows with high density ratios. Their method is a modified VOF method combined with a semi-implicit algorithm and higher-order advection scheme on a collocated grid. The surface tension force is modeled by a continuum surface force approximation. For testing their interface tracking method, a comparison between their method and a flux-corrected VOF scheme was made by simulating a spherical blob axisymmetrically rising in a liquid with a constant velocity field. A good agreement between these two methods was obtained. To further test their model, two more test problems were considered, namely bubble coalescence and a bubble rising towards a free surface. Their results qualitatively agreed with the data available from the literature.

Rider and Kothe [69] presented a new 2D algorithm for the volume tracking of interfaces (VOF–PLIC). Their scheme has second order spatial accuracy and, thanks to the unsplit multidimensional time integration, also second order temporal accuracy. Simple tests like body rotation and translations confirmed the accuracy of the method. Also they applied their method to more complicated problems, like single vortex and 16 vortices, using 322, 642 and 1282 orthogonal, uniform cells. Numerical results confirmed the algorithm's excellent solution quality and fidelity.

Scardovelli and Zaleski [21] made a review of existing simulation methods for the Navier-Stokes equation with interfaces, with emphasis on fixed grid methods, such as VOF. They also did some 3D simulations, namely for sessile droplets, sedimentation of a droplet array in creeping flow, breakup of liquid jets, and the Kelvin-Helmholtz instability. They concluded their paper with the statement that important calculations have not yet been done, and many complex flows for jets and splashing droplets have been studied only in a preliminary way.

Gueyffier et al. [70] proposed a numerical scheme based on the VOF method which uses a piece-linear interface calculation in a 3D domain. Surface tension forces were included in the model either by using the continuous surface stress or the continuous surface force models. Several numerical simulations were done, namely the sedimentation of a droplet array, the simulation of pinching pendant drops, and the simulation of bubbles rising in a viscous liquid. By comparing their results with analytical and experimental results, a good agreement was found.

Welch and Wilson [71] presented a numerical method for simulating two-phase flow with mass transfer. They used the volume of fluid method together with a mass transfer model and a model for surface tension. To account for two-phase flow, they advected the interface using Young's enhancement to calculate the density field and then the void fraction in every computational cell. To test their method, they ran several 1D problems, like the Stefan problem (using water and three saturation pressures), the sucking interface problem (with water at atmospheric pressure and three grid dimensions). They then simulated 2D film boiling on horizontal tubes using three grid dimensions (16x48, 32x96 and 64x192). Next, they simulated mushroom shaped bubbles, and compared their results for bubble shape, velocity field and temperature field with ones available from the literature. A good agreement was found.

Shin and Lee [72] proposed a numerical technique for simulating incompressible viscous fluid flow with a free surface. To calculate the flow field, they used the penalty finite element formulation; a modified VOF method was used. For verifying the scheme, they proposed two test cases: namely the broken dam problem and the solitary wave propagation problem. Their results agreed well with analytical and experimental results. To demonstrate the capability of their algorithm on a more complicated problem, they considered a mold filling process. Physically good results were also obtained.

Ginzburg and Wittum [73] have introduced a 2D two-phase model based on an implicit volume discretization of the Navier-Stokes equations on an unstructured computational grid, which either adaptively refined the interface, or produced a grid that was aligned with it. An important advantage they mentioned is the ability to accurately compute the surface tension force on the cubic spline aligned grid. They also found that spline surface reconstruction considerably reduced the parasitic currents in VOF models, even if they are based on regular grids. Their solution procedure does not use projection methods, but solves the entire linear system for the unknown pressures and velocities. To validate their model, Ginzburg and Wittum carried out three numerical experiments, comparing their results with ones available in the literature:

- First experiment: buoyant 2D bubbles with three different Eötvös numbers (1, 10 and 104) and for each Eötvös numbers four Morton numbers were considered. They used a regular computational grid of size 32x64. At steady state, bubbles shapes and streamlines were found to be in good qualitative agreement with results from other authors;
- Second experiment: two bubbles with a Morton number of 0.065 and two Eötvös numbers of 39.3 and 123.1. Again good agreement between their data and literature data was found;
- Third experiment: bubbles rising in a viscous liquid. The bubbles' shapes and the behavior of the surrounding liquid qualitatively agreed with experimental results.

Cerne, Petelin and Tiselj [74] proposed an algorithm which coupled VOF with the two-fluid method to more accurately simulate two-phase flows. The advantage of this algorithm is that flow regimes with clear phase separation are no longer need to be simulated with the two-fluid model, since these regions can be more accurately simulated with the VOF model. To test their coupled algorithm, they ran two numerical experiments, namely simulation of the vortex flow and the Rayleigh-Taylor instability. It was shown that for a coarse grid the VOF model could not accurately predict the flow, but on the same grid the coupled algorithm gave significantly better results. They concluded that their coupled algorithm enabled a more realistic calculation of a wider range of two-phase flow phenomena than the attainable with the pure VOF model.

Wörner [75] presented a numerical study which investigated the influence of the gas-liquid density ratio on bubble shape and bubble Re number by a 3D VOF scheme. Four cases were considered with liquid densities 2, 5, 10 and 50 times the gas density, Eo of 3.06, Mo of 3.09x10^{-6}, and a viscosity ratio of 1. For all cases he obtained a Re of 56; compared to a theoretical Re of 59.3 for the same Eo and Mo. For all simulations, he obtained an oblate ellipsoidal bubble that rose steadily along a rectilinear path. He did not find any notable influence of the liquid-gas density ratio on the bubble shape and bubble Re number.

Tryggvason et al. [16] presented numerous direct numerical simulations of multiphase flows using a front tracking method. The method was a hybrid front capturing, front tracking technique. The authors presented many details of the mathematical formulation of their method as well as the numerical procedure they implemented. To demonstrate the accuracy of their algorithm, many problems were modeled: homogeneous bubble flow, atomization (one of the key elements in the successful combustion of liquid fuels), flows with variable surface tension, solidification, and boiling. The obtained results for 2D or 3D cases were in good agreement with other simulations found in the literature.

Meier, Yadigaroglu and Smith [76] presented a new method to include the surface tension in PLIC–VOF methods. The interface curvature determination, by least square fitting of reference data, and the surface force implementation, resulted in spurious currents 2 orders of magnitude smaller than for the original CSF model [45] and Zaleski et al. [77] Also, in their method the curvature was estimated about 3 to 7 times more accurately than with the CSF model. Dynamic tests including droplet oscillation driven by surface tension, rising bubbles in six different regimes, and breakup of a liquid jet confirmed the accuracy of their method.

Ghidersa [78] in his PhD thesis developed a new numerical method to describe the flow and heat transfer characteristics of a gas liquid two-phase flow in channels with a hydraulic diameter on the order of 1mm. His method was implemented in the "in-house" computer code TURBIT-VOF. For spatially periodic two-phase flows in small channels, he introduced a new concept to model convective heat transfer in periodic fully developed flow. He constructed a new form for the pressure equation using a procedure that ensures the consistency between the gradient and divergence discrete operators. Using this method he simulated the bubble-train flow along a square channel for both adiabatic and diabatic flow. For adiabatic flow the results obtained with TURBIT-VOF were in good agreement with experimental data from literature, while for diabatic flow he found no data to compare his results. He also found that the capillary number was the main physical control parameter for two-phase flow in small channels.

Aulisa, Manservisi and Scardovelli [79] presented a new mixed markers and VOF algorithm for the reconstruction and advection of interfaces in 2D spaces. With their method, both the volume fraction function C (VOF methods) and surface makers were used to locate the interface within the computational cells. In their algorithm, a standard fourth order Runge–Kutta method was used to integrate the partial differential equations. In order to test their new algorithm, the widely used standard test for interface tracking algorithm was used, namely uniform translations and rotations, and non-uniform vorticity tests. A 2D square domain was used with different resolutions (162, 322, 642, 1282, and 2562). Results obtained with their new method were compared with the classical PLIC/VOF methods and for all cases a better accuracy was achieved with their new method.

Kim, Park and Lee [80, 81] proposed a new VOF based scheme for surface tracking based on a donor-acceptor method. In their method, the free surface location is traced by calculating fluid volume fluxes on the fixed finite element mesh and updating the values of the fractional VOF F using the donor acceptor scheme. For that, a baby-cell was devised to represent a uniform sub-volume at a designated position in a cell. To test the accuracy of their method, two problems were investigated, namely the broken dam problem (a rectangular column of water in hydrostatic equilibrium is confined between a vertical wall and a gate which is suddenly removed) and the solitary wave propagation problem (a single wave elevated above the surrounding undisturbed water level traveling in the direction of wave propagation with a constant velocity). Their results, obtained on different uniform and non-uniform mesh sizes, were in good agreement with experiments, other numerical results, and analytical solution. After the accuracy of their method was demonstrated, two numerical applications were considered, namely the cavity filling problem and the sloshing problem. For the cavity filling problem, 2D and 3D simulations were considered. Because no experimental results were available for such a problem, they compared their results with results obtained with commercial code Flow 3D. A good agreement was found. Also for the sloshing problem 2D and 3D simulations were considered. The results obtained with their scheme were in qualitative agreement with experimental and numerical results available from the literature.

Gao, Gu and Guo [82] presented a numerical simulation of stratified oil-water two-phase turbulent flow in a horizontal tube using the VOF model. The continuum surface force model (CSF) was used to include the effect of surface tension. In order to avoid the use of wall functions, they used the k-turbulence model for low Re numbers. Their numerical results agreed well with the experimental data available from the literature. Also, they proposed a correlation for computing the pressure drop and the oil holdup based on their numerical simulations.

Scardovelli and Zaleski [2] presented two new VOF reconstruction algorithms which are based on a least-square fit technique. Also, they presented a new mixed split Eulerian implicit-

Lagrangian explicit algorithm for the advection of the interface. Standard advection accuracy tests were made by modeling slotted disk rotation and Rieder-Kothe reversed single vortex flow. Tests showed that their reconstruction algorithm coupled with their mixed advection scheme performed consistently better than other VOF/PLIC, VOF/SLIC and unsplit algorithms.

Theodorakakos and Bergeles [83] presented a new method for adaptive grid local refinement in order to capture the sharp liquid-gas interface and so speed up the calculations. VOF was used on unstructured 2D and 3D grids. To account for surface tension forces, they used the CSF method of Brackbill [45]. Simple tests, such as square bubble convection and rectangular bubble rotation showed the accuracy of the VOF method and the fact that local refinement did not introduce any thickening or distortion in the shape of the interface. 3D simulations of water droplet impingement on Pyrex glass and of a micro-metric water droplet impinging normally on a dry wall showed that using adaptive mesh refinement considerably reduced the computational time, while the accuracy of predicting the free surface was increased.

Aulisa, Manservisi and Scardovelli [84] proposed a surface marker algorithm coupled with a local area conservation method applicable for 3D incompressible and compressible flows. Their algorithm was an improvement of their previous model [79] applied for 2D flows. To prove the efficiency and the accuracy of their new model, several 2D tests were done: the single vortex test and 2D deformation tests. Their results compared very well with the front tracking results, and are better than LS methods and VOF-PLIC methods. Then they ran several 3D tests, like translations, solid body rotations, and 3D deformation fields. The results showed no deformation of the interface. They obtained similar results when they applied their method to compressible flows.

Biausser, Guignard, Marcer and Fraunie [85] proposed a semi-Lagrangian computational method for simulating 3D two-phase flows. In order to deal with high density ratios, their method computed the Navier-Stokes equation in both fluids. Interface tracking was achieved by the segment Lagrangian VOF (SL-VOF) method. This method provides better accuracy relative to the standard VOF algorithm, and also reduces the computational time by allowing the use of a larger time step. In order to validate their new model, they ran several tests (2D and 3D). First, they considered the advection of a sphere in an uniform, constant horizontal velocity field of one cell per time step. After 40s they observed the conservation of the interface shape even for a coarse mesh. Next, they simulated sphere advection in a distorting velocity field, and they compared the analytical solution with their SL-VOF method and with the standard VOF method. The solution computed with their SL-VOF method was very close to the analytical one, as opposite to the standard VOF method. Then, they ran Rayleigh-Taylor instability tests, both in 3D and axisymmetric 2D computational domains. A very good agreement was observed between these two simulations.

Lorstad and Fuchs [86] presented an improved volume of fluid method which is applicable to high density ratios and a large range of bubble Reynolds numbers. Their algorithm was presented for 3D flows. They used a staggered Cartesian grid system, including local mesh refinements at regions with larger gradients. For modeling the surface tension force, they used the Continuum Surface Force (CSF) model. For the phase transport equation or “color” function equation they used a combination of Youngs’s method [19] together with the direction split technique of Rudman [65]. To investigate the accuracy of their method, Lorstad and Fuchs considered some cases concerning bubbles which had low density jumps relative to the surrounding liquid (stokes flow, rising bubbles at low *Re* and density ratio). Their final test consisted of the full 3D computation of air bubbles moving in water. The results they obtained showed reasonable quantitative agreement with the experimental data available from the literature.

Lorstad, Francois, Shyy and Fuchs [87] compared two popular methods for multi-fluids dynamics, namely the VOF and the immersed boundary method (IB). The continuum surface force model (CSF) was used to model the surface tension forces. Two schemes were implemented in the VOF model: the Rudman [65] algorithm and the direction averaged curvature model (DAC). Also two cases were modeled with the IB method depending on the estimation of the surface tension forces, namely a rising droplet at low *Re* numbers and a rising droplet at high Re numbers. Similar results were obtained and, moreover, good qualitative agreement with other data available from the literature was found.

Wang, Borthwick and Taylor [88] presented a finite-volume VOF method for simulating viscous free surface flows on dynamically adaptive quad-tree grids. They used fine grid resolution at the gas-liquid interfaces and coarse grid resolution where the gradient of flow variables was small. They did several numerical simulations to test the accuracy of their method. First, they considered the advection of a square patch of dyed fluid by uni-directional flow in a square domain. When they used the adaptive quad-tree grid, the results looked very similar to those obtained with the corresponding regular grid, while the computational time was significantly lower. Then, they considered the rotation of a rectangular region of dyed fluid by a forced vortex. Because the rectangular dye occupied most of the computational domain, the adaptive quad-tree grid method was less efficient than the uniform grid method regarding CPU time. Next, they simulated the 2D sloshing of a viscous liquid in a rectangular tank. The results they obtained were in reasonable agreement with linear analytical solution of the Navier-Stokes equations.

van Sint Annaland, Deen and Kuipers [89] presented a 3D VOF method that could handle large viscosity and density ratios, as well as a large value of the surface tension coefficient. To compute the surface tension force, the CSF model of Brackbill *et al.* [45] was used. To test their algorithm, several applications were done. For the standard advection test (when flow is advanced for n time steps and then reversed for the same number of time steps), their code produce very good results even though small differences were observed between the initial and final shape of the bubble. Also, they ran a simulation to compare the pressure inside the bubble with analytical solution of Youngs-Laplace equation. A 6% deviation from the analytical solution indicated their good implementation of the surface tension model. After they successfully tested their code, a comparison with the bubble diagram of Grace [90] was made. The computed Reynolds number and bubble shape agreed very well with the data obtained from the bubble diagram. They also tested the coalescence of two gas bubbles in an initially quiescent liquid. They tested both co-axial coalescence and oblique coalescence (the second bubble initial position was shifted in the x direction). Good agreement between their results and data available from the literature was found.

Lopez, Hernandez, Gomez and Faura [91] proposed an improvement of their previous reconstruction method for 2D flows, a method which allows tracking fluid structures to be thinner than the cell size. Although they used markers, the method presented in their paper remained a PLIC-VOF method. As usual, when a new method is proposed, several numerical experiments must be done to test the accuracy of the new method. The first test they did was the advection of a rectangular area of fluid, where the velocity is horizontal and varies linearly in the vertical direction. The results they obtained were better than those obtained by different authors with others methods. They then simulated a time-reversed single vortex flow. Again the results obtained with their new method were better than those obtained with other methods. Next they ran a Rayleigh-Taylor instability test. They concluded that their method produced both efficient and accurate results that are an improvement on other VOF and hybrid methods.

Yadigaroglu [92] in the first part of his paper reviewed 'one' and 'two' fluid models with interface tracking. In the second part, he presented some applications regarding the condensation of steam-air mixtures injected into a pool of water, and the numerical modeling of the critical heat flux.

Tang and Wrobel [93] presented an interface tracking method for modeling the flow of immiscible metallic liquids in mixing processes. The PLIC-VOF and CSF methods for the surface tension force were adopted by them. They presented results obtained with FLUENT, for one-sided shear flow, two-sided shear flow, and the tracking of droplets in mixing flow. For their simulations they used an immiscible metallic Pb-Zn binary alloy. Their results were in good qualitative agreement with the experimental results.

Du et al. [94] described in their paper significant improvements to the Front Tracking package in the 3D handling of topological bifurcations. They presented a new 3D front tracking algorithm "Locally Grid based tracking", and they compared their new algorithm with the LS method, VOF and other methods. In both cases, the front tracking method they proposed was more precise and accurate than the others. They provide their developed code for scientific applications.

Yadigaroglu and Lakehal [95] reviewed 'one' and 'two' fluid models for simulating two phase flows in nuclear applications. Turbulence modeling with the 'two' and 'one' fluid models was also discussed. They introduced the concept of cascade computational multi-fluid dynamics by giving the following example: condensation of large steam-air bubbles that involves the interfacial scale, the mesoscale level and the large system scale. A numerical application, namely the injection of air bubbles into water, proved the capability of the 'one' fluid VOF method coupled with LES to predict the turbulence.

Taha and Cui [96] carried out numerical simulations to model slug flow inside square capillaries containing Newtonian liquids. They used a 2mm hydraulic diameter tube with silicone oil as the liquid and air as the gas. They used the VOF method implemented in the commercial code FLUENT to simulate the motion of a single Taylor bubble rising in a flowing liquid through a capillary. The grid used in their work was 59x59x560, with refinement near the wall. They studied axisymmetric flow in circular channels, as well as the 3D asymmetric flow in square channels. At low capillary numbers they found a discrepancy between the propagation of Taylor bubbles and velocity profiles existing in circular channels and in square channels. They computed bubble velocity and diameter and their results were in very good agreement with experimental measurements from the literature. They also computed the velocity field around the bubble which again was found to be in very good agreement with the literature. Based on their hydrodynamic study, mass transfer and heat transfer was enhanced in square capillaries and is explained by the corner effect, which creates a better exchange between liquid plugs.

In their second paper, *Taha and Cui* [97] studied slug flow in vertical tubes in both stagnant and flowing liquid. The VOF method, implemented in the commercial code FLUENT, was again used for this study. To reconstruct the bubble free surface, the piece linear interface calculation was applied. They did a 2D axisymmetry simulation with 59x560 mesh elements, covering a wide range of tube diameters and liquid properties. They found a highly agitated mixing zone in the bubble wake. Based on the inverse viscosity dimensionless number *Nf*, Taha and Cui have established different patterns for the wake zone: two closed toroidal vortices, a zone with periodic undulation and loss of symmetry, and at large viscosity numbers the bubble wake opens and turbulent eddies are shed from the main bubble wake. The shape and velocity of the Taylor bubble and velocity distribution around the bubble were computed and found to be in good agreement with experimental results from literature. Also, they did numerical experiments for Taylor bubbles rising in a tube tilted 20° from the vertical.

Ghorai and Nigam [98] carried out two phase computational fluid dynamics calculations using an Eulerian model (two fluid models) and the commercial CFD code Fluent. They did numerous numerical experiments for gas velocities that ranged from 1.2 to 12.5m/s and liquid velocities from 0.0066 to 0.1m/s. For the reconstruction of the interface, they used a piecewise linear approach. For both phases, *k*- turbulence models were used. To discretize the governing equations, an unstructured non-uniform grid was used with 40x400x500 meshes. They compared their numerical results with experimental data available from the literature and they obtained a reasonably good agreement. Based on their many numerical experiments, they also proposed a new correlation for the ratio of the interfacial friction factor (f_i) to the wall friction factor (f_w). This was expressed as a function of the ratio of gas velocity to liquid velocity for wavy flow. An acceptable agreement was found between their predicted values and data from the literature.

Kim, Yang, and Jeoung [99] have analyzed molten metal flow at the filling stage of the casting processes using the VOF and finite element methods. To improve the numerical computations, they developed an adaptive grid refinement technique incorporated into the VOF method. They also developed a filling pattern technique based on the VOF method to improve the accuracy of the analysis. To test their methods, several 3D numerical simulations were made. First they simulated the filling of a step cavity for two cases, with and without mesh refinement. Then they compared their results with experiments available from the literature, they concluded that the case with mesh refinement was closer to the experiment than the case without mesh refinement. Second, they simulated a benchmark test of a casting problem, again in two cases with and without mesh refinement. As in the first case the conclusion was, based on comparison with available data, that mesh refinement produces better results than the case without refinement. Next they tested a cover of a transmission case using aluminum alloy as the molten metal material. After comparing their data with experiments available from the literature, they came to the same conclusion as in cases 1 and 2 studied before. In their numerical work they used both tetrahedral and hexahedral elements.

Lakehal and Narayanan [100] presented computational micro-fluidic flow simulations to study the effect of two phase flow regimes on heat transfer in small pipes. For tracking the interface they used the LS method. All simulations were done in a 1mm diameter tube, heated at the surface, in which air bubbles were injected into water. The CFD code TransAT was used. They investigated three regimes: bubbly, slug and churn flows. The bubbly flow regime induced a larger heat flux than the slug regime. They compared the Nusselt number for the pure water (single phase) case with that for two-phase (air-water) flow regimes, and this showed that the Nusselt number was twice as high in the two-phase flow regimes.

In summary, it can be said that the VOF method is more used for modeling of two phase flows than the level set method although is more complicated to implement in 3D and the curvature and normal to the interface are poor calculated using the discontinuous VOF function. The main advantage of the VOF method is that it conserves mass up to the machine precision.

2.1.3 Coupled level set / volume of fluid method

To overcome the drawbacks of LS and VOF methods, several researchers have used a coupled level set /volume of fluid (CLSVOF) method. These papers are discussed below.

Sussman and Puckett [27] presented a CLSVOF method for computing incompressible two-phase flows. They developed this method in order to simulate microscale jetting devices such as ink-jet printer heads. In their technique, the volume fractions were defined in terms of the LS function, which were then used to compute the curvature of the interface and, consequently, the surface tension force in the momentum equation. They used standard 2D benchmark tests in order to validate their method, such as the translation of a circle in a periodic domain and the rotation of a notched disk. They also made simulations of a 3D bubble rising in a stagnant liquid. They compared the performance of CLSVOF with LS, VOF and boundary integral methods. Their coupled technique was at least equal to and in most cases better than these standard methods.

Son and Hur [28] presented a coupled CLSVOF method for buoyancy-driven motion of fluid particles. A cubic-interpolated propagation (CIP) scheme was used to accurately calculate the advection terms in the momentum equation. The surface tension force was implemented as a body force in the momentum equation. Son and Hur applied their CLSVOF method to the buoyancy driven motion of gas bubbles and liquid drops in a stationary fluids. All cases studied were 2D or 2D axisymmetric. The numerical results showed that for all cases the particle volume was conserved within 0.01% and that the calculated terminal velocities of the gas bubbles and liquid drops were found to compare well with the numerical data reported in literature.

Son [29] continued their previous work [28] and extended the CLSVOF method to 3D incompressible two-phase flows. The momentum equation was solved with a CIP scheme, while for the advection of the level-set function an essentially non-oscillatory (ENO) scheme was used. Then, he applied this method to bubbles rising in a liquid and to droplets adhering to a vertical wall. The results obtained agreed well with data available from the literature regarding the bubble shape, the bubble rise velocity and the droplets adhering to a vertical wall.

Sussman [4] presented a CLSVOF method for computing the growth and collapse of vapor bubbles. Like in [31], the LS function was used to compute the normal to the interface The curvature was computed with second-order accuracy directly from the volume fractions. A Runge-Kutta second-order time discretization scheme was used for the momentum equation, while second order operator split (Strang-splitting) advection algorithm was used for both the LS function and the volume fractions. To test the accuracy of his algorithm, Sussmann made several tests: the standing wave problem (with no surface tension) and a rising inviscid gas bubble (with surface tension) problem. Then he applied his algorithm to the bubble oscillation and collapse problem. He demonstrated the second order accuracy of his algorithm.

Shepel, Smith and Paolucci [26] presented a streamline-upwind-Petrov-Galerkin finite element LS method that was implemented into several CFD codes such as CFX and FIDAP. Based on the ability to access solver variables, the authors proposed two formulations, namely *fluid-fluid* for the CFX software and *fluid-gas* for the FIDAP software. The first approach considered the coupled motion of the two phases, while the second one considered the motion of only the liquid phase. They also studied the effect of the re-initialization of the LS function and several test problems like the broken-dam problem, collapsing cylinder of water, and bubble rising in a viscous fluid.

Yang et al. [30] presented an adaptive CLSVOF (ACLSVOF) method for interfacial flow simulations on an unstructured triangular grid. The LS function was solved using a discontinuous Galerkin finite element method, while the advection of the volume fraction was done using a Lagrangian-Eulerian method. They presented several numerical tests like reconstruction of a linear interface, the reconstruction of a circular interface, the simple translation of a circular body fluid, the rotation of a circular body, the classical Zalesak slotted disk rotating, a single vortex flow and the simulation of drop deformation in an extensional flow. The results they obtained showed that their method was second order accurate, and mass was conserved very accurately.

Sussman et al. [31] developed a sharp interface method for simulating incompressible two-phase flows. Their algorithm coupled the LS and VOF techniques. This retained their individual advantages while overcoming their weaknesses. Notably, an aspect which should be mentioned about their CLSVOF method is that the LS function was used to compute the interface normal, the density and the viscosity used by the Navier-Stokes equations, but it was not used to compute the curvature of the interface like in other CLSVOF methods. They computed the curvature from the volume fractions. Both the LS equation and the VOF equation were discretized in time using a second-order "Strang splitting" where for one time step they swept first in the x direction and then in the y direction, then for the next time step they swept in the y direction and then the x direction. A Crank-Nicolson / TVD Runge-Kutta projection method was used for the Navier-Stokes equations. To demonstrate the accuracy of their method, they did several tests: a static 2D drop, surface tension with zero gravity drop oscillations, a standing wave problem, a traveling wave problem, a Rayleigh capillarity instability problem, 2D and 3D bubbles rising in a viscous Newtonian fluid, and bubble formation. Their results agreed well with other numerical data reported in literature. They also demonstrated that their algorithm provided: (i) improved accuracy over existing two-fluid methods for a given grid, (ii) faster execution than existing methods for a given accuracy, and (iii) robustly computed flows on coarser meshes.

Shepel and Smith [32] presented a new LS method for modeling two-phase incompressible flows with moving interface boundaries for which the initial results were presented in [26]. In this article the authors presented new results for the validation of the FE/FV LS interface tracking method for: filling a container with water, sloshing in an open tank, zero-gravity drop oscillation, a static drop, a gas bubble rising in a viscous fluid and a gas bubble bursting to a free surface. Good agreement was achieved when they compared their results with analytical results or data available in the literature.

In summary, in the last few years, CLSVOF methods have started to appear as an alternative to both standalone LS and VOF methods. By coupling LS and VOF methods one can overcome their weakness, although there is a cost, the CPU time is higher for CLSVOF than for each standalone LS and VOF method.

2.2 Surface tension models

Surface tension is a key parameter in two-phase flows, so a powerful two-phase flow computer code should include a good model to compute the surface tension force.

One of the most used surface tension models is the continuum surface force (CSF) model of Brackbill [45]. In this model, the surface tension effect is treated as a body force. It is distributed within a transition region of finite thickness at the interface and is given by:

$$F_b = \sigma k n \delta \tag{2.1}$$

in case of the level set method, or is given by:

$$F_b = \sigma k \nabla F \tag{2.2}$$

in case of the volume-of-fluid method. In Eq. (2.1) and Eq. (2.2) σ is the surface tension (a physical property), k is the curvature of the interface, n is the normal vector to the interface, F is the volume fraction and δ is the Dirac function. This body force is included in the momentum equation as a source term. This continuum formulation, on one hand, eases the implementation of the surface tension effect where only the VOF or LS function is needed. In problems with complex topological changes, the CSF model is superior to other methods in robustness and versatility. On the other hand, the CSF model has been found to generate vortex-like flows referred to as spurious currents in the literature, in the neighborhood of the interface. When the surface tension forces are dominant, these spurious currents may destroy the interface and cause instabilities at the interface. Many efforts have been made recently to reduce the spurious currents and improve on the modeling of the surface tension. Popinet and Zaleski [101] improved the pressure gradient calculation by a pressure gradient correction procedure in their front tracking method. Meier *et al.* [76] devised a new technique, which used empirical formulas obtained from a database that has been generated and stored in a data bank to determine interface curvature. Mencinger and Zun [3] proposed a new approach, where the body force is first computed at the faces of the control volume and then interpolated at the center of the control volume. In their method, the discretization is such that the body force field and the pressure gradient cancel each other out exactly when the pressure gradients and VOF gradients are discretized in the same manner. Shin *et al.* [102] presented a hybrid technique for the surface tension calculation in the context of front-tracking method, which combined the Lagrangian and the Eulerian representation of surface tension force in a manner analogous to the VOF-CSF form. Their method was based on the premise that the discrete pressure gradient be exactly balanced by the body force term at the discrete level.

Tong and Wang [103] proposed a new algorithm for implementation of the surface tension referred as the pressure boundary method (PBM). The surface tension force is incorporated into the Navier-Stokes equation via a capillary pressure gradient term in a first step of a two-step projection method. For incompressible flows with constant properties the Navier-Stokes equation is given by:

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \tau + \vec{g} \tag{2.3}$$

In Eq. (2.3) $\vec{V}$ is the velocity, ρ is the density, τ is the viscous stress tensor and $\vec{g}$ is the gravitational acceleration. Eq. (2.3) is then discretized as:

$$\frac{\vec{V}^{n+1}-\vec{V}^{n}}{\delta t}=-\nabla\cdot(\vec{V}\vec{V})^{n}-\frac{1}{\rho^{n}}\nabla p^{n+1}+\frac{1}{\rho^{n}}\nabla\cdot\tau^{n}+\vec{g}^{n} \tag{2.4}$$

Tong and Wang [103] decomposed Eq. (2.4) into the following equations using a two-step projection method:

$$\frac{\tilde{\vec{V}}-\vec{V}^{n}}{\delta t}=-\nabla\cdot(\vec{V}\vec{V})^{n}+\frac{1}{\rho}\nabla\cdot\tau+\vec{g}-\frac{1}{\rho^{n}}\nabla p_{1} \tag{2.5}$$

and

$$\frac{\vec{V}^{n+1}-\tilde{\vec{V}}}{\delta t}=-\frac{1}{\rho^{n}}\nabla p_{2} \tag{2.6}$$

with $\tilde{\vec{V}}$ is the intermediate velocity and

$$p^{n+1}=p_1+p_2 \tag{2.7}$$

Before the execution of the two-step projection procedure, p_1 is first computed from the following equation:

$$\nabla\left[\frac{1}{\rho^{n}}\nabla p_{1}\right]=0 \tag{2.8}$$

together with the pressure boundary condition at the interface given by (neglecting viscous effects):

$$p-p_v=\sigma k \tag{2.9}$$

In the first step of the projection method, an intermediate velocity field $\tilde{\vec{V}}$ is computed from Eq. (2.5). The second step involves taking the divergence of Eq. (2.6) while imposing the incompressibility condition on the velocity field $\vec{V}^{n+1}$. This results in the following Poisson equation for p_2:

$$\nabla\left[\frac{1}{\rho^{n}}\nabla p_{2}\right]=\frac{\nabla\cdot\tilde{\vec{V}}}{\delta t} \tag{2.10}$$

The velocity field $\vec{V}^{n+1}$ is then updated via Eq. (2.6) and the pressure p^{n+1} from Eq. (2.7). The solution of Eq. (2.8) with the sharp pressure boundary condition is a key element of the PBM algorithm.

2.3 Heat transfer with phase change

Below, a brief overview of the status of the numerical simulation of boiling is presented. Emphasis is given to bubble grow dynamics.

Boiling heat transfer is one of the most efficient modes of heat transfer. Pool boiling processes can be divided into several distinct regimes, such as partial nucleate boiling, fully developed nucleate boiling, transition boiling and film boiling. The most widely studied regime is nucleate boiling. Fundamental studies have been carried out on this process for over 50 years. Numerical simulation of the process has only started to advance on this topic in the last decade or so, with Dhir's team leading the way. The review below is limited to their work.

Dhir [104] identified four basic mechanisms that contribute to the total boiling heat flux under pool boiling condition. These are: transient conduction at the area of influence of a bubble growing at a nucleation site, evaporation at a vapor liquid interface, enhanced natural convection on the region in the immediate vicinity of a growing bubble and natural convection over the area that has no active nucleation site and is totally free of the influence of the former three mechanisms. He also mentioned that these mechanisms depend strongly on the magnitude of the wall superheat and other system variables. Ramanujapu and Dhir [105] experimentally studied the formation of mushroom type bubbles during nucleate boiling. They observed stems formed under merged bubbles and the entire vapor "cloud" detached from the nucleating cavity at the base, as in the case of a single bubble. They concluded that merger of vapor bubbles leads to premature removal of vapor bubbles from the surface thereby increasing the frequency of bubble departure.

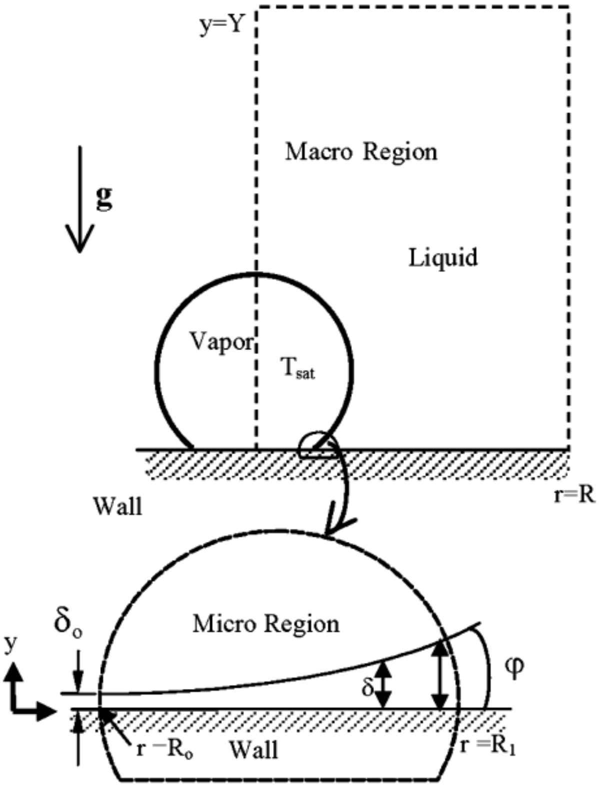


Figure 2.1: Micro and macro regions [1]

Son *et al.* [1] performed a complete numerical simulation of a growing and departing bubble on a horizontal surface. In analyzing the growth of a single bubble, they divided the computational domain into micro and macro regions as can be seen in Fig. (2.1). The micro region contains the thin film that forms underneath the bubble whereas the macro region consists of the bubble and the liquid surrounding the bubble. For flow of the liquid microlayer they used lubrication theory. Therefore the equation of mass conservation in the microlayer was written as:

$$\frac{\partial \delta}{\partial t} = v_l - q/\rho_l h_{fg} \tag{2.11}$$

In Eq. (2.11) the liquid velocity normal to the vapor-liquid interface,v_l, was obtained from the continuity equation:

$$v_l = -\frac{1}{r}\frac{\partial}{\partial r}\int_0^\delta r u_l dy \tag{2.12}$$

The momentum equation for the microlayer was written as

$$\frac{\partial p_l}{\partial p} = \mu_l \frac{\partial^2 u_l}{\partial y^2} \tag{2.13}$$

The energy equation for the film yielded:

$$q = k_l(T_w - T_{int})/\delta \tag{2.14}$$

where in Eq. (2.14) T_{int} is the temperature at the interface. Using a modified Clausius Clapeyron equation, the evaporative heat flux was written as:

$$q = h_{ev}\left[T_{int} - T_v + (p_l - p_v)\, T_v/\rho_l h_{lg}\right] \tag{2.15}$$

In Eq. (2.15) h_{ev} is the heat transfer coefficient given in Eq. (2.16) and h_{fg} is the latent heat.

$$h_{ev} = 2\left(M/2\pi\hat{R}T_w\right)^{0.5} \rho_v h_{fg}^2/T_v \tag{2.16}$$

In Eq. (2.16) M is the molecular mass and $\hat{R}$ is the universal gas constant. The pressure in the liquid and the vapor phase are related as:

$$p_l = p_v - \sigma k - \frac{A}{\delta^3} + \frac{q^2}{\rho_v h_{fg}^2} \tag{2.17}$$

In Eq. (2.17) A is the dispersion constant relating disjoining pressure to the film thickness. The combination of mass momentum and energy equation for the microlayer yielded:

$$\delta'''' = f(\delta, \delta', \delta'', \delta''') \tag{2.18}$$

For the macro region they used the level set method to capture the interface. The mass conservation equation including the effect of volume expansion due to liquid-vapor phase change was derived from the conditions of the mass continuity and energy balance at the interface:

$$\mathbf{m} = \rho\left(\mathbf{u}_{int} - \mathbf{u}\right) = k\nabla T/h_{fg} \tag{2.19}$$

$$\nabla \cdot \mathbf{u} = -\frac{1}{\rho}\left(\frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla \rho\right) + V_{micro} \tag{2.20}$$

In Eq. (2.20) V_{micro} is the rate of vapor volume production from the microlayer and was obtained from the microlayer solution as:

$$V_{micro} = \int_{R0}^{R1} \frac{k_l\left(T_w - T_{int}\right)}{\rho_v h_{fg} \delta \Delta V_{micro}} r dr \tag{2.21}$$

where ΔV_{micro} is a vapor-side control volume near the micro region.

The bubble growth predicted from their numerical analysis has been found to compare well with the experimental data reported in the literature and that obtained in their work.

Mukherjee and Dhir [106] numerically studied the bubble dynamics and heat transfer associated with lateral bubble merger during transition from partial to fully developed nucleate boiling. The effect of microlayer evaporation [1] was also incorporated. They used the level set method to capture the interface between the two fluids. Calculations were carried out for multiple bubble mergers in a line and also in a plane and the bubble dynamics and wall heat transfer were compared to that for a single bubble. Their results showed that merger of multiple bubbles significantly increases the overall wall heat transfer rate. Good agreement with data from experiments was found in bubble growth rate and bubble shapes obtained from numerical simulations.

2.4 Contact angle

A brief overview of the contact angle, both experimentally and numerically is presented in this section.

Jiang *et al.* [107] proposed a correlation based on the available experimental data of the advancing dynamic contact angle measured through the liquid phase during the displacement of a liquid-gas interface inside a glass capillary tube. They suggested that the dynamic contact angle is independent of the macroscopic geometry in which the measurement is made and depends only upon the properties of the materials involved and the contact line velocity.

Hocking [108] derived the macroscopic contact angle in an evaporating droplet based on the assumption that its deviation from the microscopic angle is accounted for by the processes taking place in the slip region. Based on this, Hocking [108] has been shown that the predicted change in contact angle in a spreading droplet is in good agreement with experimental values. He proposed an expression for the change in contact angle for a steady evaporating droplet.

Renardy *et al.* [109] implemented moving contact line in a volume-of-fluid scheme with piecewise linear interface construction. They treated the interfacial tension as a continuous body force, computed from numerical derivatives of a smoothed volume-of-fluid function. They investigated two methods for implementing the contact angle: one which extrapolates the volume-of-fluid function beyond the flow domain, and the other which treated the problem as a three-phase situation. They found that the latter approach introduced an artificial localized flow and they recommended the use of the former approach.

Friess and Lakehal [110] proposed a numerical model for treating interfacial tension in the simulation of two-phase flow with particular consideration of triple lines. The model they proposed is based on the PLIC–VOF method for interface tracking and was implemented only in 2D. The novel idea was to introduce a "triple line term" in the momentum equation which accounted for the instantaneous angle between the liquid-gas interface and the solid wall. Three test problems were considered, namely: forced and stationary flow, gravity-driven and stationary flow, and gravity-driven but not stationary flow. For the first problem, a comparison with experimental data was possible by using Hoffmann's correlation. They found that the ratio between their numerical values for the difference between dynamic and static contact angle and those from experiments ranged between 0.26 and 1.6. They found no satisfactory explanation, however, one possible reason could be the discrepancy between the We number of their simulation (1) and the experiment (ranged between 10^{-11} and 10^{-3}). They also considered a 2D liquid drop on a vertical wall and a water drop impacting on a horizontal surface, respectively.

Theodorakakos *et al.* [111] investigated numerically and experimentally the detachment of liquid droplets from porous material surfaces under the influence of a cross-flowing air. They measured the static and dynamic contact angles for different droplet sizes, air velocities and porous materials. They used the VOF method for predicting the droplet surface deformation and further detachment. The results showed good agreement between model predictions and experimental data for three different porous materials. They also studied numerically the effect of water flow connecting the liquid droplet with the porous material, the effect of gravity and the effect of water temperature.

Narayanan and Lakehal [112] presented a new dynamic contact angle model for two-phase flow. This model did not require the specification of a contact angle or contact-line velocity. For interface capture they used the LS method. Their model included a triple line force in the momentum equation based on a consideration of interfacial free energy. In order to demonstrate the capability of their model, two test cases of a droplet immersed in a different fluid were considered. The first case: same density, different viscosity; the second case: different density, same viscosity. Then, a practical application was considered: the liquid filling a prototypical microdevice with different equilibrium contact angles (70°, 90° and 110°). The results they obtained, showed that such an approach was a useful design tool.

Mukherjee and Kandlikar [113] studied numerically a vapor bubble growing on a heated wall using a static contact angle model and a dynamic contact angle model based on the contact line velocity and the sign of the contact line velocity. The complete Navier-Stokes equations were solved and the liquid vapor interface was captured using the level set method. They found that there was a little effect on the vapor volume growth rate whether a static or a dynamic contact angle was used.

Vadgama and Harris [114] measured the quasi-static advancing contact angles of refrigerant R134a on copper and aluminum surfaces over a temperature range from 0 to 80°C. They measured the contact angle using a direct optical observation technique where the liquid meniscus at the surface of a vertical plate was captured using a high magnification camera system. The values of the contact angle were found to vary between 8.3° and 5.6° for aluminum and between 5.1° and 6.5° for copper.

Choi and Bussmann [115] presented a methodology to volume tracking three materials that, in contrast with the 'onion-skin' methodology, assumed the existence of a 'triple point' at which two interfaces between three materials intersect. The methodology is iterative: it locates a point of intersection between two interfaces that minimizes an error expression. By using various advection tests, they shown that the algorithm maintained a sharp representations of triple points in translation, rotation and shear.

Fang *et al.* [116] proposed a model, within the framework of a volume-of-fluid approach, to account for surface adhesion forces by considering the effect of contact angle hysteresis. A transient model was developed by correcting boundary force balances through specification of the local contact angle and instantaneously updating the local angle values based on the variation of the volume fraction from the previous time steps. They obtained excellent agreement of the calculated liquid slug dimensions at detachment with those measured in the experiment.

Lim *et al.* [117] studied the spreading and evaporation of a small droplet ($17 - 50\mu m$ diameter) of water and ethylene glycol jetted on the heated substrate. According to their experimental results, the interfacial oscillation phenomena of water droplet whose Ohnesorge number is about 0.01 is similar to that in inviscid impact driven region, while that of ethylene glycol droplet (Ohnesorge number 0.1) is similar to that in highly viscous impact driven region followed by capillary driven extra spreading. They found that in the case of water droplet, the spreading processes ended before the evaporation became significant.

Grundke *et al.* [118] measured the surface tension and contact angle for different materials. They obtained the lowest contact angle on the polyether surface compared to the silicones surfaces. They also found out that the surface wettability plays an important role for the contact angle.

Manservisi and Scardovelli [119] presented a numerical simulation of the spreading of the single droplet impacting over horizontal dry surfaces. They presented a new variational approach by coupling an interface front-tracking algorithm to the single fluid finite element formulation of the incompressible Navier-Stokes equations which were solved on a fixed mesh. Their model have been tested for the impact and the spreading of a droplet on solid substrates with a different wettability at low Reynolds numbers. They have shown that the dynamic contact angle model is particulary important after the initial spreading phase, when the contact point is standing.

In summary, the contact angle, like the surface tension force, is one of the key parameters for modeling of two-phase flow. The contact angle is strongly influenced by the solid surface, as well as its wettability. Several correlations and models exist in the literature for predicting the dynamic contact angle. In the present study the model of Fang *et al.* [116] was adopted and implemented into FLUENT. The author have chosen the model of Fang *et al.* due to its complexity: it involves interface reconstruction, correlations to predict dynamic advancing and receeding contact angles and a relation between VOF function and normal to the interface for the calculation of the dynamic contact angle.

2.5 Conclusions

The advantages and disadvantages of different one-fluid methods are summarized in Table (2.1).

- The LS method has recently became a popular method to deal with two-phase flows because of its advantages in automatic handling of topological changes (such as merging and pinching) as well as robust geometric information (such as normals and curvatures). Moreover, the extension to 3D is quite straightforward as opposed to the VOF method;

- The main drawback of the LS method is that it does not always conserve mass. To overcome this aspect a re-initialization equation should be solved, preferably at the end of every time step. Sussman and Fatemi [25] added a ‘constraint’ to the re-initialization equation in order to improve the conservation of mass;

- Some authors recently proposed a coupled LS /VOF method (CLSVOF), which was superior to either standalone LS or VOF methods. The main idea is to solve both VOF and LS advection equations and to then compute the normals and the curvatures from the LS function instead of using the discontinuous VOF fractions;

- In the CLSVOF method of Sussman et al. [4,27] they used the LS function only to compute the normals to the interface and not the curvatures which were computed with the VOF fractions;

- Another interesting method was proposed by van der Pijl et al. [47] where they “coupled” the LS method with the VOF method, but they used the volume of fluid function just to ensure mass conservation, avoiding solving the expensive reconstruction step of the interface from the VOF fractions;

- Two approaches were observed in treating physical properties such as viscosity and density, near the interface. The first one introduces a finite thickness of the interface on the order of 3 to 4 mesh distances and uses a smoothed Heaviside function to interpolate the physical properties near the interface. Another approach is to consider that interface to have zero thickness and uses the jump conditions for the physical properties;

- It was also observed that an increasing number of the authors use the finite element method to compute the solution of incompressible two-phase flows.
- There are many models for implementing the surface tension effect in a numerical code. Since the CFD code FLUENT is used for this study, and therefore the access to the core of the code is somehow limited, the continuum surface force model of Brackbill [45] is implemented for this study. This choice was also based on the fact that the volume-of-fluid method existed in FLUENT also used the CSF model.
- Numerical studies on flow boiling have started in the last few years. Some authors divided the computational domain into two regions: microregion containing a thin liquid film and a macroregion containing the bubble and the liquid surrounding the bubble. In this study, due to the use of a commercial code, the microregion could not be considered. Instead the VOF method will be applied.

Method	Advantages	Disadvantages
Front tracking	• Extremely accurate, • Robust, • Accounts for substantial topology changes in interface, • Merging and breakage of interfaces does not occur automatically.	• Mapping of interface mesh onto Eulerian mesh, • Dynamic remeshing required, • Merging and breakage of interfaces requires subgrid model, • Numerical inaccuracy for complicated geometries.
LS	• Conceptually simple, • Easy to implement.	• Limited accuracy, • Loss of mass (volume).
Marker particles	• Extremely accurate, • Robust, • Accounts for substantial topology changes in the interface.	• Computationally expensive, • Redistribution of marker particles required.
SLIC VOF	• Conceptually simple, • Straightforward extension to 3D.	• Numerical diffusion, • Limited accuracy, • Merging and breakage of interfaces occurs automatically.
PLIC VOF	• Relatively simple, • Accurate, • Accounts for substantial topology changes in interface.	• Difficult to implement in 3D, • Merging and breakage of interfaces occurs automatically • Rather poor calculation of k and n.
Lattice Boltzmann	• Accurate, • Accounts for substantial topology changes in interface.	• Difficult to implement, • Merging and breakage of interfaces occurs automatically.

Table 2.1: Advantages and disadvantages of ‘one fluid’ methods

Based of these very recently concluded studies and these conclusions, the present study have tackled the development and the implementation of a CLSVOF method, a dynamic contact angle model and a heat transfer model with phase change within the framework of the commercial CFD code FLUENT.

Chapter 3

Theoretical models

In this chapter, the models to be implemented and/or coupled with FLUENT are presented.

3.1 The level set method

3.1.1 Generalities

In the recent years, the level set (LS) method has started to be used to model incompressible two phase flows for tracking the interface between two fluids [22, 120]. The basic idea behind this method is to consider a smooth continuous scalar function ϕ, which is zero at the interface, positive in one phase and negative in the other phase. Usually, ϕ is initialized as the signed minimum distance function to the interface, so $|\nabla\phi| = 1$ for the whole domain. Later on, when we need to solve the re-initialization equation, $|\nabla\phi| = 1$ is true only at the interface. In case of an arbitrary initial interface, the re-initialization equation should be solved at the beginning of the calculation to ensure that at least near the interface the level set function is a signed distance. This method is not available in FLUENT and will be added using UDF's (described in Chapter 4).

When the interface is advected by an external velocity field, the evolution of the level set function is given by:

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = 0 \tag{3.1}$$

3.1.2 The conservation equations

For two-phase flows, the Navier-Stokes equation has the following form:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu\mathbf{D}) + \mathbf{F}_{st} + \rho\mathbf{g} \tag{3.2}$$

and the continuity equation is:

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

In Eq. (3.2) and Eq. (3.3), $\mathbf{u}$ is the velocity vector, ρ is the density, t is the time, μ is the dynamic viscosity, $\mathbf{D}$ is the rate of deformation tensor with the components $D_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$, $\mathbf{F}_{st}$ is the body force due to the surface tension, and $\mathbf{g}$ is the gravity vector.

Eq. (3.2) and Eq. (3.3), which are the conservation equations for two-phase flow, look similar to those for single phase flow with the following exceptions: the momentum equation has a surface tension term, and ρ and μ are discontinuous. In the LS method, the density and the dynamic viscosity are described by the following formulae:

$$\rho(\mathbf{x}, t) = \rho_g + (\rho_l - \rho_g) H_\epsilon(\phi(\mathbf{x}, t)) \tag{3.4}$$

$$\mu(\mathbf{x}, t) = \mu_g + (\mu_l - \mu_g) H_\epsilon(\phi(\mathbf{x}, t)) \tag{3.5}$$

where the subscripts l and g denote liquid and gas phases, respectively. $H_\epsilon(\phi)$ is the smoothed Heaviside function which is used instead of the discontinuous Heaviside function for numerical stability reasons. This smoothed Heaviside function was first proposed by Sussman et al [23] and has the following form:

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if} \quad \phi < -\epsilon \\ (\phi + \epsilon)/(2\epsilon) + \sin(\pi\phi/\epsilon)/(2\pi) & \text{if} \quad |\phi| \leq \epsilon \\ 1 & \text{if} \quad \phi > \epsilon \end{cases} \tag{3.6}$$

where ϵ is a parameter, whose value ranges from one to two times the local mesh size close to the interface. Physically, ϵ represents half of the interface thickness.

The surface tension term in Eq. (3.2) is approximated using the CSF (Continuum Surface Force) model of Brackbill *et al.* [45] using the following relation:

$$\mathbf{F}_{st} = \sigma k(\phi) \mathbf{n} \delta_\epsilon(\phi) \tag{3.7}$$

where σ is the surface tension coefficient, $k(\phi)$ is the mean curvature of the interface, and $\mathbf{n}$ is the normal vector of the interface, as defined by:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \qquad k(\phi) = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$$

Furthermore, $\delta_\epsilon(\phi)$ is the smoothed delta function, which is defined as the derivative of $H_\epsilon(\phi)$ with respect to ϕ.

3.1.3 Re-initialization

One problem with the LS method is that, even after the first time step integration of Eq. (3.1), the level set function ϕ ceases to be a signed distance function to the interface. Consequently, the level set formulation needs to reinitialize ϕ regularly, preferably at every time step [22]. To do this, we should solve Eq. (3.8) for the steady state solution:

$$\phi_\tau + \operatorname{sign}(\phi_0)(|\nabla\phi| - 1) = 0, \qquad \phi(\mathbf{x}, 0) = \phi_0 \tag{3.8}$$

For numerical purposes, it is useful to smooth the sign function with:

$$\operatorname{sign}_\epsilon(\phi_0) = 2\left[H_\epsilon(\phi_0) - 1/2\right] \tag{3.9}$$

It is evident that we need the level set function to be a signed distance just near the interface, so usually Eq. (3.8) is solved for $\tau = 0 \cdots \epsilon$, with τ being the integration time. To discretize the gradients in Eq. (3.8) one needs to use a high order scheme like ENO or WENO [121], otherwise oscillations of the interface occur. The WENO scheme, which is discussed in Appendix A, is implemented in our code only for Cartesian grids.

Eq. (3.8) can now be written as:

$$\frac{\partial\phi}{\partial t} + w \cdot \nabla\phi = \operatorname{sign}\phi_0 \tag{3.10}$$

with

$$w = \operatorname{sign}\phi_0 \frac{\nabla\phi}{|\nabla\phi|}$$

where w is the characteristic velocity pointing outward from the free surface.

To discretize the “convective” flux in Eq. (3.8), the Godunov [122] formulation was used:

$$G(\phi)_{i,j,k} = \begin{cases} \sqrt{\max((a^+)^2,(b^-)^2) + \max((c^+)^2,(d^-)^2) + \max((e^+)^2,(f^-)^2)} - 1 & \text{if} \quad \phi^0_{ijk} > 0 \\ \sqrt{\max((a^-)^2,(b^+)^2) + \max((c^-)^2,(d^+)^2) + \max((e^-)^2,(f^+)^2)} - 1 & \text{if} \quad \phi^0_{ijk} < 0 \\ 0 & \text{if otherwise} \end{cases} \tag{3.11}$$

where

$$a = \phi^-_{x,i,j,k} \qquad b = \phi^+_{x,i,j,k}$$

and

$$a^+ = \max(a, 0) \qquad a^- = \min(a, 0)$$

with a and b computed with a fifth order WENO scheme. Similar definitions were used for c, d, e and f.

The time integration of Eq. (3.8) was done explicitly using the Euler method:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = L(\phi^n) \tag{3.12}$$

with $L(\phi)$ being a general function of ϕ and gradients of ϕ.

3.2 The volume of fluid method

The basic idea of the volume of fluid (VOF) method is to consider a color function, defined as the volume fraction of one of the fluids within each cell, to capture the interface. This function will be one if the cell is filled with the gas phase, zero if the cell is filled with the liquid phase, and between zero and one in the cells where there is an interface. The VOF method belongs to the so called “one” fluid method, where a single set of conservation equations is solved for the whole domain, see Eq. (3.2) and Eq. (3.3). In these equations, $\mathbf{u}$ is the velocity vector, ρ is the density, t is the time, μ is the dynamic viscosity, $\mathbf{D}$ is the rate of deformation tensor with the components $D_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$, $\mathbf{F}_{st}$ is the body force due to the surface tension, and $\mathbf{g}$ is the gravity vector.

In the VOF method, the density and the dynamic viscosity are described by the following formulae:

$$\begin{aligned} \rho(\mathbf{x}, t) &= \rho_l + (\rho_g - \rho_l)F \\ \mu(\mathbf{x}, t) &= \mu_l + (\mu_g - \mu_l)F \end{aligned}$$

where l and g denote liquid and gas phases, respectively; and F is the gas volume fraction.

After solving Eq. (3.2) and Eq. (3.3), the color function is advected with the velocity field:

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \tag{3.13}$$

In general, a VOF algorithm solves the problem of updating the volume fraction field F given the fixed grid, the velocity field $\mathbf{u}$, and the field F at the previous step. In 2D, the interface is considered to be a continuous, piecewise smooth line. In order to reconstruct the interface, one needs to first determine which cells contain the interface, and then to determine the location in these cells by considering the volume fraction in the neighboring cells. The simplest type of VOF methods are the Simple Line Interface Calculation (SLIC) [123] or the SOLA-VOF algorithm [18]. They are first order accurate for the reconstruction of the interface. Typically, the reconstructed interface is made up of a sequence of segments aligned with the grid. Fig. (3.1) shows the exact color VOF function for a smooth circular arc, while Fig. (3.2(a)) shows the interface reconstruction using the SLIC method. More accurate VOF techniques attempt to fit the interface through piecewise linear segments. They are now known as Piecewise Linear Interface Calculation (PLIC) [20]. Fig. (3.2(b)) shows a reconstructed interface using the VOF/PLIC method.

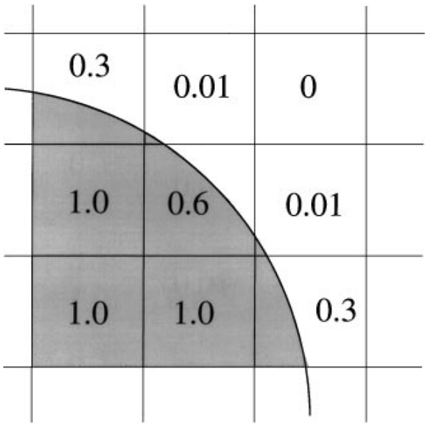
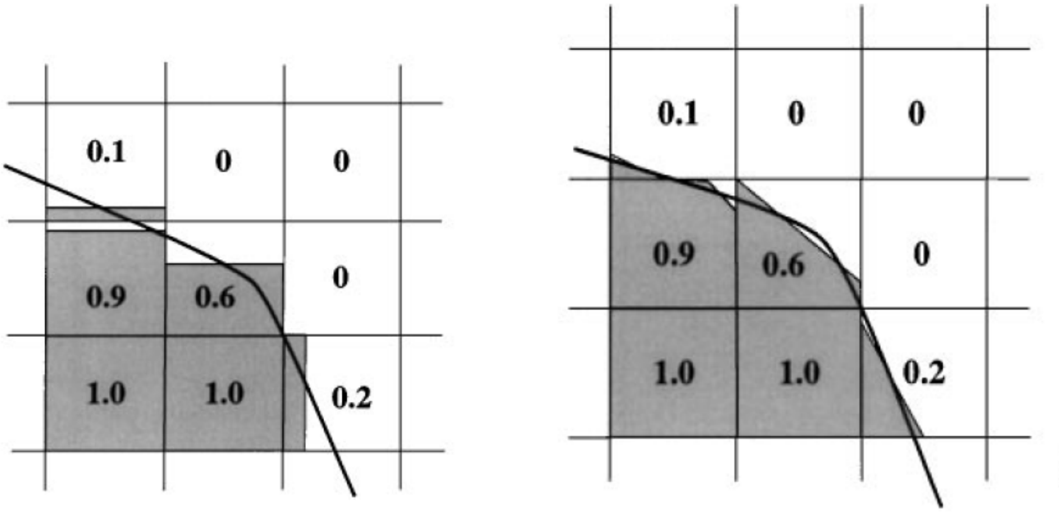


Figure 3.1: The exact VOF color function for a circular arc over a square grid.



(a) SLIC

(b) PLIC

Figure 3.2: Interface reconstruction schemes.

Because F in Eq. (3.13) is not a continuous function, the VOF/PLIC algorithm is divided in two parts: a reconstruction step and a propagation step. The key part of the reconstruction step is the determination of the orientation of the segment. This is equivalent to the determination of the unit normal vector $\mathbf{n}$ to the segment. Then the normal vector $\mathbf{n}$ and the volume fraction F uniquely determine the straight line. Appendix B shows the calculation of interface position in 2D and 3D. The second step of the VOF algorithm is the propagation. Usually, the propagation step is done using a fractional step or an operator split method which updates F by advecting the interface along one spatial direction at a time. Intermediate F values are calculated during this process, and the final F field is obtained only after advection of the interface along all coordinate directions. Two types of approach are often used, namely the Lagrangian approach and the Eulerian approach. For the Eulerian approach, Eq. (3.13) first is written in a more useful form:

$$\frac{\partial F}{\partial t} + \nabla \cdot \mathbf{u}F = F\nabla \cdot \mathbf{u} \tag{3.14}$$

then Eq. (3.14) is advected as follow:

$$\frac{F^{*} - F^{n}}{\delta t} + \frac{\partial uF^{n}}{\partial x} = F^{*}\frac{\partial u}{\partial x} \tag{3.15}$$

$$\frac{F^{**} - F^{*}}{\delta t} + \frac{\partial vF^{*}}{\partial y} = F^{*}\frac{\partial v}{\partial y} \tag{3.16}$$

$$\frac{F^{n+1} - F^{**}}{\delta t} + \frac{\partial wF^{**}}{\partial z} = F^{*}\frac{\partial w}{\partial z} \tag{3.17}$$

The volume fraction at time n+1 is obtained with Eq. (3.15), Eq. (3.16) and Eq. (3.17) by using an Eulerian operator split method. In the Lagrangian approach, the motion of the interface segments is computed directly. For each cell, three contributions are calculated: the area fluxes $VL_{i,j}$ and $VR_{i,j}$ entering the cell (i,j) from cells (i-1,j) and (i+1,j, and the area $VC_{i,j}$ of the fluid contained at the beginning of the step in the control cell which remains there, see Fig. (3.3). The updated value of volume fraction is then given by:

$$F_{i,j} = VL_{i,j} + VC_{i,j} + VR_{i,j} \tag{3.18}$$

The interface is then advected in a similar way, with the starting sweep direction being alternated at every time step.

Scardovelli and Zaleski [2] proposed a mixed advection scheme, where a Lagrangian step is performed in one direction and a Eulerian step in the other direction.

Unsplit algorithms are geometrically more complex, because they also need to take into account fluxes along the transverse direction.

The surface tension term in Eq. (3.2) is approximated using the Continuum Surface Force model of Brackbill *et al.* [45] by the following relation:

$$F_{st} = \frac{\rho k \nabla F}{\frac{1}{2}\left(\rho_l + \rho_g\right)} \tag{3.19}$$

where in Eq. (3.19) ρ is the discontinuous density, k is the curvature of the interface, and ρ_l and ρ_g are the liquid and, respectively, the gas densities.

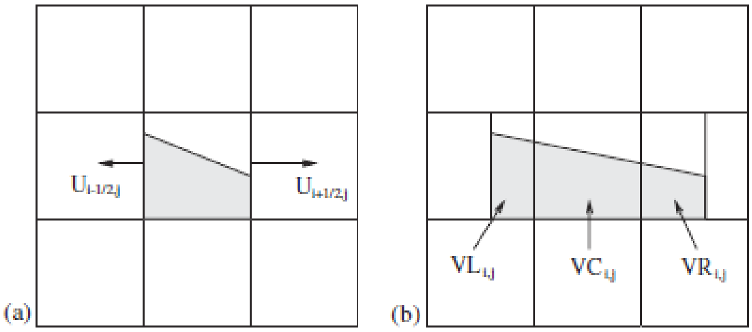


Figure 3.3: Lagrangian advection of the interface along the x-direction: (a) interface reconstruction in the central cell before advection; (b) contributions along the x-direction of the central cell after advection. From [2]

3.3 Dynamic contact angle

In this section a 3D dynamic contact angle model is described, based on volume fraction, interface reconstruction, and experimentally available advancing and receding static contact angles from the literature. To define the advancing contact angle, one needs to dynamically add volume to a droplet to the maximum volume permitted without increasing the interfacial area between the liquid and solid phases. For the definition of the receding contact angle, one needs to remove volume from the droplet to the minimum volume permitted without decreasing the interfacial area between the liquid and solid phases. These two angles refer to the static values. When the contact line starts to move, the contact angle will deviate from the equilibrium value and we have the dynamic advancing contact angle and the dynamic receding contact angle.

3.3.1 Mathematical formulation

If the surface tension is assumed to be constant between each pair of the three phases (liquid, gas and solid), the Young's equation reads as (see Fig. (3.4)):

$$\sigma \cos\theta = \sigma_{SV} - \sigma_{sl} \tag{3.20}$$

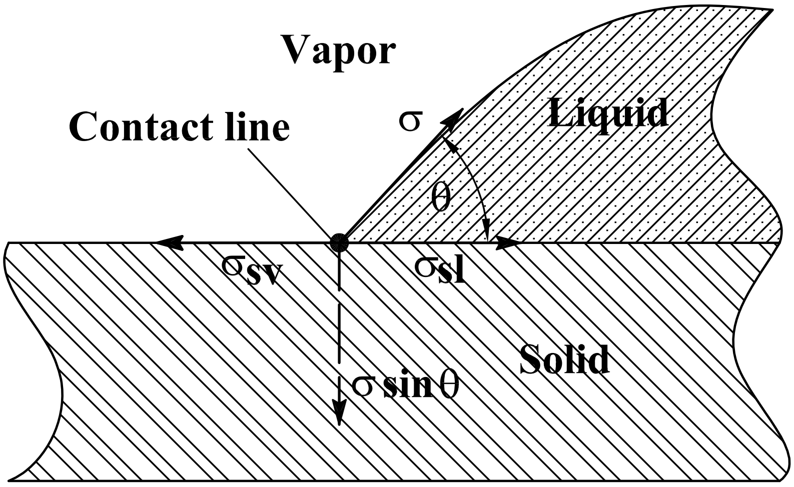


Figure 3.4: Equilibrium contact line and contact angle.

In Eq. (3.20) θ_s refers to the static contact angle, σ_{LG} is the liquid gas surface tension, and σ_{LS} and σ_{GS} refer to the solid-liquid and to the gas-solid surface tensions, respectively. The real dynamic contact angle can take any value within the interval $\theta_R \leq \theta \leq \theta_A$, where θ_R and θ_A are the dynamic receding and the dynamic advancing contact angles. To compute these values, the correlation of Tanner [124] for the receding angle, see Eq. (3.21), and the correlation of Jiang *et al.* [107] for the advancing angle, see Eq. (3.22) were used:

$$\theta_{rec}^3 - \theta_R^3 = 72Ca \tag{3.21}$$

$$\frac{\cos\theta_{adv} - \cos\theta_A}{\cos\theta_{adv} + 1} = \tanh\left(4.96Ca^{0.702}\right) \tag{3.22}$$

where θ_{rec} refers to the static receding contact angle, θ_{adv} refers to the static advancing contact angle, and Ca is the capillarity number given by:

$$Ca = \frac{\mu U_{cl}}{\sigma} \tag{3.23}$$

where in Eq. (3.23) U_{cl} is the contact line velocity, which is computed knowing the contact line position at time t and at time $t + \Delta t$. Once the upper and lower limits of the contact angle are calculated, we can evaluate the contact angle, based on the geometrical reconstruction of the interface.

3.3.2 The contact angle model

The first step in determining the contact angle between the interface and boundary is similar to the reconstruction step from the advection algorithm of the volume fraction F. Given the normal vector to the interface $\mathbf{m} = [m1, m2, m3]$, then the equation of the interface, which is a plane in 3D, is given by:

$$m1x + m2y + m3z = \alpha \tag{3.24}$$

where α is the plane constant. For 2D cases, $m3 = 0$ and thus the interface is a line. Scardovelli and Zaleski [125] presented analytical relations for determining α. These are summarized in Appendix B. Once the plane constant α has been determined, we used the same relation which connects α with the volume fraction F to evaluate the contact angle like as in [116]. Using the following notations : $g = m_2/m_1$, $h = \alpha/m_1$, $f = m_3/m_1$, for 3D we get:

$$F = \frac{1}{6gf}\left[h^3 - \sum_{i=1}^{3} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^3 + \sum_{i=1}^{3} H\left(h - \Delta x_1 - g\Delta x_2 - f\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right) \left(h - \Delta x_1 - g\Delta x_2 - f\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right)^3\right] \tag{3.25}$$

where in Eq. (3.25) H is the discontinuous Heaviside function. For 2D Eq. (3.25) becomes:

$$F = \frac{1}{2g}\left[h^2 - \sum_{i=1}^{2} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^2\right] \tag{3.26}$$

Now observing that $\tan(\theta) = \sqrt{m_1^2 + m_2^2}/m_3$, Eq. (3.25) becomes:

$$F = \frac{\tan\theta}{6g\sqrt{1+g^2}}\left[h^3 - \sum_{i=1}^{3} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^3 + \sum_{i=1}^{3} H\left(h - \Delta x_1 - g\Delta x_2 - \frac{\sqrt{1+g^2}}{\tan\theta}\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right) \left(h - \Delta x_1 - g\Delta x_2 - \frac{\sqrt{1+g^2}}{\tan\theta}\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right)^3\right] \tag{3.27}$$

For the 2D case we have $\tan(\theta) = m_1/m_2$ so Eq. (3.26) becomes:

$$F = \frac{\tan(\theta)}{2}\left[h^2 - \sum_{i=1}^{2} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^2\right] \tag{3.28}$$

Eq. (3.27) and Eq. (3.28) are solved iteratively for θ with a Newton iterative method. The derivative of Eq. (3.27) with respect to θ is given by:

$$\frac{\partial F}{\partial \theta} = A + B + C \tag{3.29}$$

In Eq. (3.29) A, B and C are given by the following relations:

$$A = \frac{1}{6g\sqrt{1+g^2}\cos^2\theta}\left[h^3 - \sum_{i=1}^{3} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^3 + \right.$$
$$\sum_{i=1}^{3} H\left(h - \Delta x_1 - g\Delta x_2 - \frac{\sqrt{1+g^2}}{\tan\theta}\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right)$$
$$\left.\left(h - \Delta x_1 - g\Delta x_2 - \frac{\sqrt{1+g^2}}{\tan\theta}\Delta x_3 + \frac{m_i\Delta x_i}{m_1}\right)^3\right]$$

$$B = -\frac{\tan\theta}{2g\sin^2\theta}\left[H\left(h-f\right)\left(h - \frac{\sqrt{1+g^2}}{\tan\theta}\right)^2\right]$$

$$C = \frac{\tan\theta}{2g\sin^2\theta}\left[H\left(h-g-f\right)\left(h - g - \frac{\sqrt{1+g^2}}{\tan\theta}\right)^2 + \right.$$
$$\left. H\left(h-1-f\right)\left(h - 1 - \frac{\sqrt{1+g^2}}{\tan\theta}\right)^2\right]$$

For the 2D case A, B and C in Eq. (3.29) have the following form:

$$A = \frac{1}{2\cos^2\theta}\left[h^2 - \sum_{i=1}^{2} H\left(h - \frac{m_i\Delta x_i}{m_1}\right)\left(h - \frac{m_i\Delta x_i}{m_1}\right)^2\right]$$
$$B = -\frac{\tan\theta}{\sin^2\theta}H\left(h-g\right)\left(h - \frac{1}{\tan\theta}\right)$$
$$C = 0.0$$

3.4 Heat transfer with phase change

For solving problems involving heat transfer, besides the Navier-Stokes equations Eq. (3.3) and Eq. (3.2), the energy equation is also required:

$$\rho c_p\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T\right) = \nabla\cdot k\nabla T + S_E \tag{3.30}$$

where in Eq. (3.30) ρ is the density, c_p is the specific heat, T is the temperature, $\mathbf{u}$ is the velocity vector, t is the time, k is the thermal conductivity and S_E is the source term. Most previous work assumed that the vapor phase remains at the saturation temperature and therefore Eq. (3.30) was not solved in the gas phase. However, this is not necessarily true, especially if the bubble is in contact with a heated wall. In this study the energy equation is also solved in the gas phase and the temperature of the vapor-liquid interface is maintained at the saturation value.

In cases of phase change, a mass transfer across the interface occurs and this has been handled inside the model. The mass source for the vapor case in case of boiling is:

$$S_G = \frac{(k_l F_l + k_g F_g)\left(\nabla T \cdot \nabla F_l\right)}{h_{fg}} \tag{3.31}$$

and for the liquid phase:

$$S_L = -S_G \tag{3.32}$$

There is also a source term in the energy equation:

$$S_E = -S_G \cdot h_{fg} \tag{3.33}$$

The source terms S_G and S_L belong to the mass conservation equation.

3.5 Summary

A basic overview of the models to be implemented and/or coupled in FLUENT have been described. Chapter 4 gives the details of how this was done for adiabatic flows and describes the process for diabatic flows.

Chapter 4

FLUENT discretization procedure and code development

This chapter gives details of the FLUENT (version 6.3.26 [126]) discretization procedure together with information regarding implementation of the LS method (and the coupling with VOF method), dynamic contact angle model and heat transfer model into FLUENT.

4.1 Fluent discretization description and choices

FLUENT is a collocated (pressure and velocity are discretized on the same grid points) finite volume code, so one needs to solve the integral form of Eq. (3.1) which in conservative form reads as:

$$\int_V \frac{\partial \rho \phi}{\partial t} + \int_V \nabla \cdot (\rho \mathbf{u} \phi) = 0 \tag{4.1}$$

For the unsteady term FLUENT employs a first order Euler discretization. For the convective term, the Green-Gauss theorem is applied and the volume integral is transformed into a surface integral.

FLUENT uses a mid-point rule integration of the surface integral which is second-order accurate. It also provides five schemes to interpolate the face values ϕ_f, namely: first order upwind, second order upwind, power law, QUICK and MUSCL. Because the MUSCL (Monotone Upwind Schemes for Conservative Laws) scheme is less diffusive than the others, it was used in all of our numerical applications to interpolate face values in Eq. (3.1) and Eq. (3.2). In Eq. (3.2), if the pressure field is known, one can solve for the velocity field. However, the pressure field is not known *a priori* and must be obtained as part of the solution. FLUENT offers several pressure-velocity coupling algorithms like SIMPLE (Semi-Implicit Pressure Linked Equations), SIMPLEC (SIMPLE Consistent), and PISO (Pressure Implicit with Split of Operators). Several pressure velocity coupling algorithms are presented in Appendix C. Since FLUENT uses a collocated grid where pressure and velocity are stored at cell-centers, an interpolation procedure is needed to interpolate the pressure face values from the cell-center values. For two-phase flow, FLUENT provides two schemes, namely body-force-weighted which computes the pressure by assuming that the normal gradient of the difference between pressure and body forces is constant, and PRESTO! (PREssure STaggering Option) which uses the discrete continuity balance for a "staggered" control volume about the face to compute the face pressure. In all our simulations the PISO scheme was used for the velocity-pressure coupling, and the PRESTO! scheme for the pressure interpolation.

FLUENT allows the choice from two types of solvers: pressure-based or density-based. The pressure-based approach was developed for low speed incompressible flows, while the density based approach was developed for high-speed compressible flows. In both methods the velocity field is obtained from the momentum equations. In the density-based approach, the continuity equation is used to obtain the density field, while the pressure field is obtained from the equation of state. In the pressure-based approach, the pressure field is determined by solving a pressure correction equation obtained from the continuity and momentum equations.

4.1.1 Pressure-based algorithm

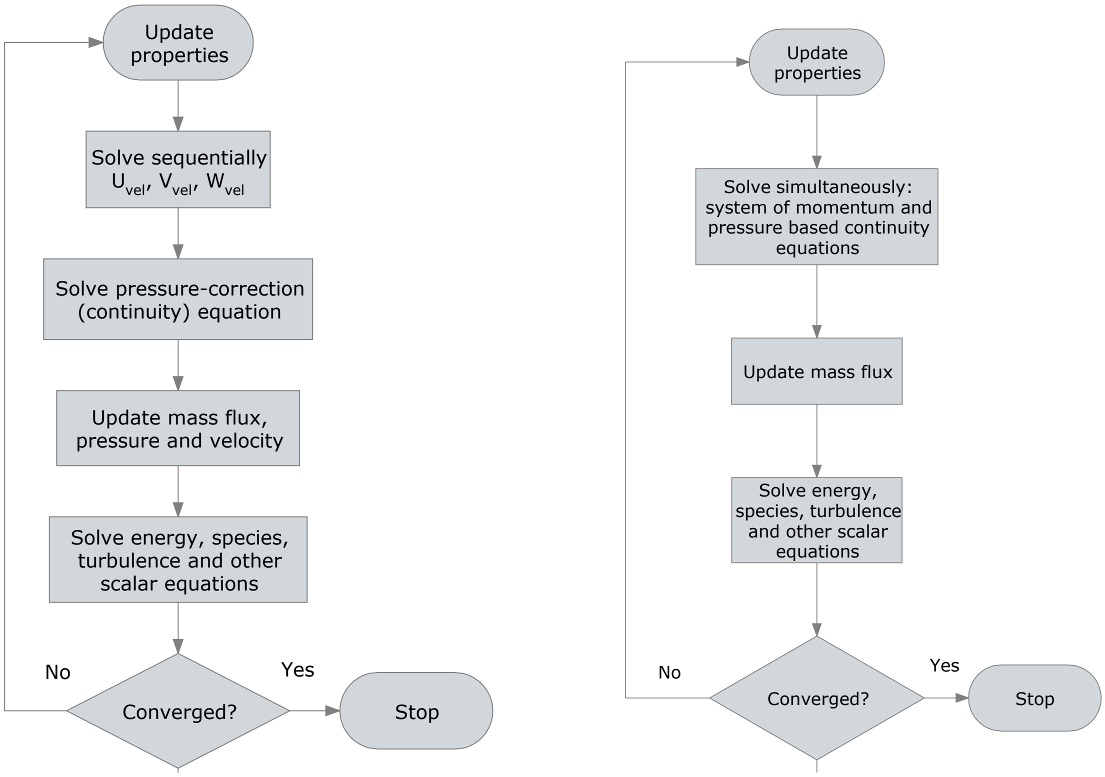
The pressure-based solver employs an algorithm which belongs to a general class of methods called the projection method. Since the governing equations are nonlinear and coupled to one another, the solution process involves iterations wherein the entire set of governing equations is solved repeatedly until the solution converges. FLUENT provides two pressure-based solver algorithms: a segregated one and a coupled one.

Segregated algorithm

The pressure-based solver uses a solution algorithm where the governing equations are solved sequentially (segregated from one another). Because the equations are non-linear, the solution loop must be carried out iteratively in order to obtain a converged solution. The algorithm is illustrated in Fig. (4.1)on the left.

Coupled algorithm

The pressure-based coupled algorithm solves a coupled system of equations comprising the momentum equations and the pressure-based continuity equation. Thus, in the coupled algorithm, steps 2 and 3 in the segregated algorithm are replaced by a single step in which the coupled equations are solved. Since the momentum and continuity equations are solved together, the rate of solution convergence significantly improves when compared to the segregated solver.


Update properties
Solve sequentially U_{vel}, V_{vel}, W_{vel}
Solve pressure-correction (continuity) equation
Update mass flux, pressure and velocity
Solve energy, species, turbulence and other scalar equations
No
Converged?
Yes
Stop
Update properties
Solve simultaneously: system of momentum and pressure based continuity equations
Update mass flux
Solve energy, species, turbulence and other scalar equations
No
Converged?
Yes
Stop

(a) segregated (b) coupled

Figure 4.1: Pressure-based solution methods: (left) segregated algorithm and (right) coupled algorithm.

However, the memory requirement is double that of the segregated algorithm. The algorithm is illustrated in Fig. (4.1)on the right.

4.1.2 Density-based algorithm

The density-based solver solves the governing equations simultaneously. The algorithm is illustrated in Fig. (4.2).

4.1.3 Discretization procedure

A more general integral form of Eq. (3.1) including diffusivity and source terms is given by:

$$\int_V \frac{\partial \rho\phi}{\partial t} dV + \oint \rho\phi\vec{v} \cdot d\vec{A} = \oint \Gamma_\phi \nabla\phi \cdot d\vec{A} + \int_V S_\phi dV \tag{4.2}$$

where ρ is the density, $\vec{v}$ is the velocity vector, $\vec{A}$ is the surface area vector, Γ_ϕ is the diffusion coefficient for ϕ, $\nabla\phi$ is the gradient of ϕ and S_ϕ is the source of ϕ per unit volume.

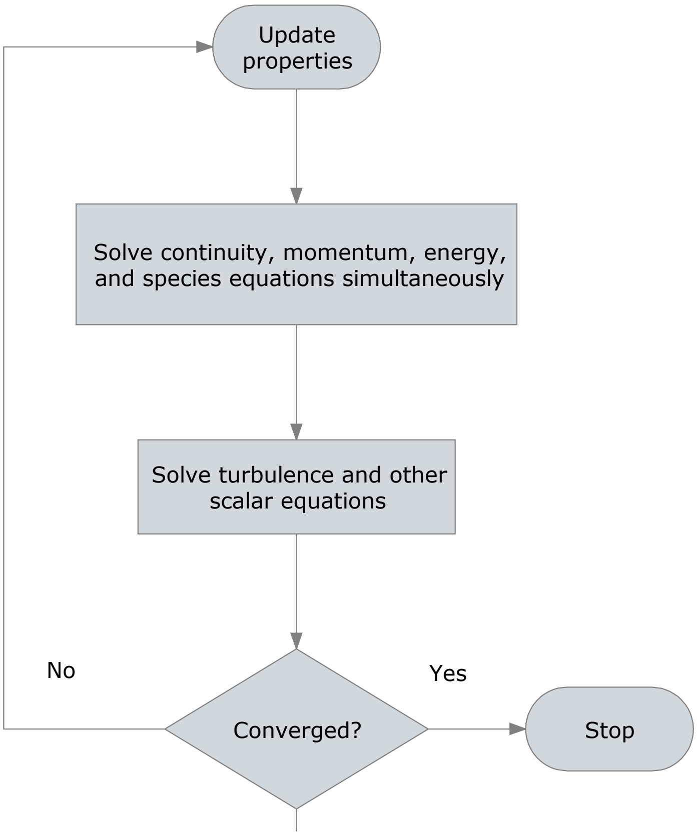

Update properties
Solve continuity, momentum, energy, and species equations simultaneously
Solve turbulence and other scalar equations
No
Yes
Converged?
Stop

Figure 4.2: Density-based solution algorithm

Eq. (4.2) is applied to each control volume in the computational domain. A 2D control volume is shown in Fig. (4.3) and the discretization of Eq. (4.2) yields:

$$\frac{\partial \rho\phi}{\partial t}V + \sum_{f}^{N_{faces}} \rho_f \vec{v}_f \phi_f \cdot \vec{A}_f = \sum_{f}^{N_{faces}} \Gamma_\phi \nabla\phi_f \cdot \vec{A}_f + S_\phi V \tag{4.3}$$

where N_{faces} is the number of faces enclosing the cell, ϕ_f is the value of ϕ convected through face *f*, $\rho_f \vec{v}_f \cdot \vec{A}_f$ is the mass flux through the face, $\vec{A}_f$ is the area of the face, $\nabla\phi_f$ is the gradient of ϕ at face *f*, and V is the cell volume.

As noted earlier in Section 4.1, FLUENT provides five schemes to interpolate the face values ϕ_f from the cell centers. We will review quickly these schemes, while more details can be found in [126] and [127].

First order upwind scheme

With this scheme, quantities at the cell faces are determined by assuming that the cell center values represent a cell-average value and hold throughout the entire cell. Thus, when first-order upwinding is selected, the face value ϕ_f is set equal to the cell-center value of ϕ in the upstream

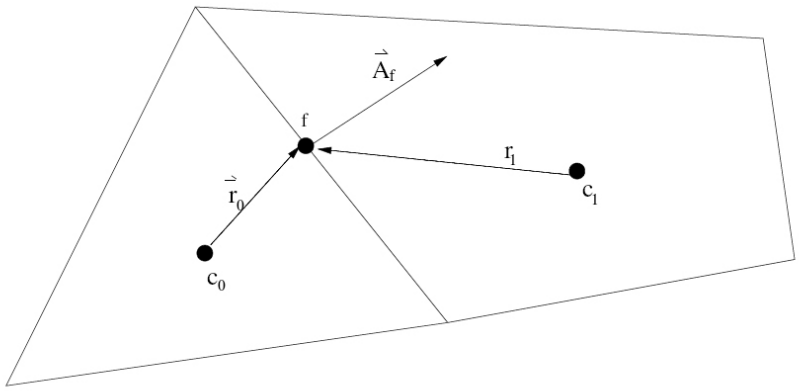


Figure 4.3: A typical control volume.

cell.

Power-law scheme

The power-law scheme interpolates the face value of a variable using the exact solution of the one dimensional convection diffusion equation:

$$\frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}\Gamma\frac{\partial \phi}{\partial x}$$

which when integrated yields the following solution describing how ϕ varies with x:

$$\frac{\phi(x) - \phi_0}{\phi_L - \phi_0} = \frac{\exp(Pe\frac{x}{L}) - 1}{\exp(Pe) - 1} \tag{4.4}$$

where Pe is the Peclet number, $Pe = \frac{\rho u L}{\Gamma}$.

Second-order upwind scheme

When second-order upwinding is selected, the face value ϕ_f is computed using the following expression:

$$\phi_{f,SOU} = \phi + \nabla\phi \cdot \vec{r} \tag{4.5}$$

where in Eq. (4.5) ϕ and $\nabla\phi$ are respectively the cell-centered values and its gradient in the upstream cell, and $\vec{r}$ is the displacement vector from the upstream cell centroid to the face centroid.

QUICK scheme

QUICK-type schemes are based on a weighted average of a second-order upwind and central interpolation of the variable. For the east face 'e' in Fig. (4.4), if the flow is from left to right, such a value can be written as:

$$\phi_e = \theta \left[\frac{S_d}{S_c + S_d} \phi_P + \frac{S_c}{S_c + S_d} \phi_E \right] + (1 - \theta) \left[\frac{S_u + 2S_c}{S_u + S_c} \phi_P - \frac{S_c}{S_u + S_c} \phi_W \right] \tag{4.6}$$

If $\theta = 1$ in Eq. (4.6) we have a central second-order interpolation, while $\theta = 0$ yields a second-order upwind value. A traditional QUICK scheme is obtained by setting $\theta = 1/8$. The QUICK scheme implemented in FLUENT uses a variable, solution dependent value of θ.

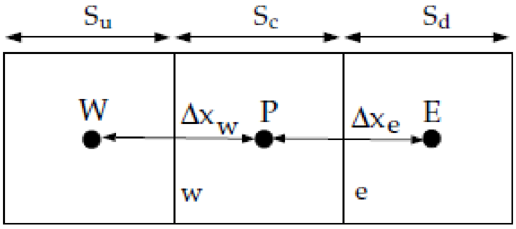


Figure 4.4: One-dimensional control volume

MUSCL scheme

In the third-order MUSCL scheme, the face value of a scalar variable is given by:

$$\phi_f = \theta \phi_{f,CD} + (1 - \theta) \phi_{f,SOU} \tag{4.7}$$

where $\phi_{f,CD}$ stands for face value obtained with a central differencing scheme and $\phi_{f,SOU}$ stands for a face value obtained with a second order upwind scheme. The first term in Eq. (4.7) is written as follows:

$$\phi_{f,CD} = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}(\nabla \phi_0 \cdot \mathbf{r_0} + \nabla \phi_1 \cdot \mathbf{r_1}) \tag{4.8}$$

where ϕ_0 and ϕ_1 are the cell center values of the two neighboring control volumes sharing the same face (see Fig. (4.3)), $\nabla \phi_0$ and $\nabla \phi_1$ are the gradients of the scalar computed at cell centers and $\mathbf{r_0}$ and $\mathbf{r_1}$ are the direction vectors pointing from cell centers to the face center.

A linearized form of Eq. (4.3) can be written as:

$$a_P \phi = \sum_{nb} a_{nb} \phi_{nb} + b$$

where the subscript nb refers to the neighbor cells and a_P and a_{nb} are the linearized coefficients for ϕ and ϕ_{nb}. Similar equations can be written for each cell in the grid. This results in a set of algebraic equations with a sparse coefficient matrix.

4.1.4 Multigrid solver

A point implicit Gauss-Seidel linear equation solver is used in conjunction with an algebraic multigrid (AMG) method to solve the system of dependent linear scalar equations resulting from the previous section. Gauss-Seidel is a robust point implicit solver that rapidly removes local errors in the solution. Global errors, however, are reduced at a rate inversely related to the grid size, which means that the overall solution will be attained slowly. The multigrid method allows these errors to be reduced more quickly. This method depends on the following principles:

- The solution of a problem is solved using a Gauss-Seidel solver until the local errors are removed. Global errors still exist.
- The resulting solution is 'transferred' to a coarser mesh (this process is called restriction), where the former global errors now behave as more local errors.
- Gauss-Seidel is used again until these new local errors are removed.
- Now that the original global errors have been solved, the solution is 'transferred' back to the original mesh (this process is called prolongation).

More detailed information can be found in the FLUENT manual [126].

4.1.5 Convergence

Convergence is determined by looking at the so-called residuals. If the values of the residuals decrease below the convergence criterion (10^{-3} is the standard FLUENT value, but this can be changed), the solution is considered to be converged. The definition of the scaled residual for the momentum equation in the x direction is:

$$R^u = \frac{\sum_{all\ cells} |\sum_{nb}(a_{nb}u_{nb}) + b - a_P u_P|}{\sum_{all\ cells} |a_P u_P|} \tag{4.9}$$

This definition can be easily extended to the scaled residuals for the y- and z-momentum equations. The unscaled residual for the continuity equation is defined in a different way:

$$R^c = \sum_{all\ cells} |rate\ of\ mass\ creation\ in\ cell| \tag{4.10}$$

This residual is then scaled with the largest absolute value of the continuity residual in the first five iterations:

$$R^c_{scaled} = \frac{R^c_{iteration\ N}}{R^c_{max\ of\ iterations\ 1-5}} \tag{4.11}$$

4.1.6 Under-relaxation

Because of the non-linearity of the equations that are solved, it is necessary to control the change of the variables. This is done by under-relaxation. In a simple form, the new value of a variable ϕ produced during each iteration depends upon the old value ϕ_{old}, the computed change $\Delta\phi$ and the under-relaxation factor α, and it is defined as follows:

$$\phi = \phi_{old} + \alpha\Delta\phi \tag{4.12}$$

4.2 Code development

The following sections provide an explanation of how the models described in Chapter 3 were implemented into FLUENT. All together over 20000 lines of C code were written to accomplish this.

4.2.1 LS implementation

In FLUENT, the LS method and the contribution of the surface tension force to the Navier-Stokes equation were introduced by implementing user defined functions (UDF). Two approaches were used. In the first, the source term (surface tension contribution) in Eq. (3.2) was computed directly at the cell-centers. In the second, the approach of Mencinger and Zun [3] was implemented where the body force is first computed at the faces of the control volume and is then interpolated at the cell centers (see Fig. (4.5) from [3]) using:

$$\begin{aligned} f^x_P V_P &= I_e\delta y_e \mathbf{f_e}\cdot\mathbf{\Delta_e} + I_w\delta y_w \mathbf{f_w}\cdot\mathbf{\Delta_w} - I_n\delta y_n \mathbf{f_n}\cdot\mathbf{\Delta_n} - I_s\delta y_s \mathbf{f_s}\cdot\mathbf{\Delta_s} \\ f^y_P V_P &= I_n\delta x_n \mathbf{f_n}\cdot\mathbf{\Delta_n} + I_s\delta x_s \mathbf{f_s}\cdot\mathbf{\Delta_s} - I_e\delta x_e \mathbf{f_e}\cdot\mathbf{\Delta_e} - I_w\delta x_w \mathbf{f_w}\cdot\mathbf{\Delta_w} \end{aligned} \tag{4.13}$$

where I_e is the interpolation coefficient for the east face, δy_e and δx_e are the components of the vector pointing between the FV-corner, $\mathbf{f_e}$ is the body force computed at the east face, and $\mathbf{\Delta_e}$ is the vector pointing between node P and E, see Fig. (4.5). Similar definitions were used for the other faces.

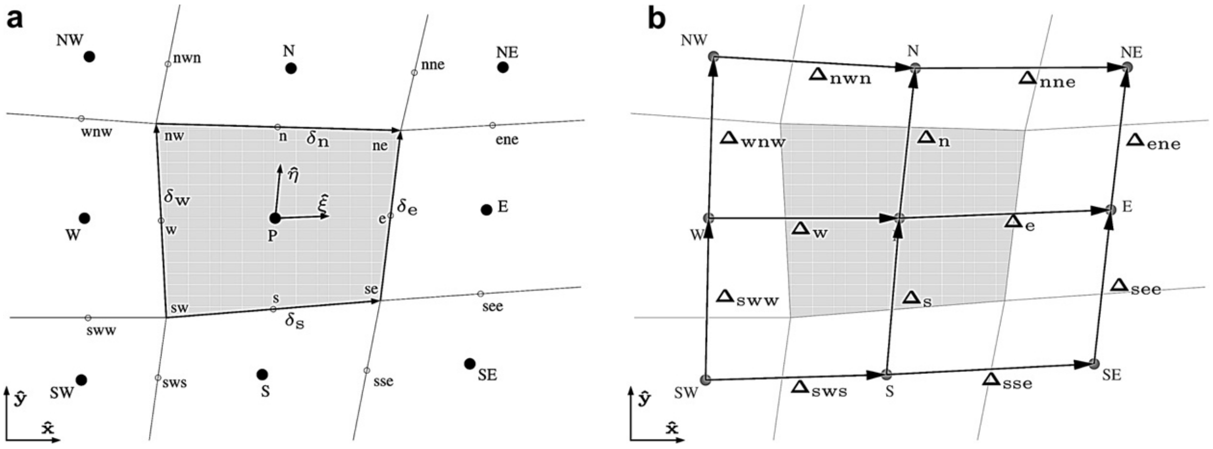


Figure 4.5: A typical non-orthogonal control volume from [3].

An equivalent formulation of Eq. (4.13) allows this formulation to be easily extended to 3D:

$$
\begin{aligned}
f_P^x V_P &= I_e A_{e,x} \mathbf{f_e} \cdot \mathbf{\Delta_e} + I_w A_{w,x} \mathbf{f_w} \cdot \mathbf{\Delta_w} - I_n A_{n,x} \mathbf{f_n} \cdot \mathbf{\Delta_n} - I_s A_{s,x} \mathbf{f_s} \cdot \mathbf{\Delta_s} \\
&\quad - I_t A_{t,x} \mathbf{f_t} \cdot \mathbf{\Delta_t} - I_b A_{b,x} \mathbf{f_b} \cdot \mathbf{\Delta_b} \\
f_P^y V_P &= I_n A_{n,y} \mathbf{f_n} \cdot \mathbf{\Delta_n} + I_s A_{s,y} \mathbf{f_s} \cdot \mathbf{\Delta_s} - I_e A_{e,y} \mathbf{e_n} \cdot \mathbf{\Delta_e} - I_w A_{w,y} \mathbf{f_w} \cdot \mathbf{\Delta_w} \\
&\quad - I_t A_{t,y} \mathbf{f_t} \cdot \mathbf{\Delta_t} - I_b A_{b,y} \mathbf{f_b} \cdot \mathbf{\Delta_b} \\
f_P^z V_P &= I_t A_{t,z} \mathbf{f_t} \cdot \mathbf{\Delta_t} + I_b A_{b,z} \mathbf{f_b} \cdot \mathbf{\Delta_b} - I_e A_{e,z} \mathbf{e_n} \cdot \mathbf{\Delta_e} - I_w A_{w,z} \mathbf{f_w} \cdot \mathbf{\Delta_w} \\
&\quad - I_n A_{n,z} \mathbf{f_n} \cdot \mathbf{\Delta_n} - I_s A_{s,z} \mathbf{f_s} \cdot \mathbf{\Delta_s}
\end{aligned}
\tag{4.14}
$$

where $A_{e,x}$, $A_{e,y}$ and $A_{e,z}$ are the x, y and z components of the east area face vector. Similar definitions were used for the other faces.

The LS method implemented in FLUENT by the UDF's is available for both serial and parallel solvers. To summarize, with the LS implementation, the following CLSVOF algorithm can be used in FLUENT:

- *enable and initialize a User Defined Scalar (UDS) transport equation. This is the level set function;*
- *enable 16 User Defined Memory (UDM) locations. In these locations the curvature and the normal to the interface plus other intermediate values were saved;*
- *based on the initial solution, compute the curvature and the normal to the interface using the LS function. A DEFINE_ADJUST function was used to compute these values. 4 UDMs were used to store the curvature and the x, y and z components of the normal to the interface;*
- *solve the Navier-Stokes equations to get the velocity and pressure fields. The Navier-Stokes*

equations have an additional source term – the surface tension force. These source terms were computed with a DEFINE_SOURCE function.

- *advect Eq. (3.1) and Eq. (3.13) with the velocity field;*
- *to ensure mass conservation for the LS function, couple LS with VOF. In other words, since the VOF and LS functions refer to the same interface, the new interface location should coincide using either the VOF or LS function. The following equation is solved:*

  $$H(\phi) = F_L$$

  or if we expand the definition of the Heaviside function:

  $$(\phi + \epsilon)/(2\epsilon) + sin(\pi\phi/\epsilon)/(2\pi) = F_L \tag{4.15}$$

  Eq. (4.15) is solved after each time step using a Newton iterative method. A DEFINE_EXECUTE_AT_END function was used;
- *to keep the interface thickness constant and to ensure that the LS function is a distance function to the interface, a re-initialization equation, see Eq. (3.8), is solved after each time step. A DEFINE_EXECUTE_AT_END function was used;*
- *go to the next time step.*

4.2.2 Dynamic contact angle implementation

The dynamic contact angle model was implemented into FLUENT using User Defined Functions. The values of the contact angle and other parameters are stored for the current and the previous time step, in order to handle the situation when the contact line shifts from one cell to another. Once the contact angle is computed and returned to the FLUENT solver, the surface normal at the live cell next to the wall is given by:

$$\mathbf{n} = n_w \cos\theta + n_t \sin\theta \tag{4.16}$$

where n_w and n_t are the unit vectors normal and tangential to the wall, respectively. To summarize, the following algorithm was used:

- *for the cell containing the interface, compute the plane (line) constant α;*
- *by knowing its position at the current and at previous time steps, a contact line velocity can be computed;*
- *with the contact line velocity known, the dynamic advancing and dynamic receding contact angles can be computed using Eq. (3.22) and Eq. (3.21), based on the input values of static advancing and static receding contact angles;*

- *the dynamic contact angle is then computed using Eq. (3.27) (or Eq. (3.28) for the 2D case) and if the value falls between receding and advancing angles, then this value will be returned to the FLUENT solver, otherwise the advancing or the receding angle is returned to the FLUENT solver;*
- *move to the next time step.*

4.2.3 Heat transfer model implementation

The heat transfer model used in this work was implemented into FLUENT using UDF's. The following algorithm was used:

- *at the beginning of the time step or at the beginning of each iteration, compute the gradient of the volume fraction based on the solution of the previous time step or iteration;*
- *compute the source terms using Eq. (3.31), Eq. (3.32) and Eq. (3.33) using the gradient of the volume fraction and the physical properties of the fluid;*
- *at the end of the time step set the vapor liquid interface temperature to the saturation value;*
- *move to the next time step.*

4.3 Novel aspects of the new code

With the increasing computer capabilities, a new question have raised: which way is better to follow: to develop a new code from scratch, or to use a commercial code as a starting point and eventually implement your own models? This question is not easy to answer since each way has advantages and disadvantages. For instance a commercial code is well documented, provides a very good pre and post-processing tools and a very robust and fast solver, but the access to the source code is somehow limited meaning that not all the models could be implemented. On the other hand, with an in-house code you can implement any model, but the solver is not usually as fast and robust as a commercial one, the parallel implementation is not easy to do and, most important, the time to develop the code is not so short. Taking into consideration all these aspects, it was decided to use a commercial code together with UDFs to achieve the goals of this work. More than 20000 lines of C code were written and hooked to FLUENT to provide it with more capabilities. To summarize, the following novel aspects were added to FLUENT:

- a 3D, parallel LS method coupled with the existing VOF method inside FLUENT ⇒ CLSVOF method;
- a 3D, parallel dynamic contact angle model to simulate flows with wetted boundaries;

- a 3D, parallel heat transfer model to model diabatic flows with phase change.

Chapter 5

Numerical results

This chapter presents several numerical verification and validations of the developed CLSVOF method, dynamic contact angle model and heat transfer model that were implemented into FLUENT.

5.1 Numerical validations of the CLSVOF method

This section presents several numerical tests which were used to assess the accuracy and the stability of the CLSVOF method developed. For all these simulations, ϵ in Eq. (3.6) was set to $1.5\Delta x$, where Δx is the mesh size in the x direction. For most of the simulations the CLSVOF method is compared with the VOF method. However, the results obtained with the LS method for the same cases are not shown because the stand-alone LS method loses mass. To demonstrate this, we simply advect the LS and VOF functions with the velocity field given by the buoyancy effect, using the same setup as in Section 5.1.2. The re-initialization equation Eq. (3.8) was also solved. The two contours obtained with the LS and VOF functions are showed in Fig. (5.1). It is very clear that LS lost a very significant amount of mass. This known drawback of the LS method was also stressed by Sussman *et al.* [23] and by Tornberg and Engquist [128].

Benchmark tests

5.1.1 Verification cases

In order to verify that Eq. (3.8) and Eq. (4.15) were implemented correctly, we performed a simulation that consisted of a bubble rising in a viscous fluid due to gravity. In the present study, calculations were performed on a 2D computational domain of 0.08x0.1 m. The bubble was placed at (0.04, 0.02) m and had a radius of 0.005 m. The no slip wall boundary condition was used for all boundaries. The physical properties of the fluids used are presented in Table (5.1). Fig. (5.2) depicts the level set function inside the domain after 0.2 s when solving the re-

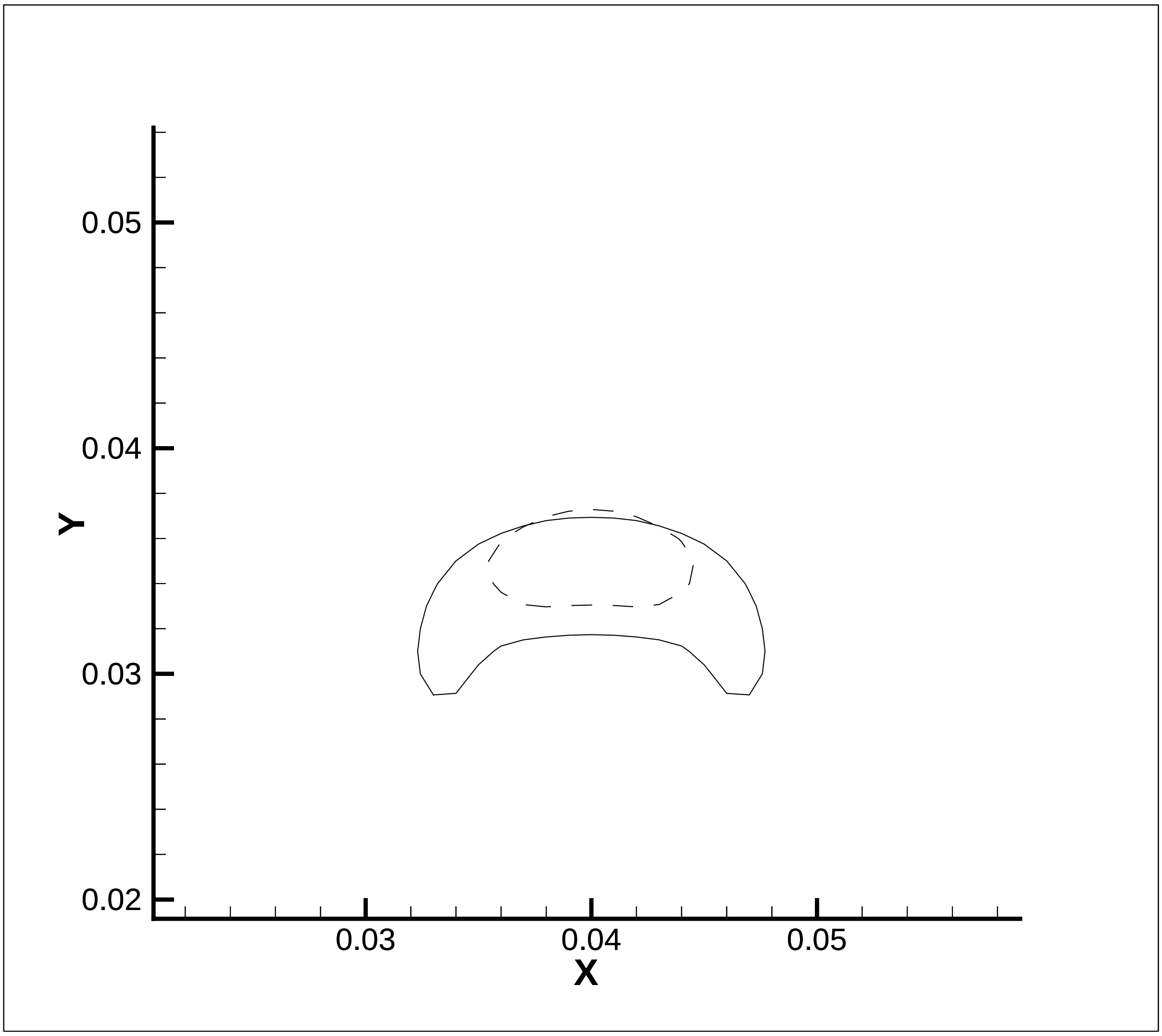


Figure 5.1: LS-VOF comparison. Dashed contour–LS function. Continuous contour–VOF function.

initialization equation after each time step. In Fig. (5.3) the same level set function inside the domain is plotted after 0.2s. It is clear that in the first case (Fig. (5.2)) the re-initialization equation is solved correctly and the level set function is a signed distance to the interface. With Fig. (5.3) we proved that the level set function ceased to be a signed distance to the interface when advecting the level set equation (Eq. (3.1)). Fig. (5.4) shows the bubble contour with both LS and VOF functions when solving the coupling equation (Eq. (4.15)) after 0.2s. In Fig. (5.5) the same contours are shown without the coupling between LS and VOF. Again it is clear that: (i) LS method lost a significant fraction of the mass and (ii) by coupling LS with VOF one can conserve mass in case of a LS method.

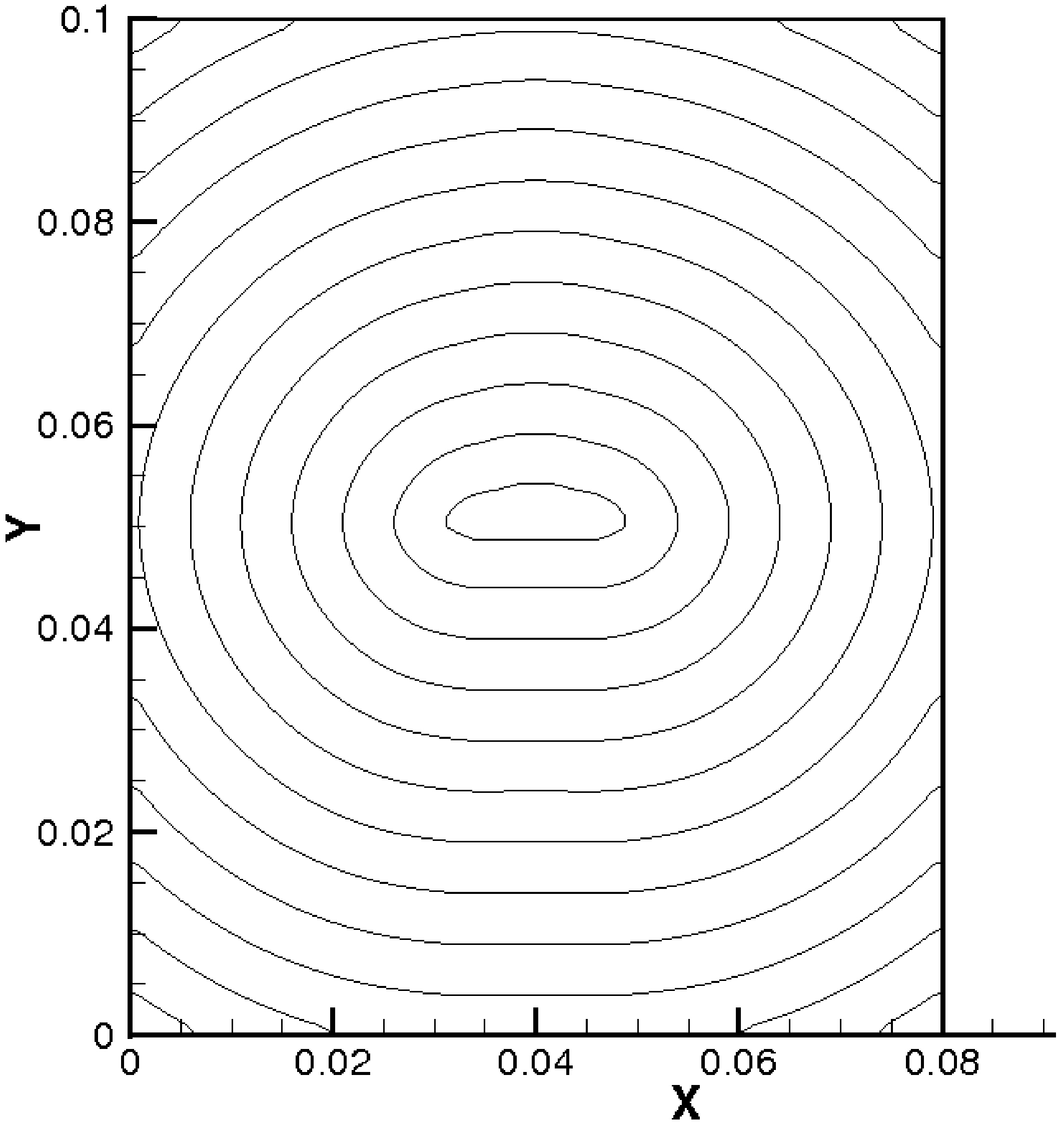


Figure 5.2: The level set function after solving the re-initialization equation.

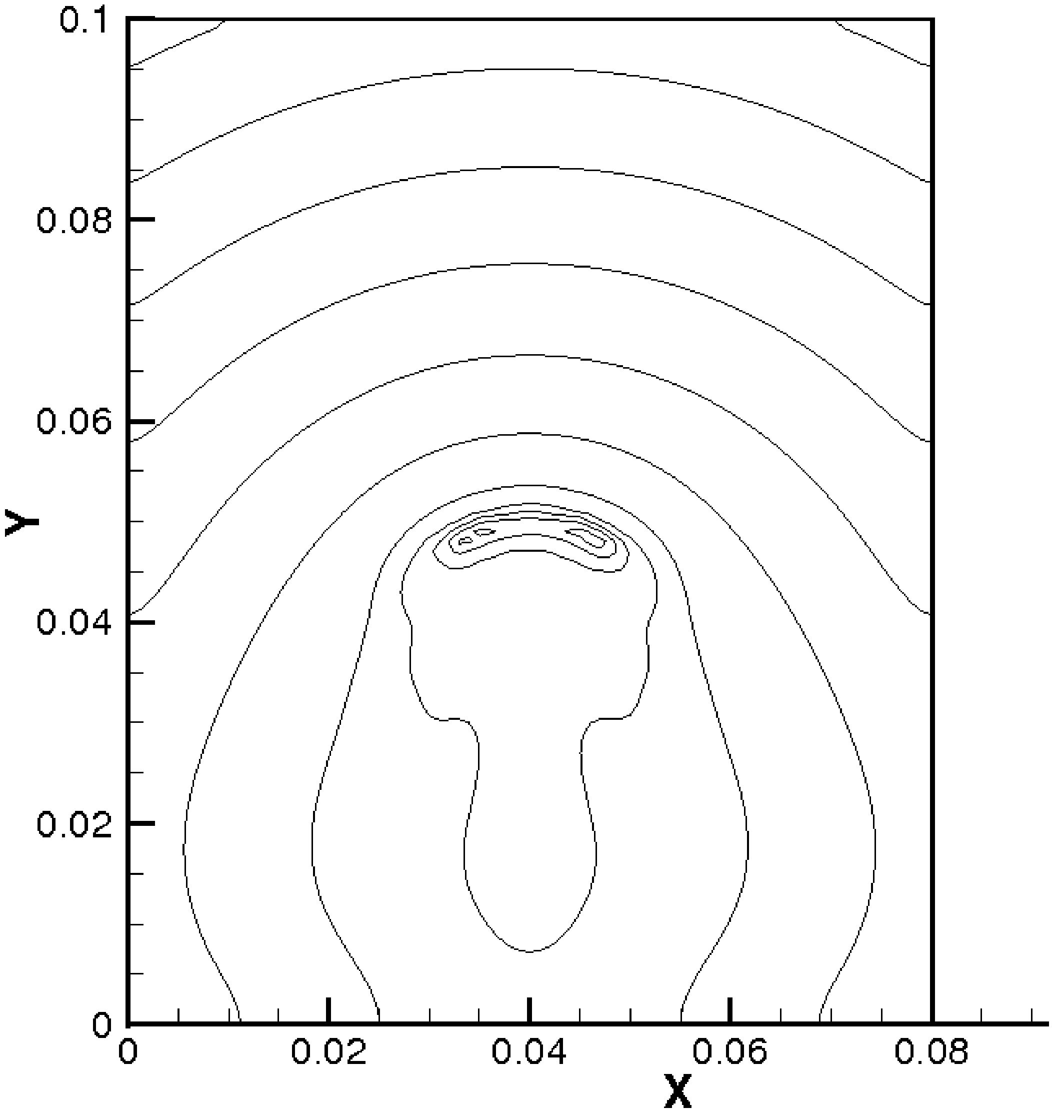


Figure 5.3: The level set function without solving the re-initialization equation.

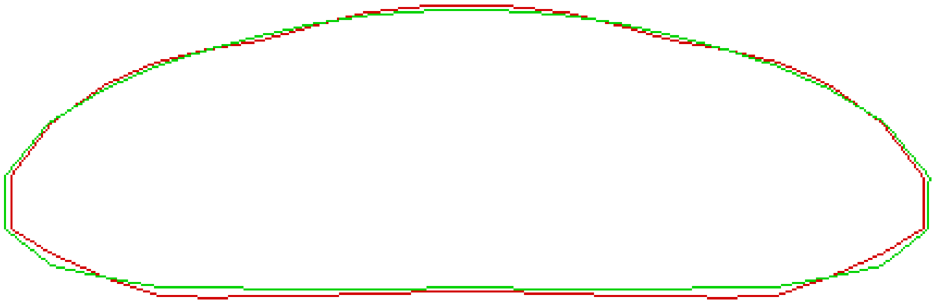


Figure 5.4: Level set contour (red) and volume-of-fluid contour (green) after solving the coupling equation between LS and VOF.

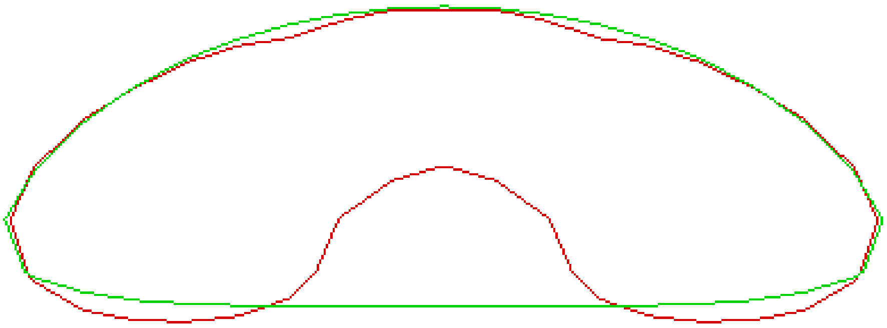


Figure 5.5: Level set contour (red) and volume-of-fluid contour (green) without coupling between LS and VOF (with large loss of mass).

5.1.2 Static bubble

A stationary spherical bubble without gravity is often used in the literature [21,31,45,48,76] to test the surface tension implementation for spurious currents. Theoretically, in the absence of external forces and initial velocities, the velocity field should remain zero throughout. However, if surface tension is included and artificial motions are generated, these can be ascribed directly and exclusively to the surface tension algorithm. The velocity magnitude reveals the efficacy of the treatment, while the pressure increase inside the bubble, compared to the analytical value, reflects the accuracy of the algorithm.

In the present study, calculations were performed on a 2D computational domain of 0.08x0.1 m. The bubble is placed at (0.04,0.02) m and has a radius of 0.005 m. The no slip wall boundary condition was used for all boundaries. The physical properties of the fluids used are presented in Table (5.1).

Table 5.1: Physical properties of the fluids.

Properties	Density kg/m^3	Viscosity Ns/m^2	Surface Tension N/m
liquid	1000	0.1	0.1
gas	10	0.001	

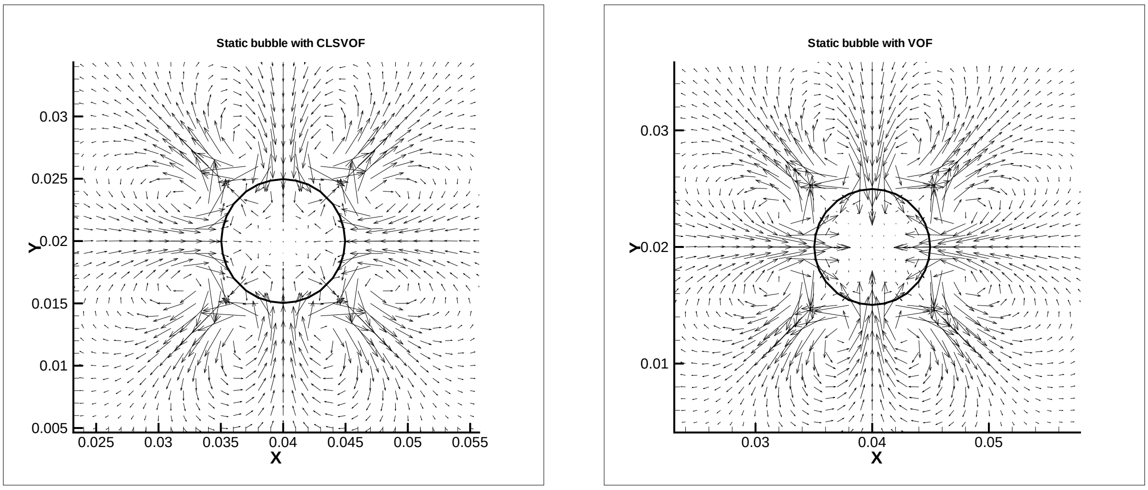
We compared the VOF method available in FLUENT with the CLSVOF technique developed here. The parasitic current fields around the bubble after 1s are presented in Fig. (5.6) for both CLSVOF and VOF using three different grid resolutions: 80x100, 160x200 and 320x400. As expected, and confirmed in Table (5.2), the CLSVOF method performs much better than the VOF method.

Table 5.2: Spurious currents for static bubble after 1.0s.

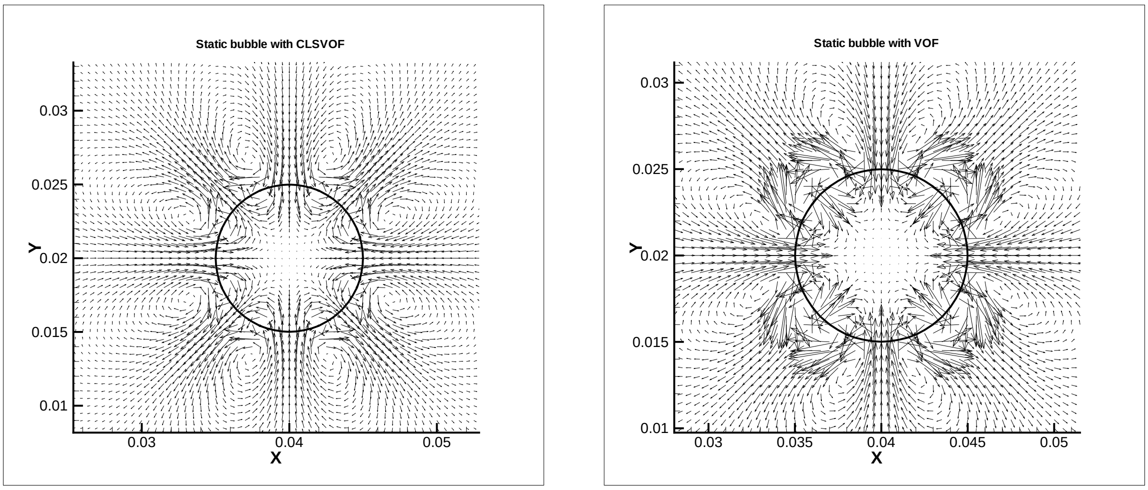
Method	$u_{max}(m/s)$	$u_{mean}(m/s)$	Kinetic energy	E_1^i
CLSVOF				
80x100	2.616e-3	8.126e-5	38.86e-8	N.A.
160x200	1.318e-3	3.676e-5	8.281e-8	10.8e-5
320x400	0.756e-3	1.856e-5	2.244e-8	2.6e-5
VOF				
80x100	4.097e-3	15.78e-5	150.9e-8	N.A.
160x200	2.486e-3	4.203e-5	14.24e-8	180.0e-5
320x400	2.453e-3	2.892e-5	9.373e-8	245.0e-5

The strength of the spurious currents is reduced on average by approximately 51% by the introduction of the level set method. Moreover, if we consider the two refined grids, 160x200 and 320x400, with CLSVOF we observe a reduction of the strength of the spurious currents by approximately 42%, whereas with VOF the reduction is only 1.3%.

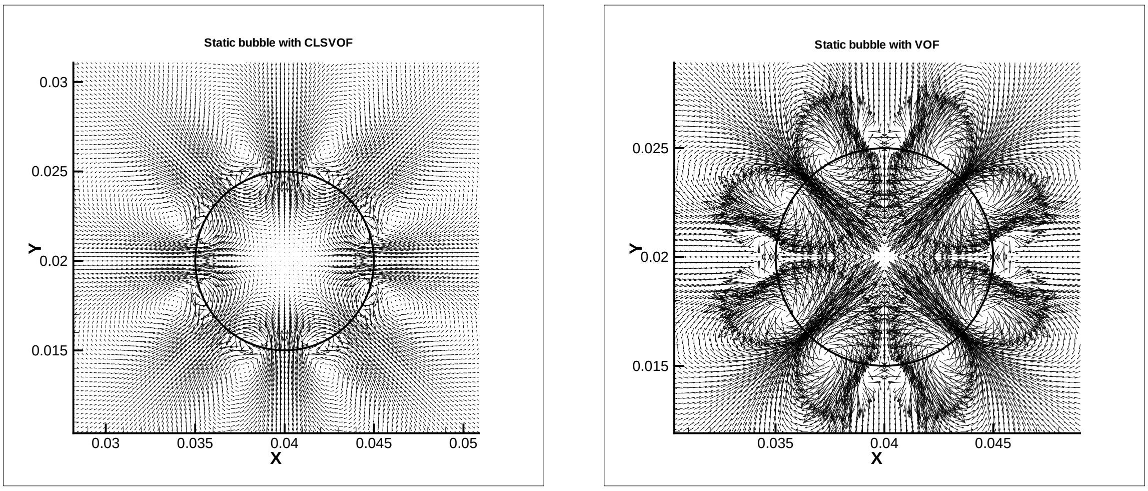
In Table (5.2), besides the maximum and mean velocities, the total kinetic energy was also



(a) CLSVOF 80x100 (b) VOF 80x100



(c) CLSVOF 160x200 (d) VOF 160x200



(e) CLSVOF 320x400 (f) VOF 320x400

Figure 5.6: Static bubble parasitic currents for 2D CLSVOF and VOF.

computed, defined as $W_k = \sum_p \rho_P |u_P|^2 V_P$.

Table (5.2) also presents the error E_1^i between successive mesh resolutions. This is computed using the L_1^i norm, which is well suited [26,32] for interfacial flow problems:

$$E_1^i = \int_\Omega |H(\phi_r) - H(\phi_c)| d\Omega \tag{5.1}$$

where $H(\phi)$ is the discontinuous Heaviside function, and ϕ_r and ϕ_c are solutions from, respectively, refined and coarse grids. For the numerical integration of Eq. (5.1) a domain of 1000x1000 mesh points was assumed and the solution was interpolated from the actual domain to the computational one. Again, it is very clear that CLSVOF performs better than VOF in this case.

5.1.3 Droplet deformation due to a vortex velocity field

In this section the results of the simulation of the time reversed single vortex flow are presented for the same conditions as in [69].As opposite as in [69], the simulations took into account the surface tension effects. A circle of fluid of radius 0.15 m, initially centered at (0.5,0.75) m in a unit square computational domain, is deformed by the vortex velocity field defined by the following time periodic stream function:

$$\Psi(x, y) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\pi t/T)$$

where T is the period and

$$u = -\frac{\partial \Psi}{\partial y} \qquad v = \frac{\partial \Psi}{\partial x}$$

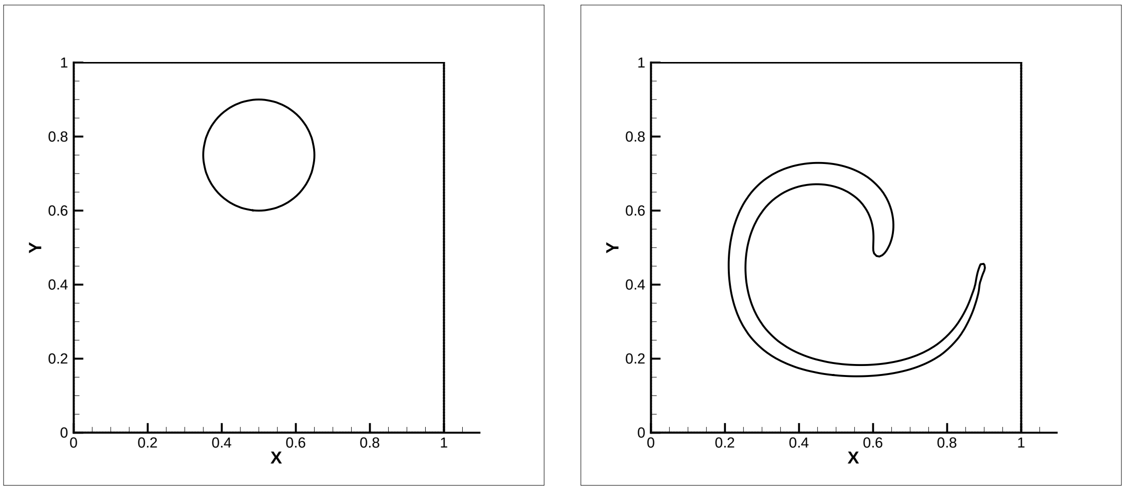
Air-water were used as fluids (see Table (5.8) for their physical properties). A no slip wall boundary condition was used for all boundaries. The circle of fluid undergoes the largest deformation at $t = T/2$ and returns to its initial state at $t = T$. This type of flow can be used to assess the capacity of the re-initialization method to resolve the thin fluid filaments formed as a result of interface stretching. For all the simulations presented below, CLSVOF was used with the re-initialization equation solved for 10 time steps after each step, with a period T of 6s. Fig. (5.7) presents the interface deformation computed on a grid of 128x128 mesh points. As can be seen, the droplet undergoes severe deformation at $T/2$ whilst its final position is very close to the initial one, with no formation of small bubbles. The same case is presented in Fig. (5.8) on a grid of 256x256 mesh points and in Fig. (5.9) on a grid of 512x512 mesh points. These show that interface deformations decrease in amplitude with increasing grid resolution.

In Table (5.3) the droplet coordinates after one period are shown and also indicates the error E_1^i between successive mesh resolutions as computed in the Subsection 5.1.2. Even for the coarsest grid, the droplet coordinates are very close to the initial ones, while for the finest grid the initial and final droplet coordinates differ by a few microns. The presently proposed CLSVOF method

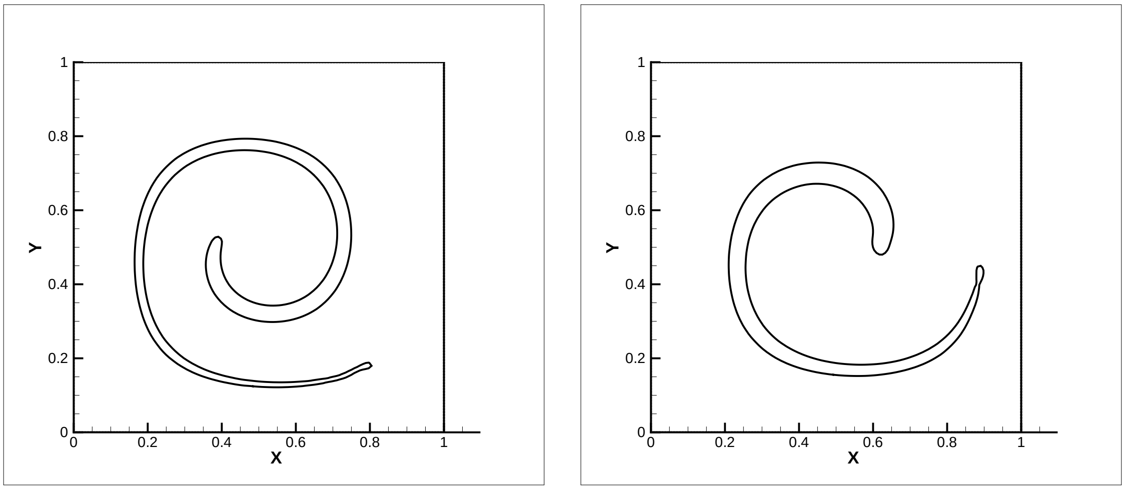
Table 5.3: Droplet coordinates after one period of shear flow with CLSVOF.

Grid	$x_{initial}$ (m)	$y_{initial}$ (m)	x_{final} (m)	y_{final} (m)	E_1^i
128x128	0.5	0.75	0.499905	0.749952	N.A.
256x256	0.5	0.75	0.499988	0.749989	1.39e-4
512x512	0.5	0.75	0.500003	0.749994	1.14e-4

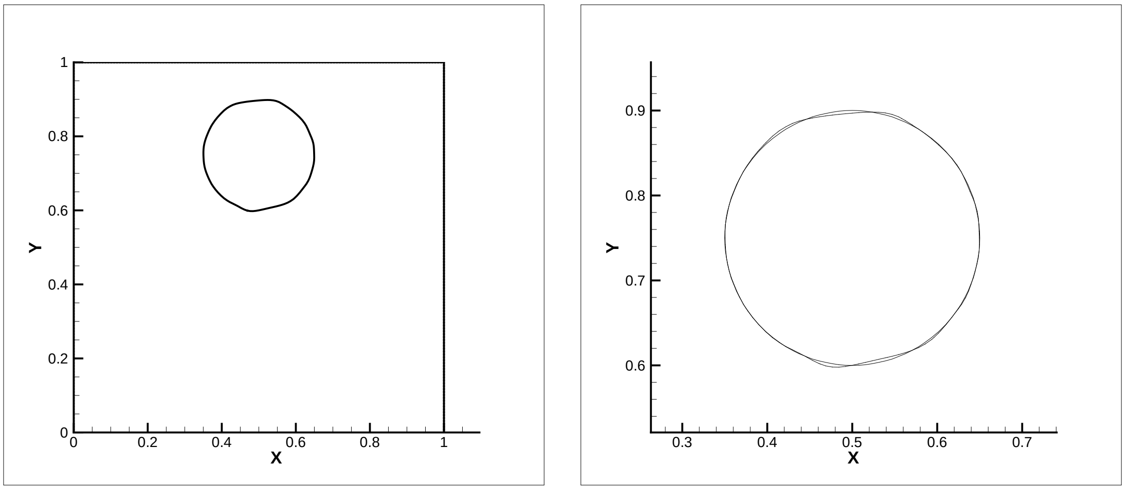
is thus also proven to work well under these circumstances.



(a) t=0s (b) t=1.5s

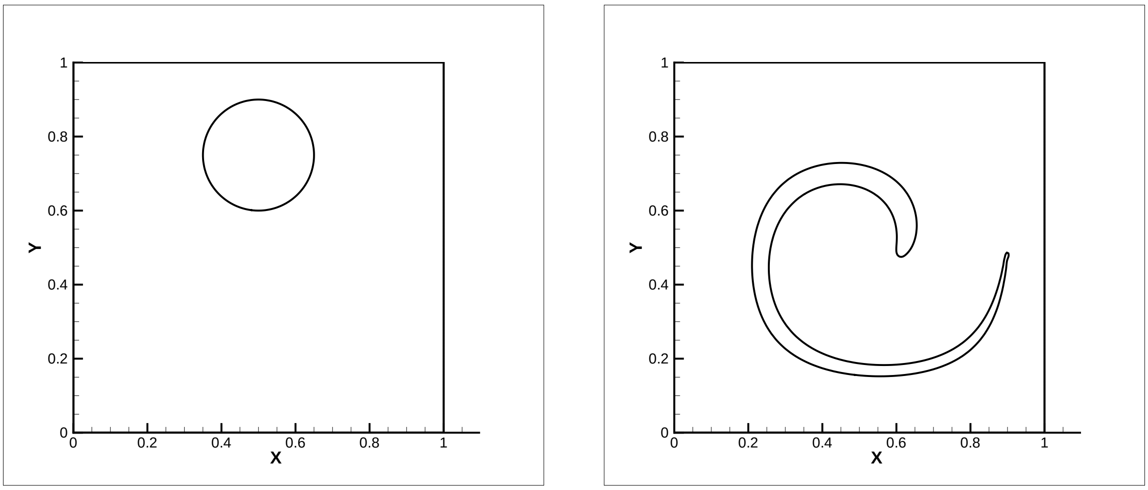


(c) t=3s (d) t=4.5s

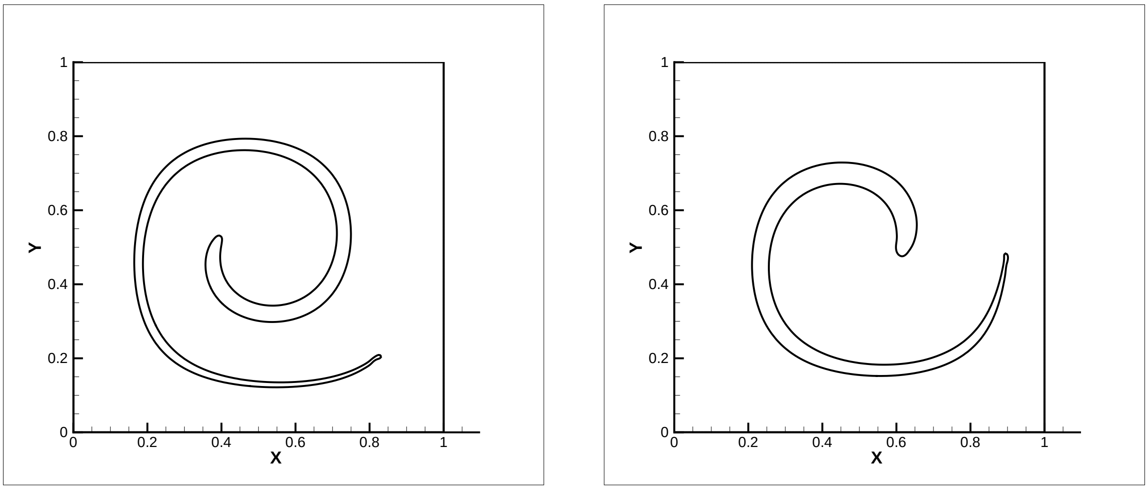


(e) t=6s (f) initial and final droplet interface

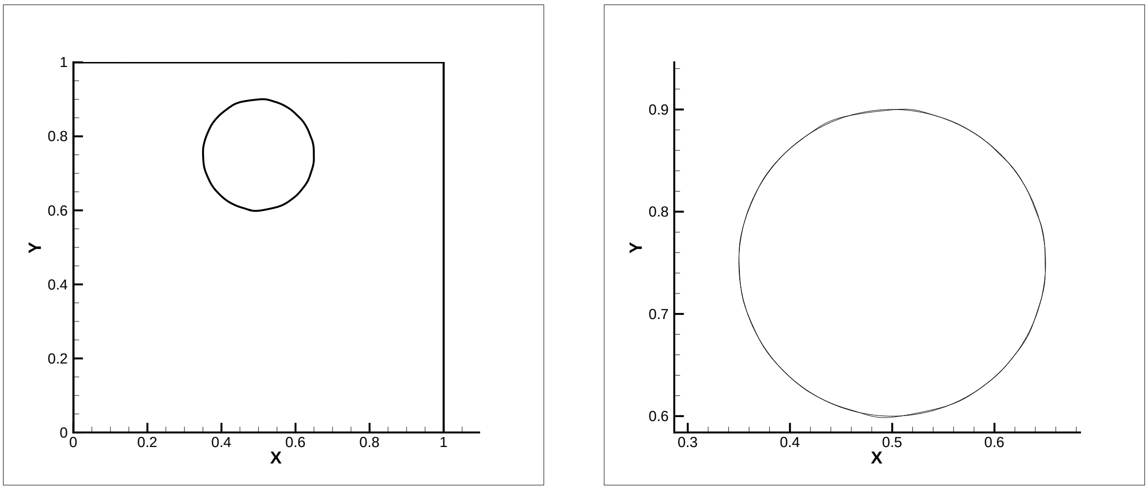
Figure 5.7: Interface location for a 2D droplet deformed by a vortex, with T=6s and a grid of 128x128.



(a) t=0s (b) t=1.5s

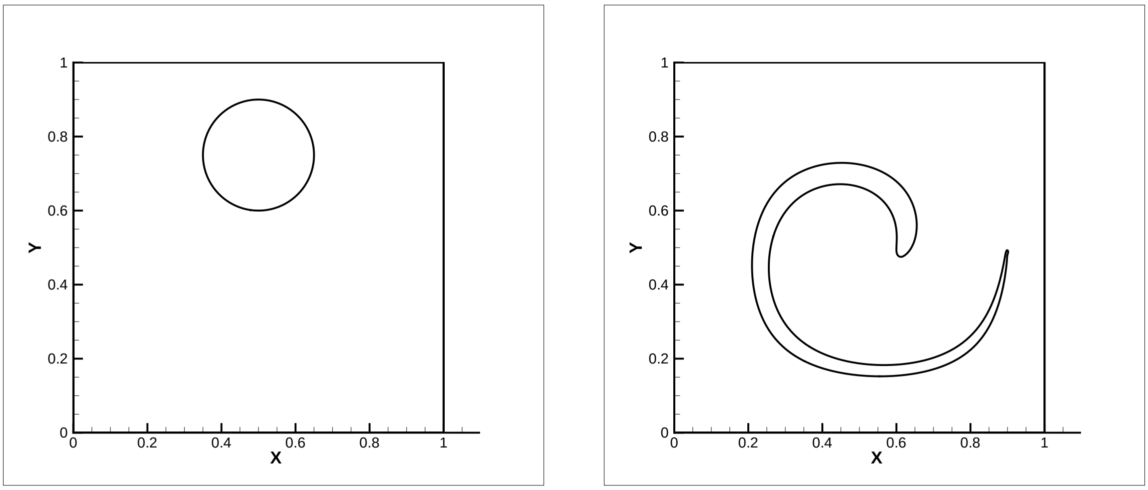


(c) t=3s (d) t=4.5s



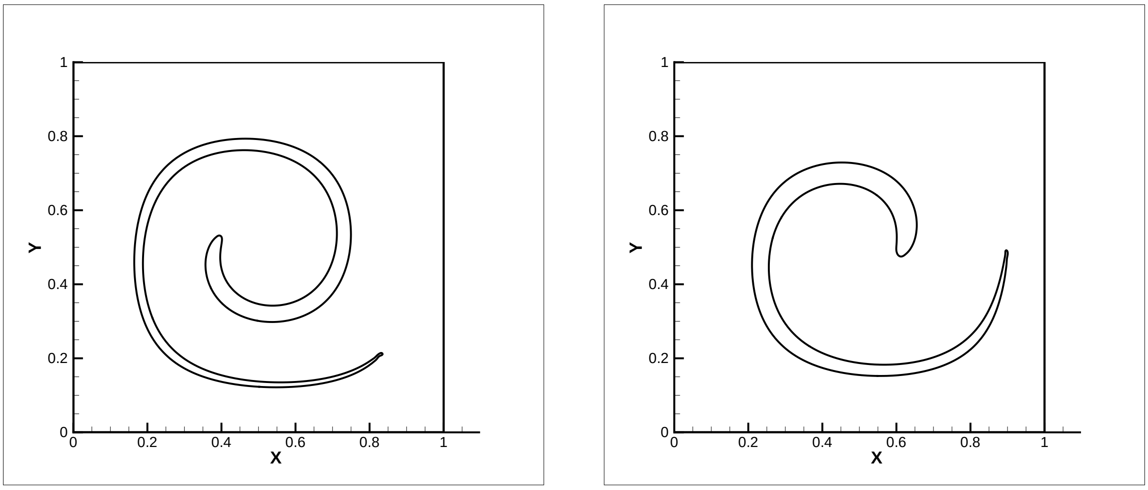
(e) t=6s (f) initial and final droplet interface

Figure 5.8: Interface location for a 2D droplet deformed by a vortex with T=6s and a grid of 256x256.



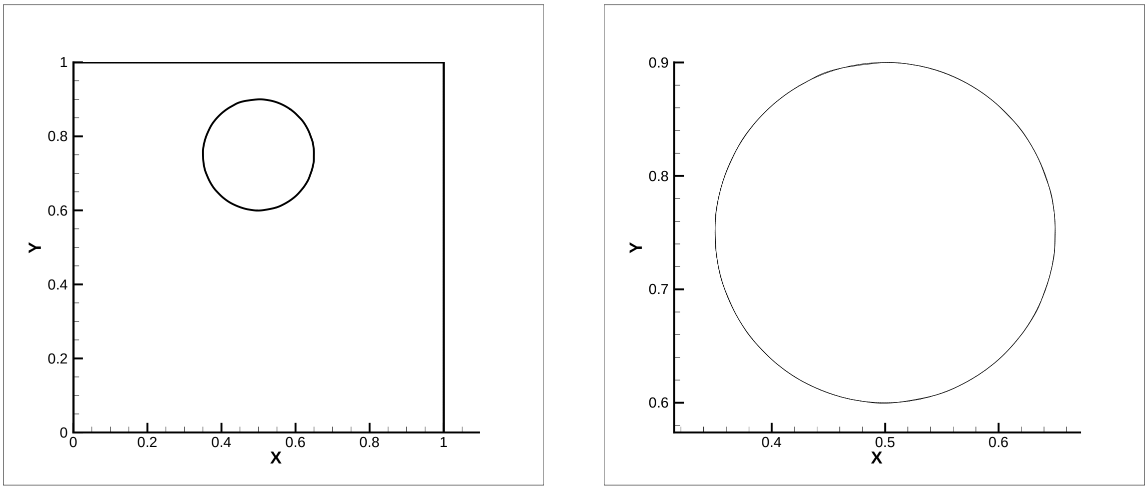
(a) t=0s

(b) t=1.5s



(c) t=3s

(d) t=4.5s



(e) t=6s

(f) initial and final droplet interface

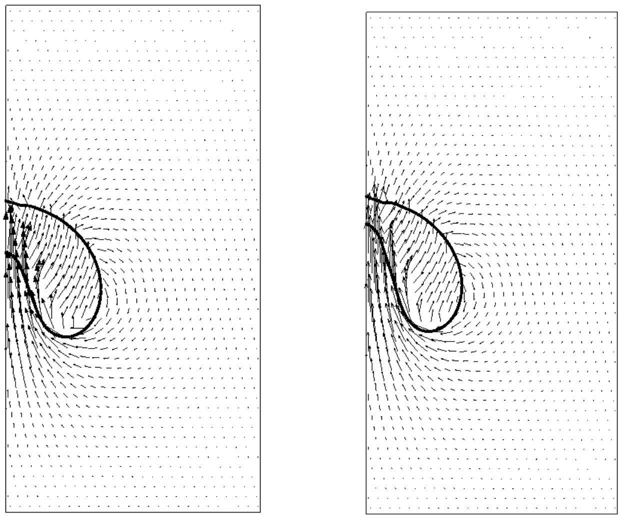
Figure 5.9: Interface location for a 2D droplet deformed by a vortex with T=6s and a grid of 512x512.

5.1.4 Inviscid gas bubble

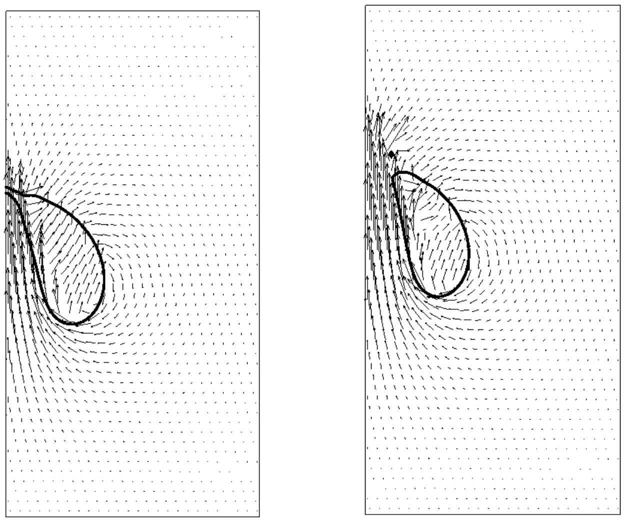
In this section we compared the results obtained with our CLSVOF method with those obtained with the CLSVOF method of Sussman [4]. A 2D-axisymmetric air bubble rise in a inviscid fluid was considered. A unit spherical bubble centered at (0,2) in a domain of 3x6m was set as the initial condition. For this case a surface tension of 0.005 N/m and a gravity vector $\mathbf{g} = (0, -1.0)\ m/s^2$ were used. The physical properties of the liquid are similar to those of the water, with the viscosity set equal to zero. Three grid sizes were considered 32x64, 64x128 and 128x256. Fig Fig. (5.10) depicts the interface position for different time slots. With our CLSVOF method we obtained similar shapes as Sussman [4], the only difference is that in our simulation the bubble breaks up a few time steps later than in Sussman's simulation. For convenience, Fig. (5.11) shows the interface position obtained by Sussman [4]. Nevertheless, we can conclude that our CLSVOF method produced similar results as the CLSVOF method of Sussman [4] for this case.

5.1.5 Rayleigh-Taylor instability

In this section we present the results of a Rayleigh-Taylor instability test, where a heavy fluid of density $\rho_1 = 1.225\text{kg/m}^3$ lies above a light fluid of density $\rho_2 = 0.1694\text{kg/m}^3$ in a rectangular domain 1m wide by 4m high. A similar viscosity of $3.13\text{x}10^{-3}\text{kg/ms}$ was used for both fluid considered immiscible. This test was done by many peoples, including López *et al.* [91]. The shape of the interface is initially a cosine function, $y = -0.2cos(\pi x/\lambda)$ with λ being the computational domain width. Three grid size were used, namely 64x256, 128x512 and 256x1024. Fig. (5.12) shows the interface position at different times. In Fig. (5.13) and Fig. (5.14) similar snapshots are presented for finer grids. It is clear that the solution is mesh independent, and even for the coarsest grid the CLSVOF algorithm is able to maintain thin filaments.

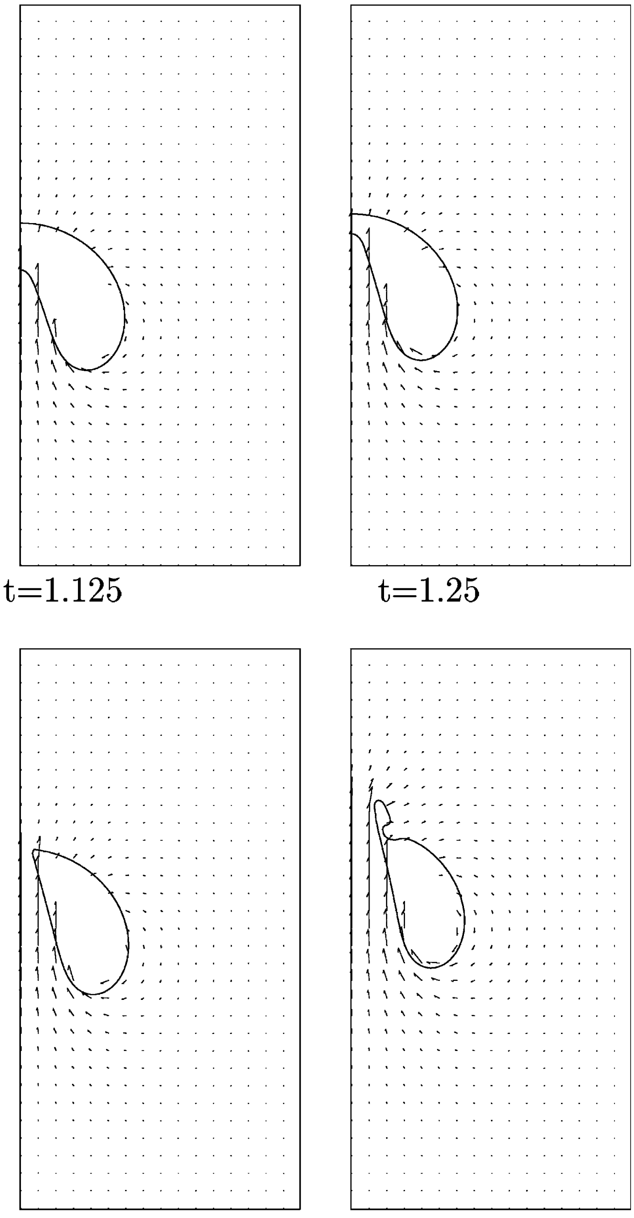


(a) t=1.125s (b) t=1.25s



(c) t=1.375s (d) t=1.75s

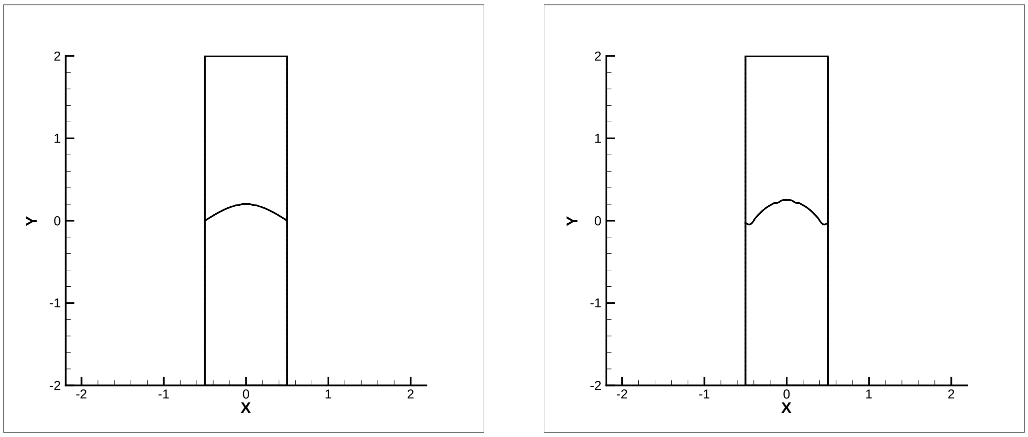
Figure 5.10: Interface position for an inviscid, axisymmetric gas bubble rising in liquid; 128x256 grid.



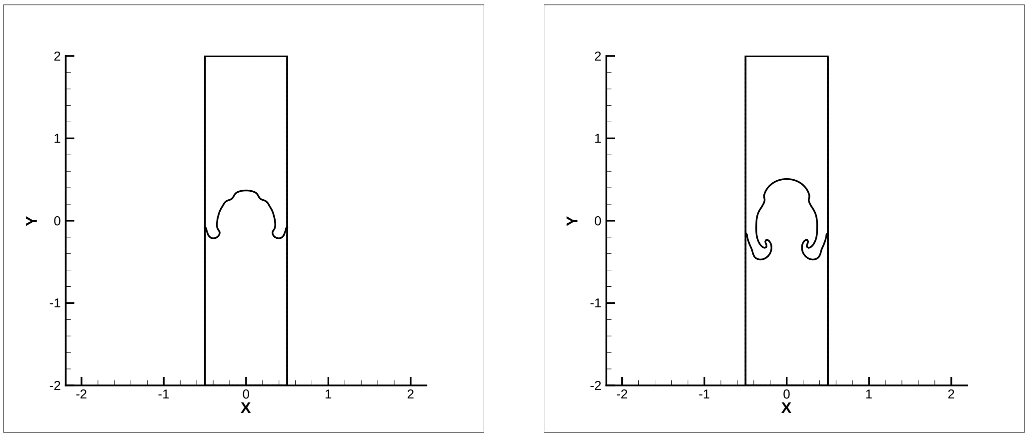
t=1.125 t=1.25

t=1.375 t=1.75

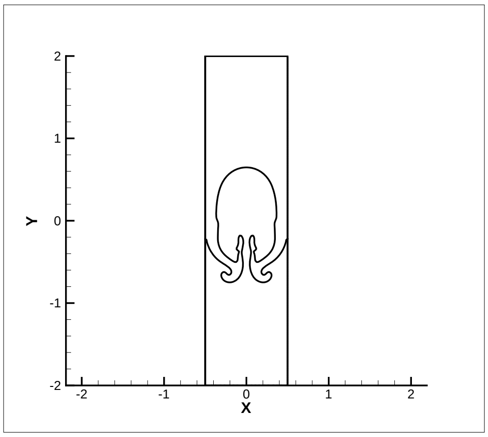
Figure 5.11: Interface position for an inviscid, axisymmetric gas bubble rising in liquid from Sussman [4].



(a) t=0.0s (b) t=0.5s

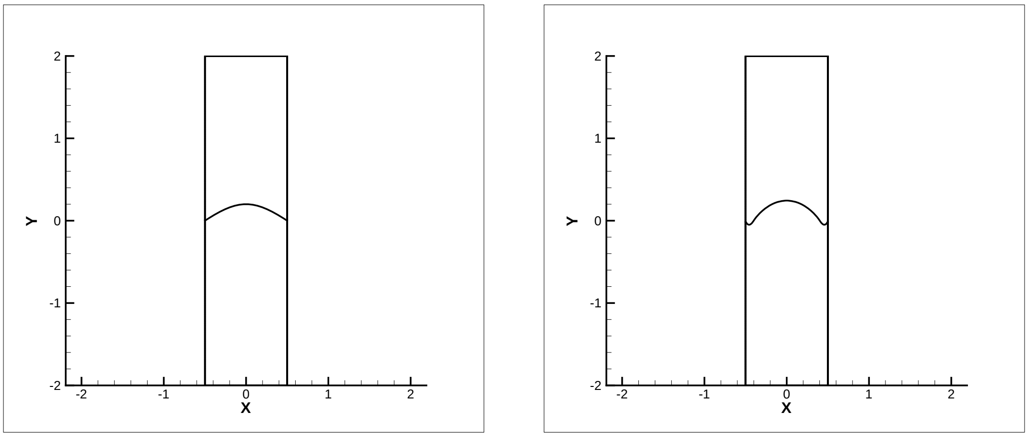


(c) t=1.0s (d) t=1.5s

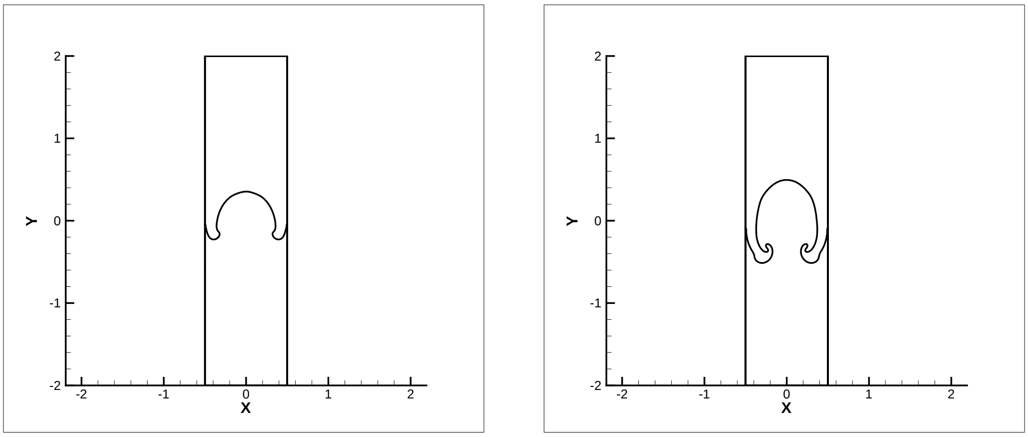


(e) t=2.0s

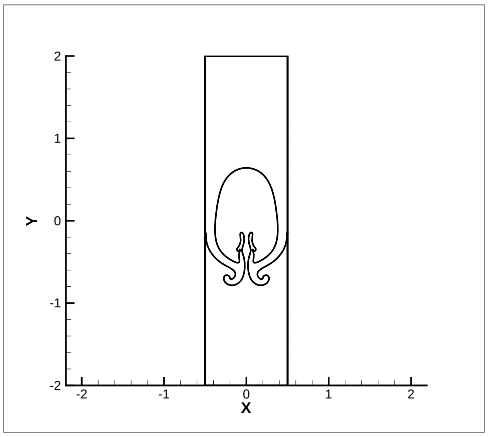
Figure 5.12: Interface evolution for Rayleigh-Taylor instability at different times on a grid of 64x256 mesh points.



(a) t=0.0s (b) t=0.5s

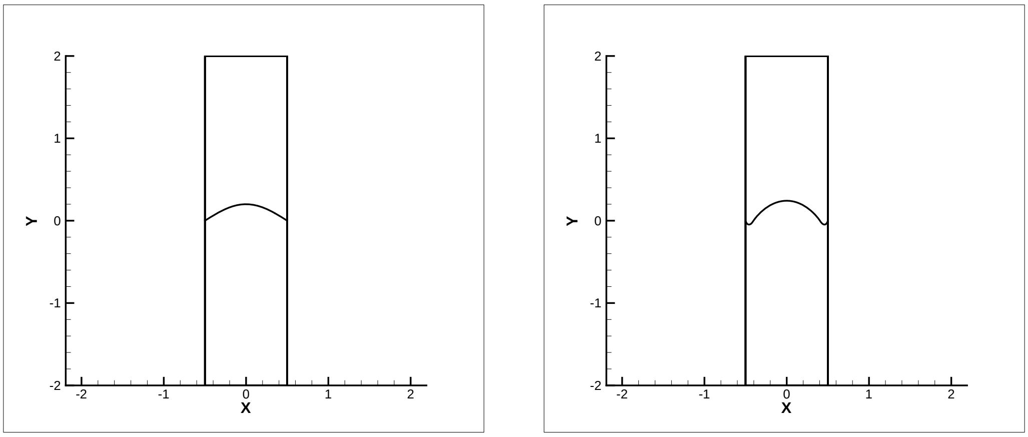


(c) t=1.0s (d) t=1.5s

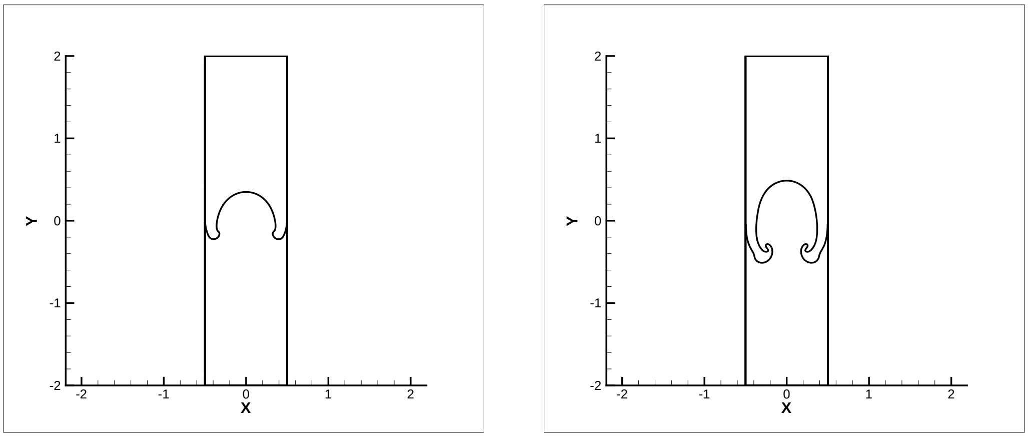


(e) t=2.0s

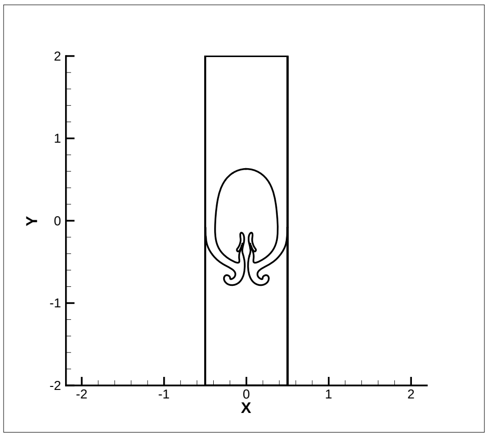
Figure 5.13: Interface evolution for Rayleigh-Taylor instability at different times on a grid of 128x512 mesh points.



(a) t=0.0s (b) t=0.5s



(c) t=1.0s (d) t=1.5s



(e) t=2.0s

Figure 5.14: Interface evolution for Rayleigh-Taylor instability at different times on a grid of 256x1024 mesh points.

Comparison with experimental data

5.1.6 Bubble rising in a stagnant liquid

The modeling of a rising bubble in an initially stagnant viscous fluid is a popular benchmark test for interface tracking and capturing methods [22, 24, 36]. Here, in the first part, simulation results for an air bubble rising in a water-sugar solution will be compared with the experimental data from Bhaga and Weber [9]. In the second part, results for a series of air bubbles rising in water will be presented and compared with experimental data of Zun and Groselj [5].

Air water-sugar simulations.

In the experiments of Bhaga and Weber [9], the viscosity of the liquid was varied by an order of magnitude by changing the sugar concentration. Therefore, in the present simulations three aqueous solutions of different viscosities were considered: $\mu_L = 2.73, 1.28$ and 0.54 kg/ms. The surface tension was set to that of water $\sigma = 0.078$ N/m while the air viscosity was $\mu_G = 0.0000178$ kg/ms. Although their fluid density varied by 4%, our simulations assumed $\rho_L = 1350$ kg/m^3.

Initially, at t=0, the bubble is approximated as a sphere of diameter $d_e = (6V/\pi)^{1/3}$. In the experiment, the volume V of the generated bubbles was 9.3 cm^3, so that $d_e = 2.61$ cm. A 3D simulation was performed on a parallelipipedic domain of $5d_e$x$10d_e$x$5d_e$ with a grid of 50x100x50. The no slip wall boundary condition was used for all boundaries. For comparison, the problem was simulated using our CLSVOF method and the standard VOF method. Grace [90] has analyzed a large body of experimental data on shapes and rise velocities of bubbles in quiescent viscous liquids and has shown that these data can be condensed into one diagram, provided that an appropriate set of dimensionless numbers is used. The dimensionless numbers *Mo* (Morton), *Eo* (Eötvös) and *Re* (Reynolds) are given by:

$$Mo = \frac{g\mu_L^4}{\rho_L\sigma^3}, \qquad Eo = \frac{gd_e^2\rho_L}{\sigma}, \qquad Re = \frac{\rho_L v_\infty d_b}{\mu_L}$$

with v_∞ is the terminal bubble rise velocity and d_b is the bubble equivalent diameter. Th *Eo* number represents the ratio between the buoyancy force and the surface tension force, the *Re* number represents the ratio between the inertial forces and the viscous forces and the *Mo* number is a dimensionless group which characterize the fluid properties.

In Table (5.4) the results for the bubble rise velocity (via Reynolds number) for both CLSVOF and VOF methods are presented. For the first two cases with high liquid viscosity, both CLSVOF and VOF perform in the nearly the same manner. However for the last case, where the surface tension forces becomes more important, CLSVOF performs slightly better than VOF when compared to the experimental results of Bhaga and Weber [9].

Table 5.4: CLSVOF, VOF and experimental data for an air bubble rising in a water-sugar solution.

	Mo	$E\ddot{o}$	Re_{num}	Re_{exp}	$\epsilon^a[\%]$
CLSVOF	850.5	114.77	2.061	2.09104	1.43
VOF	850.5	114.77	2.061	2.09104	1.43
CLSVOF	41.1	114.77	6.3043	7.3583	14.32
VOF	41.1	114.77	6.3043	7.3583	14.32
CLSVOF	1.31	114.77	17.2575	20.727	16.74
VOF	1.31	114.77	16.7375	20.727	19.25

---

[a] $\epsilon = \frac{Re_{exp} - Re_{num}}{Re_{exp}} \cdot 100$

Fig. (5.15) presents the bubble rise velocity as a function of time using CLSVOF and VOF. Similar results were found for both schemes. For the first two bubbles, with Morton numbers of 850 and 41.1 respectively, a steady state in terms of shape and rise velocity is reached quite rapidly. However, for the third bubble, although a steady state has been reached in terms of bubble shape, the bubble rise velocity is only a quasi-steady state. In Fig. (5.16) the interface location is presented for t=0s, 0.2s, 0.4s, 0.6s, 0.8s, 1.0s and 1.2s for a Morton number Mo=850 for both the CLSVOF and VOF methods. Similar plots are shown for Morton numbers Mo=41.1 and Mo=1.31 in Fig. (5.17) and Fig. (5.18), respectively. In these figures the bubble contour is presented in the XY plane for $Z = 0.5z_{max}$, where Z is the Cartesian coordinate. As an influence of various liquid viscosities, the bubble reached different final configurations from a dimpled ellipsoidal-cap to a skirted cap, similar to those observed by Grace [90].

A 16% relative error for the prediction of the bubble rise velocity when using the implemented CLSVOF method seems high, but we need also to consider the different aspects of the experiments and simulations. For instance, in the experiments the bubble velocity is measured a long time after it is formed and the bubble has reached the steady state in terms of shape and velocity. On the other hand, in the numerical simulations we are often limited, especially in 3D cases, to use a quite coarse mesh. In this case, we assigned 10 mesh cells along the bubble diameter with a total number of 256000 mesh cells. Since we initialize the bubble as a sphere, we can also measure the initial error in curvature calculation when using CLSVOF. We obtained an average error of 14.72%. Unfortunately we cannot evaluate directly the error in curvature calculation when using VOF (FLUENT doesn't provide this information), but, using the results from the static bubble simulations, we can say that the error in curvature calculation is higher when using VOF than when using CLSVOF. A mesh refinement will certainly improve the results. According to Bhaga and Weber [9] until $Re = 110$ the wake structure behind the bubble is closed and consequently symmetric, so we are safe to refine the mesh using a 2D-axisymmetric domain. Nevertheless, the implemented CLSVOF method slightly improved the obtained 3D results when compared with the built-in VOF method. Another reason for the error obtained in the prediction of the bubble rise velocity could be the influence of the walls. Again in 3D cases we are quite limited in using a tight domain, for this case the ratio between the bubble diameter and domain width is 0.2. According to *Clift et al.* [129] for a ratio above 0.125 we have a influence in the bubble rise velocity by lowering its value.

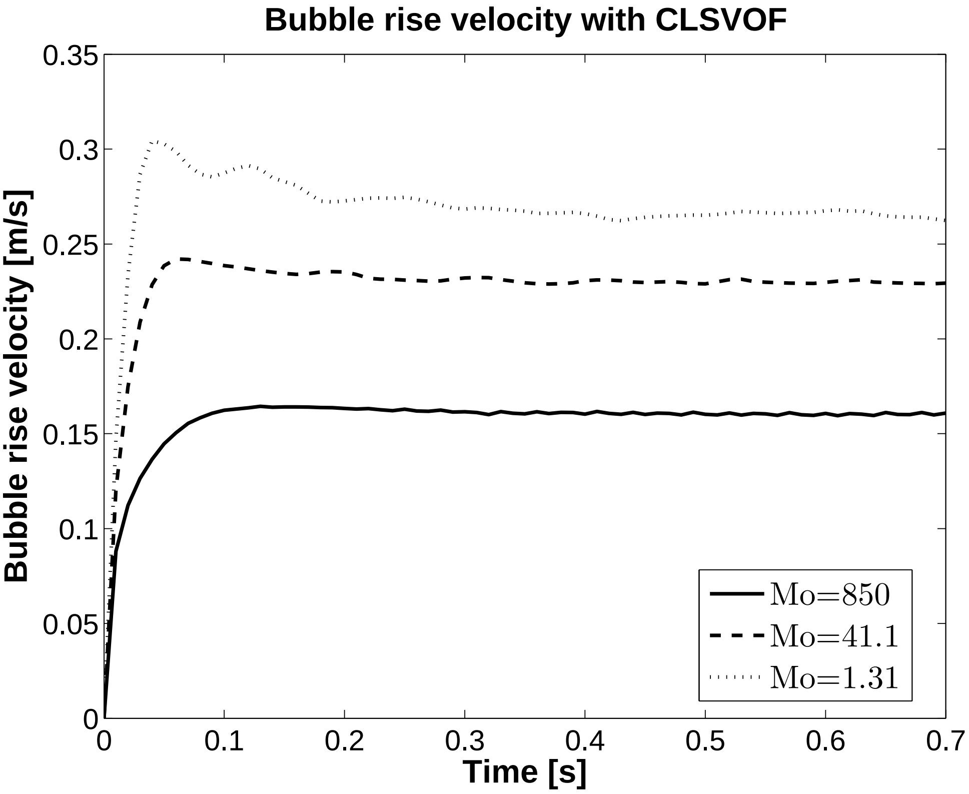
To further validate the code, we reran the 3D cases presented above together with other cases using a 2D-axisymmetric domain and CLSVOF. A $2.5d_e$x$20d_e$ domain was used with a grid of 50x400. For this grid, the average error in curvature calculation is 8%. For the 2D-axisymmetric simulation FLUENT requires that the axis has to lie on the X coordinate. Consequently, the axis boundary condition was used at $y = 0$ and the no slip wall boundary condition was used for the other three boundaries. The gravity vector used was $(-9.81, 0.0)$. We ran additional three cases besides the ones presented above: Morton numbers 266, 5.51 and 0.103. Table (5.5) shows a comparison between the experimental Re number and the numerical Re number obtained with our CLSVOF. Besides Case III we predicted the bubble rise velocity within 5%. Bhaga and Weber [9] also measured the bubble height h_b and width w_b. Fig. (5.19) depicts the comparison between the measured bubble heights and widths and the predicted bubble heights and widths. A very good agreement is achieved, although for the last case the bubble height is over predicted by 12%, which is due to the development of the nonphysical skirt observed in our simulations and also reported by other authors [26,36]. Fig. (5.19) shows also the ±7% experimental uncertainty reported by Bhaga and Weber [9]. Fig. (5.20) depicts the comparison between the measured drag coefficient, C_D, and the predicted one by CLSVOF. A good agreement is obtained, except for the Case VI where the drag coefficient is under predicted by approx. 20%. This discrepancy could be explained by the fact that due to the development of the skirt reported above, several small bubbles are formed behind the initial bubble, reducing the equivalent diameter of the bubble. Fig. (5.21) shows the bubble contour together with the velocity profile for Case IV in Table (5.5).

Table 5.5: 2D-axisymmetric bubble rise velocity with CLSVOF compared to Bhaga and Weber [9] data.

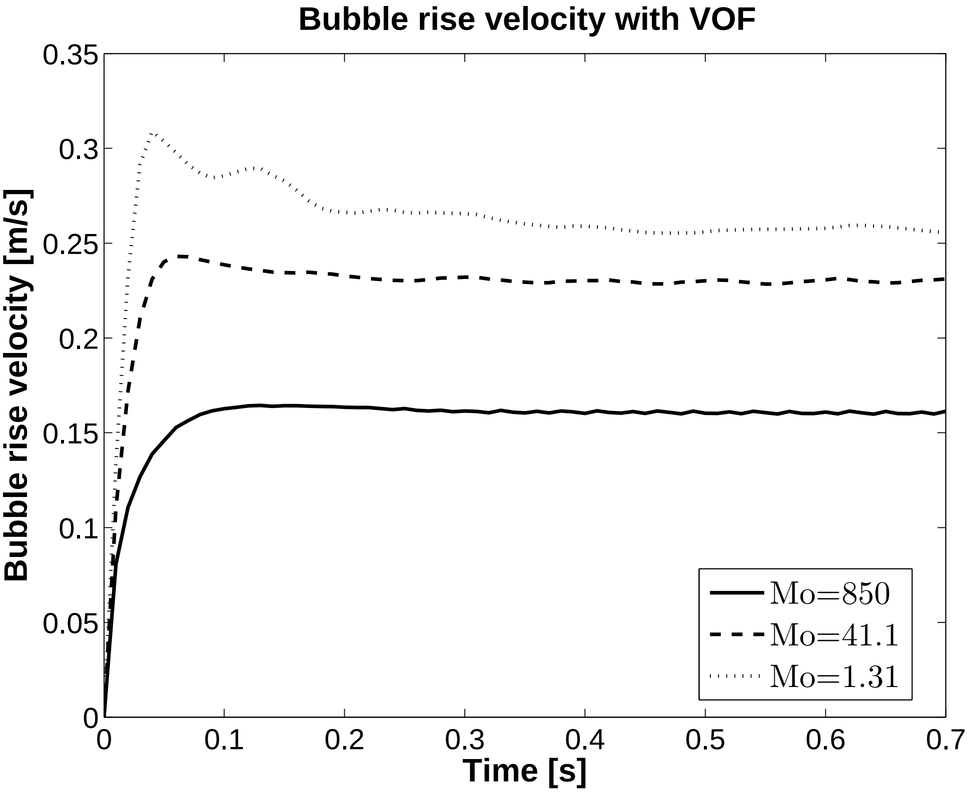
	Morton	Re_{exp}	Re_{num}	error[a][%]
Case I	850	2.09	2.065	1.2
Case II	266	3.57	3.52	1.4
Case III	41.1	7.16	6.47	9.6
Case IV	5.31	13.3	12.58	5.4
Case V	1.31	20.4	19.38	5
Case VI	0.103	42.2	40.04	5.1

---

[a] error $= \frac{Re_{exp} - Re_{num}}{Re_{exp}}$

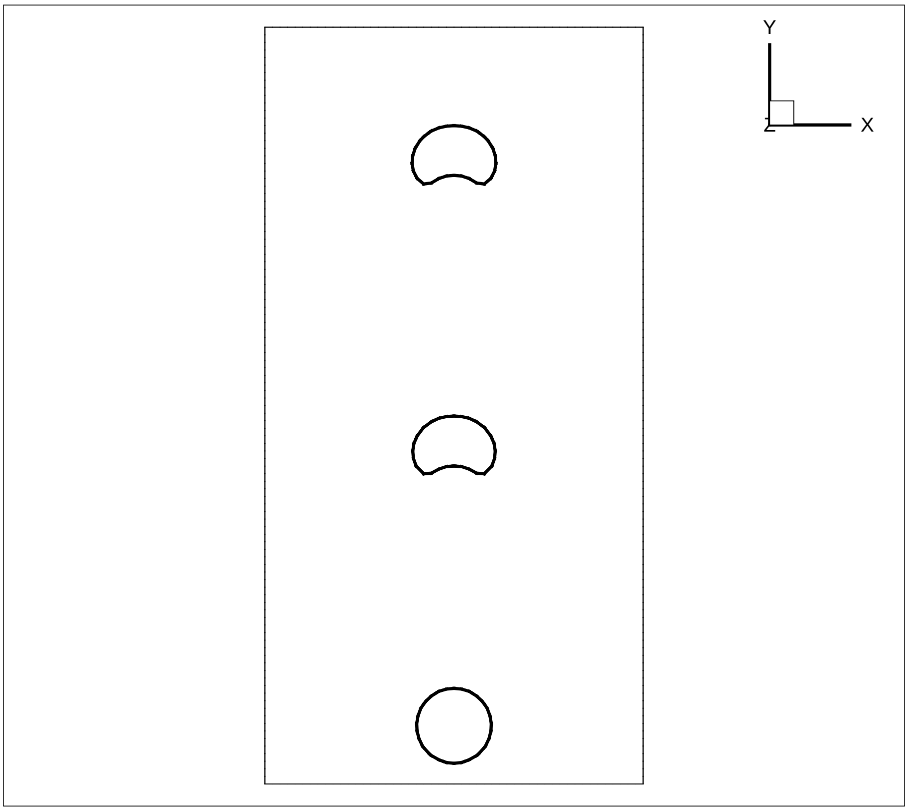


(a) CLSVOF

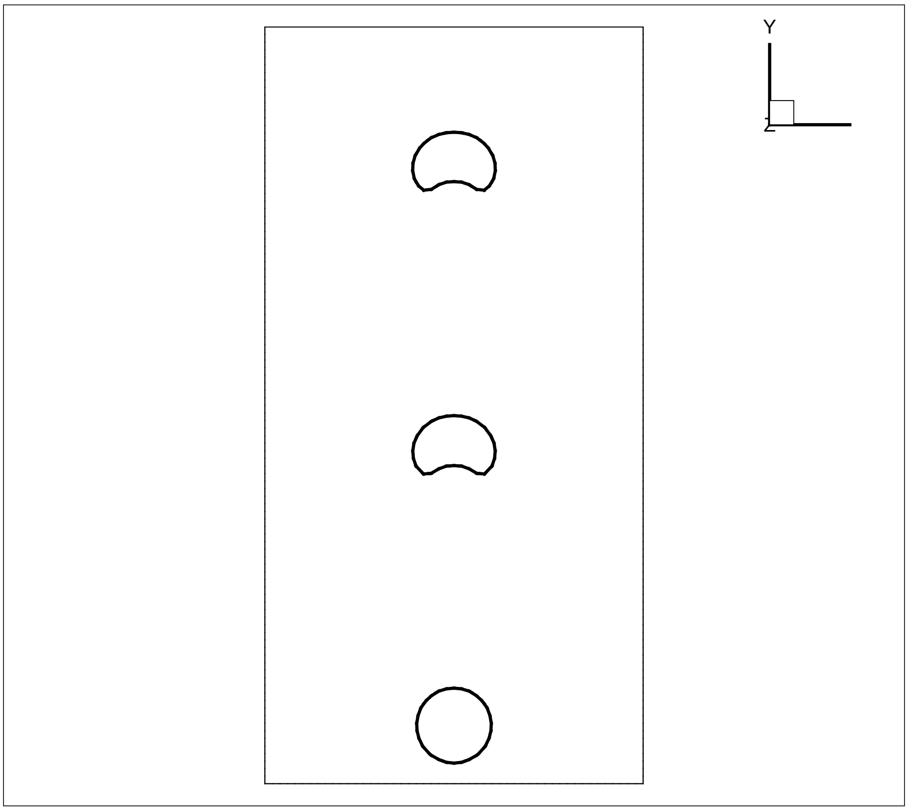


(b) VOF

Figure 5.15: Bubble rise velocity as a function of time with CLSVOF and VOF.

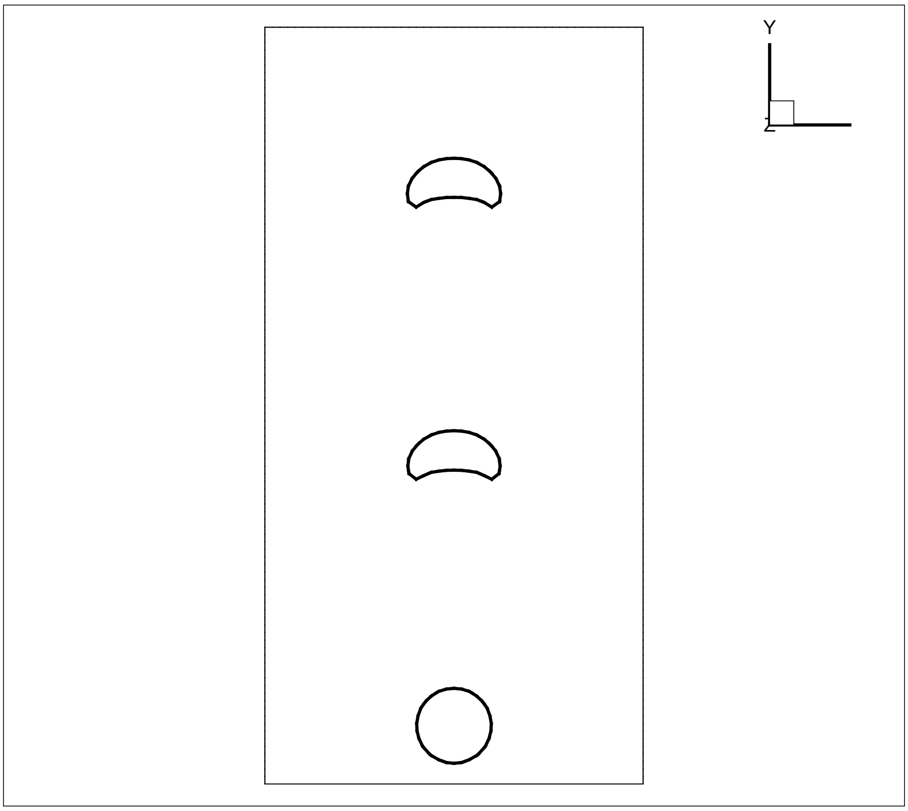


(a) CLSVOF

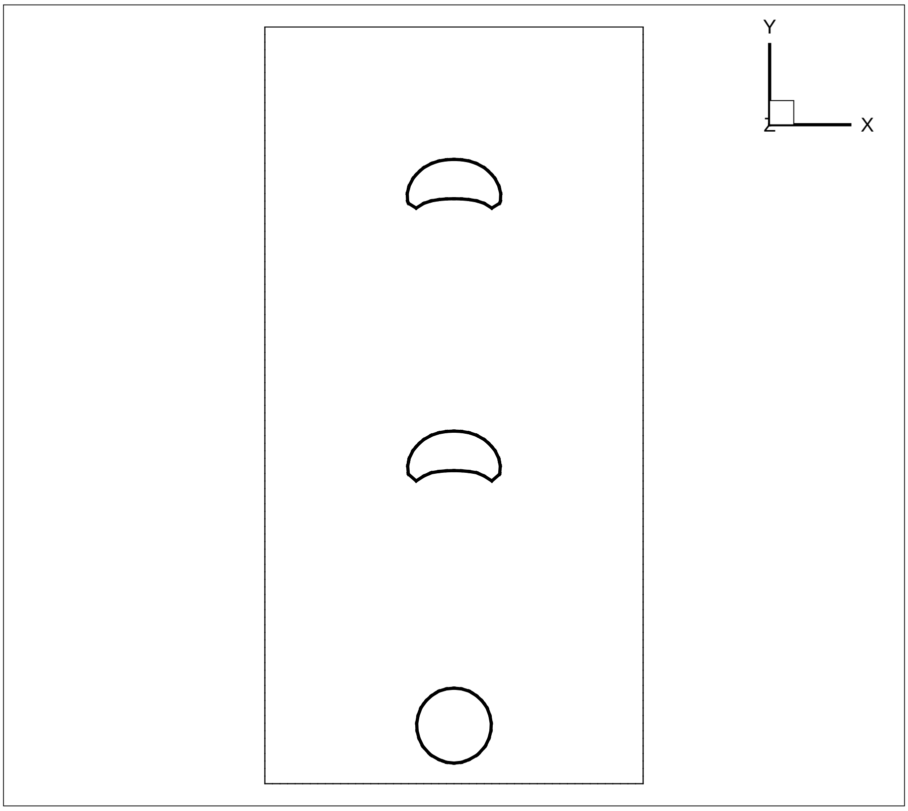


(b) VOF

Figure 5.16: Interface position after t=0s, 0.6s and 1.2s, with CLSVOF and VOF for Morton number Mo=850.

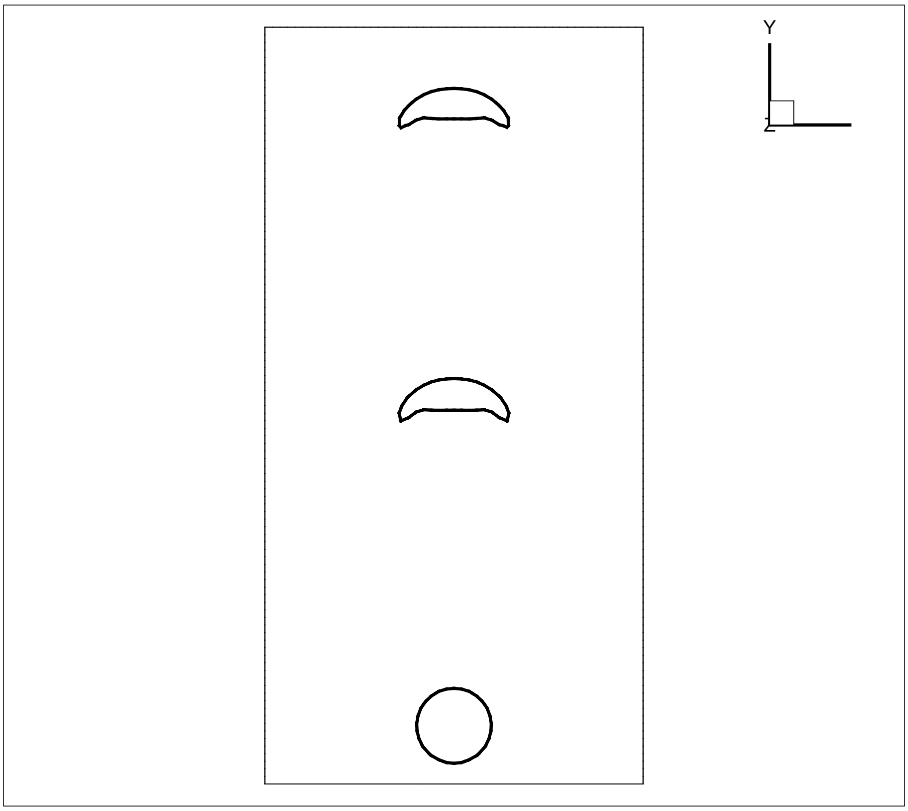


(a) CLSVOF

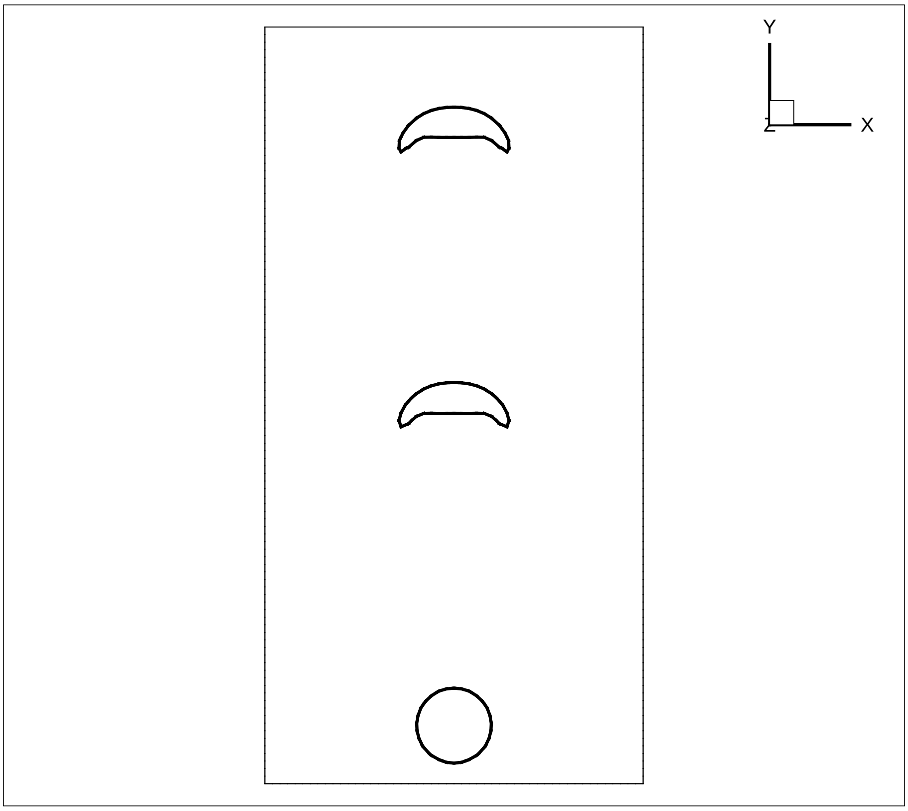


(b) VOF

Figure 5.17: Interface position after t=0s, 0.4s and 0.8s, with CLSVOF and VOF for Morton number Mo=41.1.



(a) CLSVOF



(b) VOF

Figure 5.18: Interface position after t=0s, 0.4s and 0.75s, with CLSVOF and VOF for Morton number Mo=1.31.

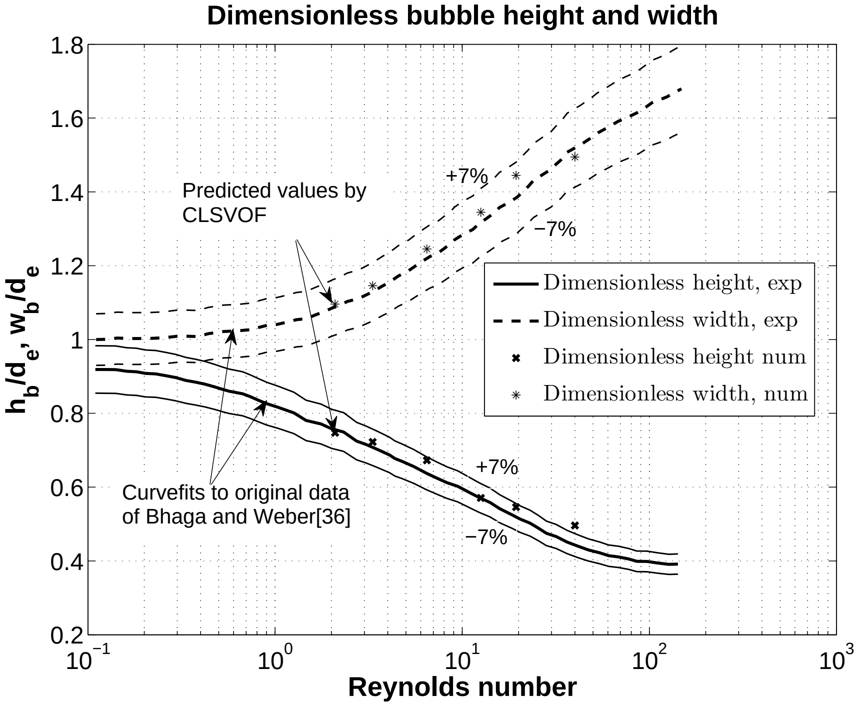


Figure 5.19: Dimensionless height and width of the bubble.

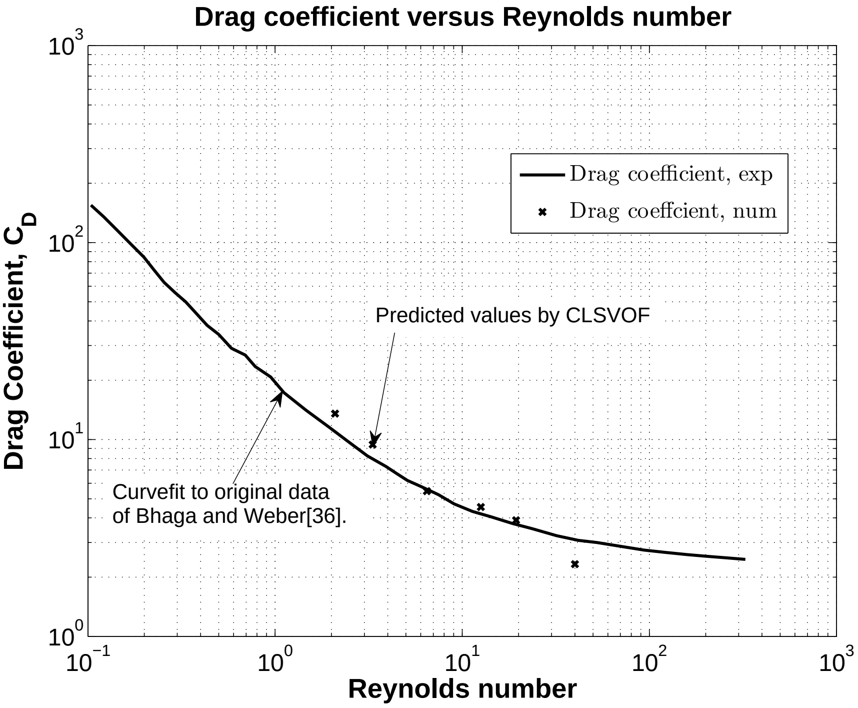


Figure 5.20: Drag coefficient versus Reynolds number.

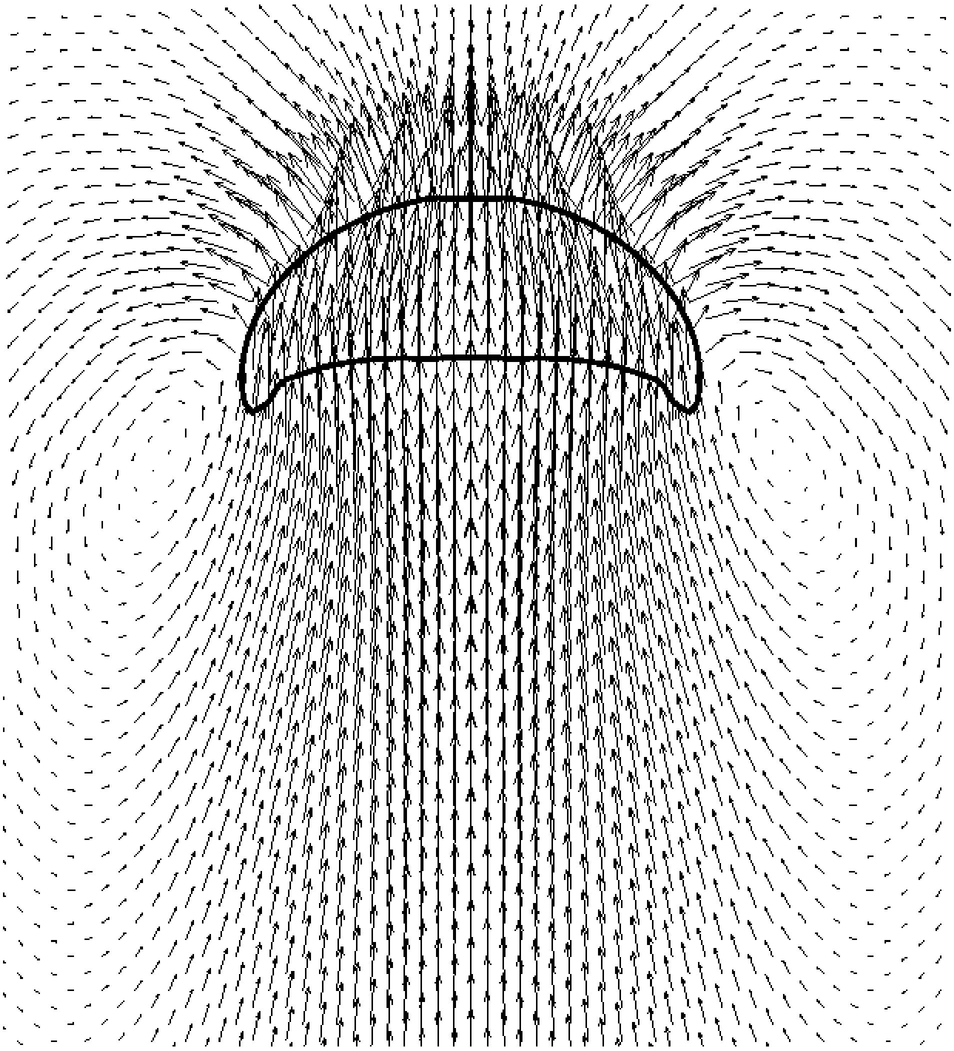


Figure 5.21: Bubble contour and velocity profile for Morton number 5.31.

Air-oil simulations

Hnat and Buckmaster [10] studied experimentally rising air bubbles in clear mineral oil. We simulated two cases here: the first case assumed an air bubble of 6 mm diameter rising in clear mineral oil which corresponds to case A in Table 1 from Hnat and Buckmaster [10]; the second case considered an air bubble of 8.3 mm diameter rising in clear mineral oil which corresponds to case B in the same Table 1 from Hnat and Buckmaster [10]. A 2D-axisymmetric domain was assumed of $5d_e$x$20d_e$ with a mesh of 100x400. The axis boundary condition was used at $y = 0$ and the no slip wall boundary condition was considered for the other three boundaries. Table (5.6) shows the physical properties of the fluids used in our simulations. Table (5.7) shows the comparison between the experimental bubble rise velocity and the numerical bubble rise velocity computed with our CLSVOF. A very good agreement was achieved. Fig. (5.22) shows the bubble shape predicted by CLSVOF for these two cases. A very good agreement is achieved compared to Fig. 1 from Hnat and Buckmaster [10].

Table 5.6: Physical properties of mineral oil and air.

Properties	Density kg/m^3	Viscosity Ns/m^2	Surface Tension N/m
mineral oil	875.5	0.118	0.0322
air	1.225	1.7894e-5	

Table 5.7: 2D-axisymmetric bubble rise velocity with CLSVOF compared to Hnat and Buckmaster [10] data.

	Diameter [mm]	Morton	v_{exp}[cm/s]	v_{num}[cm/s]	error[a][%]
Case I	6	0.065	21.5	21.6	0.4
Case II	8.3	0.065	27.0	26.0	3.7

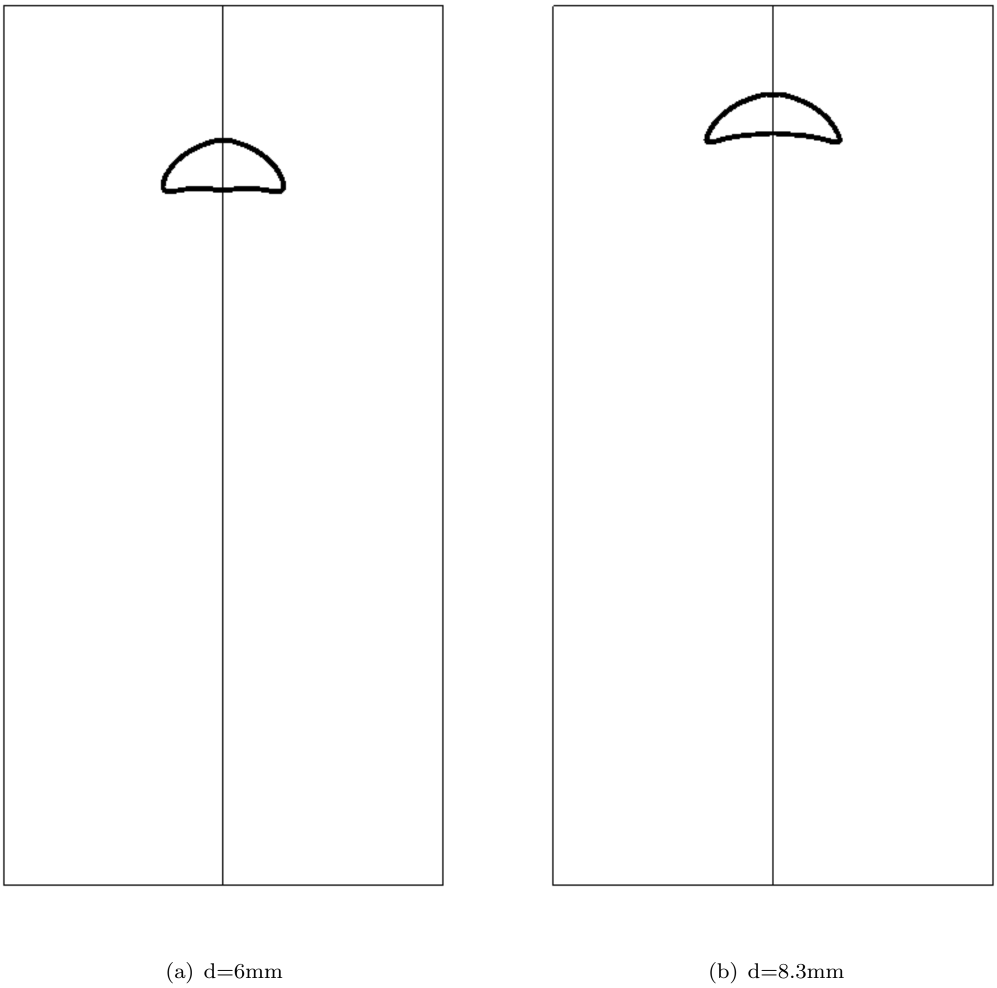
---

[a] error $= \frac{v_{exp}-v_{num}}{v_{exp}}$

Air-water simulations.

An air bubble rising in still water is one of the most difficult cases to simulate due to the physical properties of the fluids involved. Here an air bubble of 5mm in diameter rising in water is simulated in a parallelipipedic domain of $4d_e$x$32d_e$x$4d_e$ with a mesh of 40x320x40. The no slip wall boundary condition was used for all boundaries. The fluid properties are given in Table (5.8). For this case the dimensionless numbers Mo and Eo are: $Mo = 2.66e - 11$ and $Eo = 3.4$.

In Fig. (5.24) the interface position is presented for t=0s, 0.025s, 0.05s, 0.075s, 0.1s, 0.125s, 0.15s, 0.175s, 0.2s, 0.225s and 0.25s for both CLSVOF and VOF. For this simulation, we can see clearly the difference between CLSVOF and VOF: namely the shape and trajectory of the bubble



(a) d=6mm (b) d=8.3mm

Figure 5.22: Predicted bubble shape by CLSVOF for air bubble rising in clear mineral oil.

Table 5.8: Physical properties of the fluids.

Properties	Density kg/m^3	Viscosity Ns/m^2	Surface Tension N/m
water	1000	0.001	0.072
air	1.225	1.7894e-5	

appears to oscillate using the CLSVOF scheme, see Fig. (5.23), while for the VOF scheme, the trajectory of the bubble is unrealistically shifted to the right after half of the simulation time without any oscillation. Vortex shedding behind the rising bubble is thus better handled by the CLSVOF scheme than the VOF scheme. In Fig. (5.23) we plotted the x and y or z coordinates of the center of the bubble. To determine for example the x coordinate we used the following formula:

$$x = \frac{\sum F_g x_C V_C}{\sum F_g V_C} \tag{5.2}$$

where in Eq. (5.2) F_g is the gas volume fraction in the computational cell, x_C is the x coordinate of the computational cell and V_C is the cell volume. Similar relations were used for the y and z coordinates.

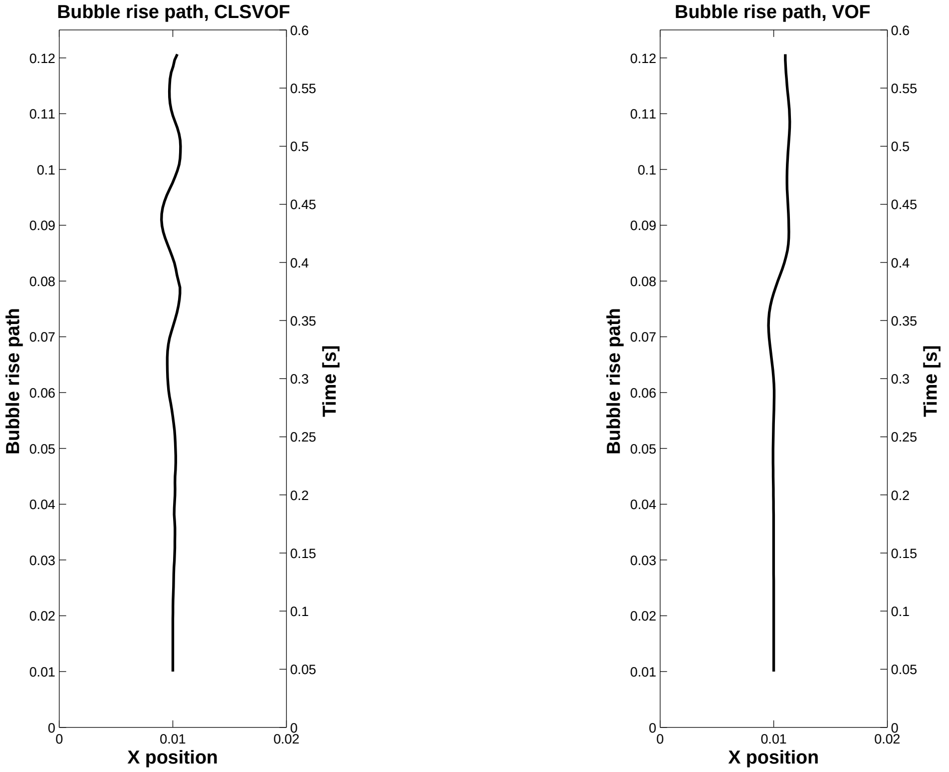


Figure 5.23: Characteristic bubble path with 3D CLSVOF and VOF.

Fig. (5.25) and Fig. (5.26) present the iso-contours of the pressure field calculated by CLSVOF and VOF at the time when the shift of the bubble path was observed according to the VOF application. From these figures we may conclude that the asymmetry of the iso-contours in Fig. (5.25) is attributed by surface interface oscillations which could be captured by LS implementation into VOF, while standalone VOF scheme promoted only one side kick. It is worth mentioning that in this case the mesh resolution is quite low, only 10 mesh cells along the bubble diameter, and thus we could observe the surface interface oscillations when using VOF method, which, according to Tomiyama [130] for a VOF method one needs to assign at least 14 mesh cells along the bubble diameter in order to capture the capillary waves on the interface.

Fig. (5.27) and Fig. (5.28) present the iso-contours of the pressure field calculated by CLSVOF and VOF at the following different snapshots: 0.455s, 0.46s, 0.475s, 0.5s, respectively. From these figures we may see how CLSVOF application promotes bubble tilting as well as interface waves which occur at different scales. This phenomenon appears realistic when compared to the experimental observation for particular bubble size. Fig. (5.28) proves that VOF itself is not able to capture small scale interfacial oscillations, presumably due to the discontinuities in the color function.

Comparisons of average bubble rise velocity, pitch and frequency of bubble helical motion values obtained with CLSVOF and VOF against the experimental data of Zun and Groselj [5] are shown in Table (5.9). The measured value of bubble rise velocity is 7.8% higher than predicted with CLSVOF and 12.4% higher than predicted with VOF, using standard surface tension of water in both cases. Comparison with the Peebles and Garber data [11] show also good agreement, see Table (5.12). Fig. (5.29) and Fig. (5.30) show the bubble contour and velocity profile at different times with a standard surface tension (see Table (5.8)) with CLSVOF and VOF, respectively. There is almost no vortex behind the bubble when using VOF, while LS implemented into VOF clearly captures the vortex created behind the bubble.

Table 5.9: Bubble rise velocity and bubble mean lateral displacement for different surface tensions.

Case	Lateral displacement [mm]	Rise velocity [m/s]	Pitch [m]	Frequency [s^{-1}]
CLSVOF[a]	0.885	0.218	0.0398	5.44
CLSVOF[b]	1.275	0.257	0.0283	9
CLSVOF[c]	0.725	0.177	0.0218	8
VOF[d]	1.7	0.209	-	-
Experiment[e]	2.0	0.235	0.047	5

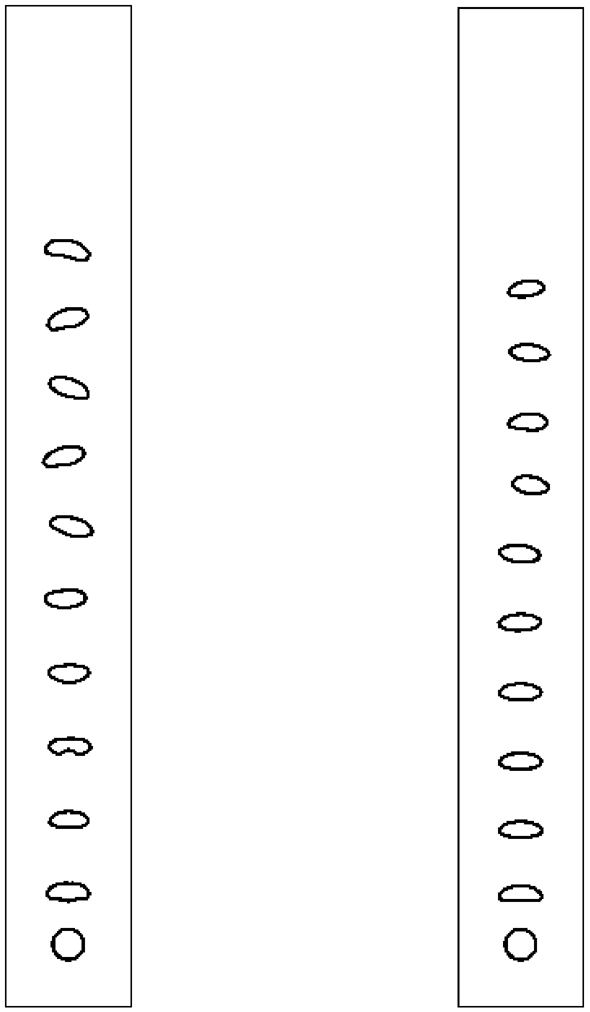
---

[a] Standard surface tension value (see Table Table (5.8)).
[b] Surface tension with 50% higher than the standard value.
[c] Surface tension with 50% lower than the standard value.
[d] Standard surface tension value.
[e] Zun and Groselj [5] Figs (3), (4) and (10).

From the bubble path we can also compute the mean lateral bubble displacement using the following formula:

$$s = \left(\frac{1}{m} \sum_{i=1}^{m} (x_i - x)^2 \right)^{0.5} \tag{5.3}$$

In Eq. (5.3) x_i denotes the bubble location in comparison with the datum location x of the rectilinear path, and m is the number of points taken into consideration. Zun and Groselj [5] obtained for the 5mm bubble a mean lateral displacement of 2mm. It should be noted here that their experimental observation of 5mm bubbles revealed random rocking in addition to periodic bubble displacement due to helical movement which contributed to rather large values of the mean lateral displacement measurements.



(a) CLSVOF (b) VOF

Figure 5.24: Interface position after t=0s, 0.05s, 0.1s, 0.15s, 0.2s, 0.25s, 0.30s, 0.35s, 0.40s, 0.45s and 0.5s, for an air bubble rising in water with CLSVOF and VOF.

Using VOF the mean lateral displacement obtained was almost double compared to CLSVOF: 0.885mm for CLSVOF and 1.7mm for VOF. Comparison of VOF results, however, is much more questionable since the VOF simulation did not reveal any tendency towards bubble lateral space displacement except the shift of the rectilinear bubble path after half of the simulation time. One may talk also about the weakness of CLSVOF application which at this stage enables us to simulate only periodic bubble movement. The pitch p of the bubble path can be easily determined in CLSVOF application. From the pitch and the bubble rise velocity the mean oscillation frequency in bubble motion can be calculated easily: f = v/p. Here, VOF application failed due to the aforementioned shift in rectilinear path, while CLSVOF predicted the frequency within 9% of accuracy.

Because surface tension plays an important role for determining the bubble rise velocity and the bubble mean lateral displacement, an additional two cases were run: one where the surface tension was increased by 50%, and the other where the surface tension was decreased by 50%. Table (5.9) also shows the results obtained with the two latter cases. As expected, increasing the surface tension decreases energy dissipation at the surface interface and thus enables higher rise velocities and vice versa.

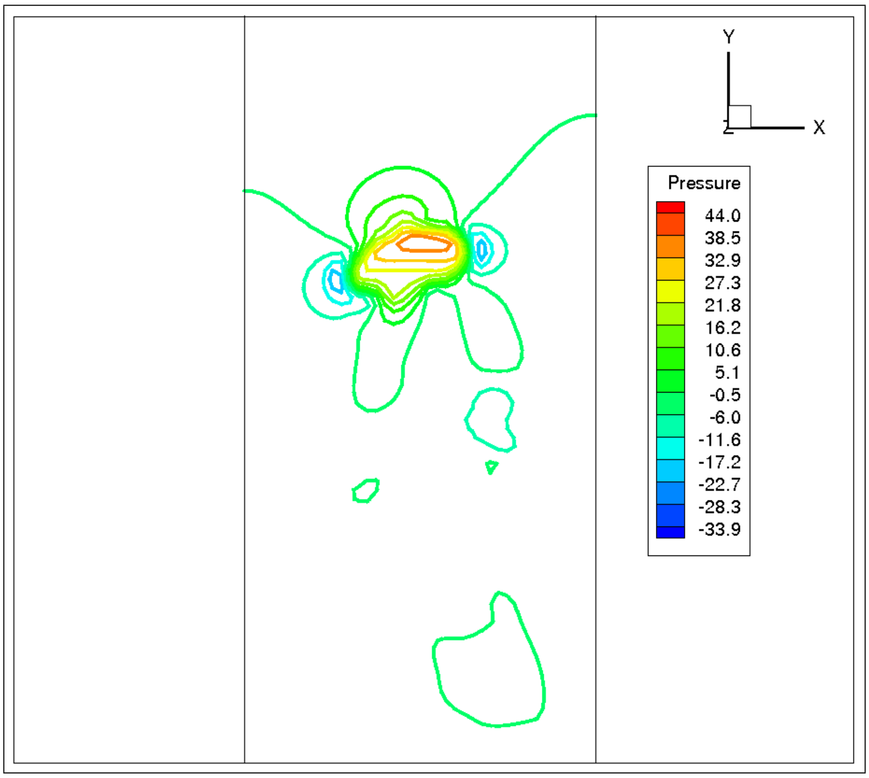
Using dimensionless analysis and CLSVOF, the bubble rise velocity was also computed for bubble diameters of 2, 3 and 4mm. To do this, the same domain and bubble size were used like for the case of 5mm, modifying the fluid properties. For these cases the moving wall approach was utilized, meaning that the bubble was set in the middle of the domain (0.01,0.08,0.01) m, the vertical walls were moving downward with the rising velocity, while the top boundary is not anymore a wall but a velocity inlet, and the bottom boundary is an outlet. We used the Π theorem defining the following 6 dimensionless independent groups:

$$\Pi_1 = \frac{\mu_v}{\mu_l},\ \Pi_2 = \frac{\rho_v}{\rho_l},\ \Pi_3 = \frac{\rho_l g L^2}{\sigma},\ \Pi_4 = \frac{t\mu_l}{\rho_l L^2},\ \Pi_5 = \frac{uL\rho_l}{\mu_l},\ \Pi_6 = \frac{g\mu_l^4}{\rho_l \sigma^3} \tag{5.4}$$

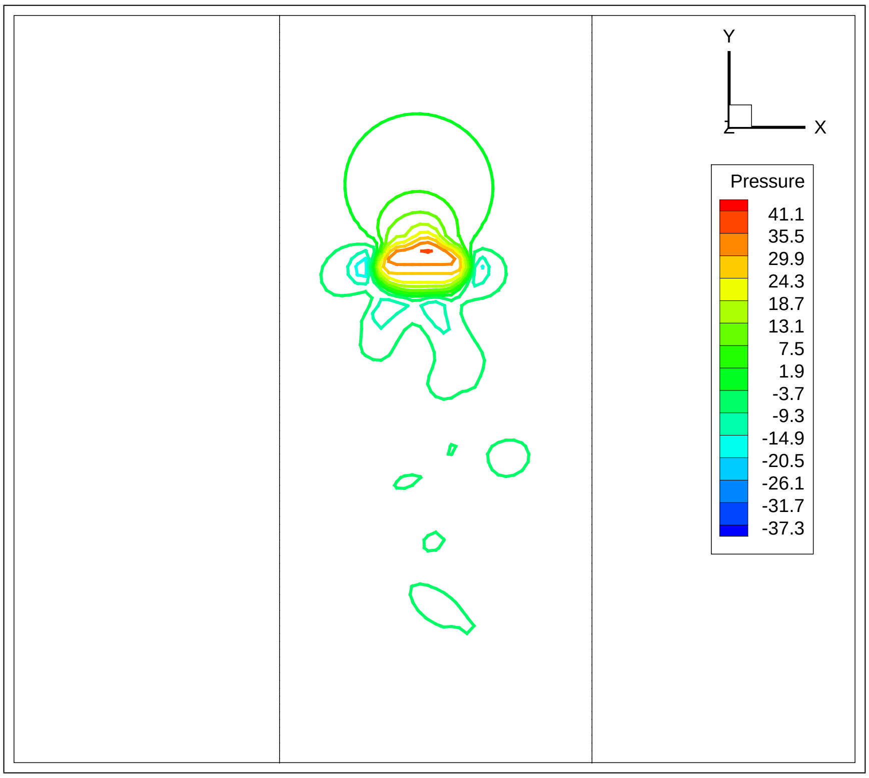
After some mathematical manipulations, for each case we obtained different fluid properties to be used with the domain for the 5mm bubble case. These properties are shown in Table (5.10). Fig. (5.31) shows the comparison between the computed bubble rise velocity with CLSVOF and the measured one from [5]. A very good agreement is obtained, with a maximum relative error of 5.1%, see Table (5.11).

Table 5.10: Physical properties of the fluids for different diameters.

Diameter	ρ_l	ρ_v	μ_l	μ_v	σ	g
mm	kg/m^3	kg/m^3	Ns/m^2	Ns/m^2	N/m	m/s^2
5	998	1.225	0.001003	1.7894e-5	0.072	9.81
4	638.83	0.784	8.971e-4	1.6e-5	0.072	9.81
3	359.34	0.441	7.7692e-4	1.386e-5	0.072	9.81
2	159.71	0.196	1.1317e-5	1.1317e-5	0.072	9.81

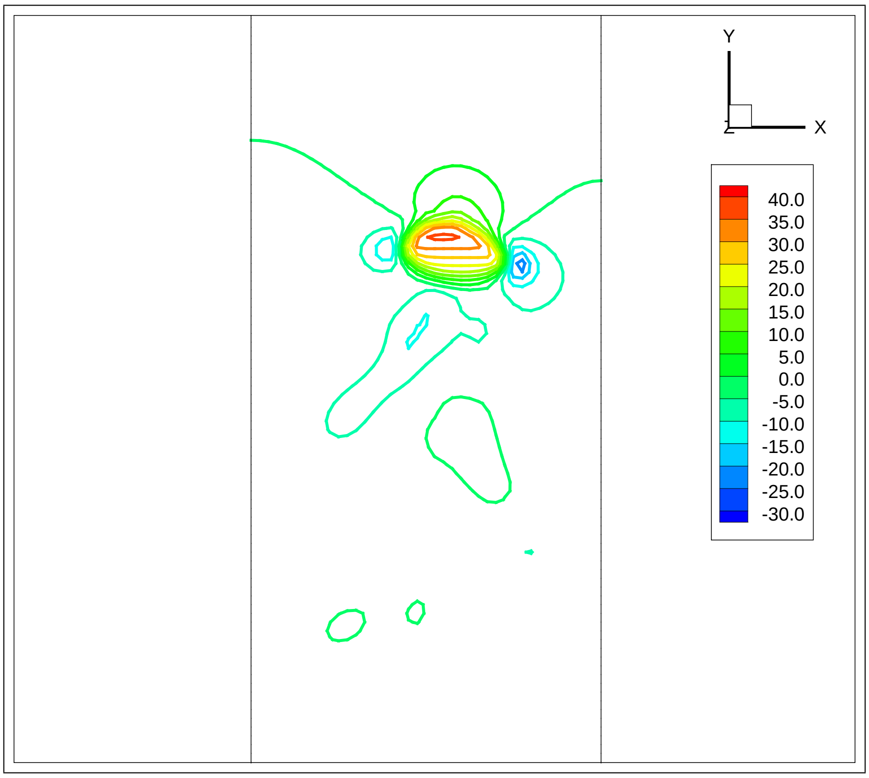


(a)

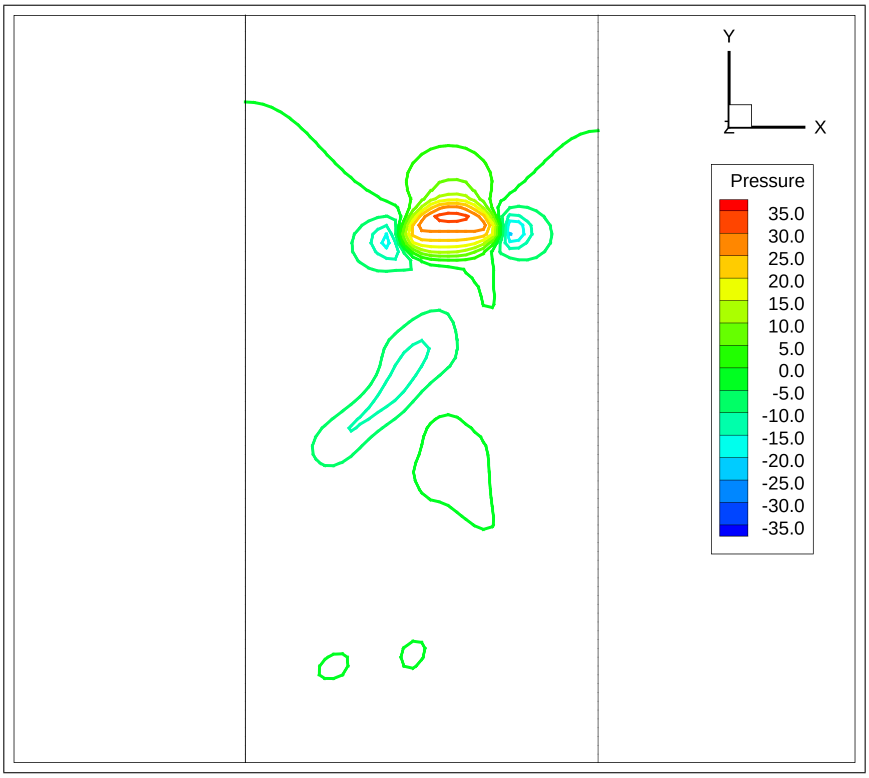


(b)

Figure 5.25: Pressure contour for 5mm bubble with CLSVOF, time=0.35s, 0.37s.

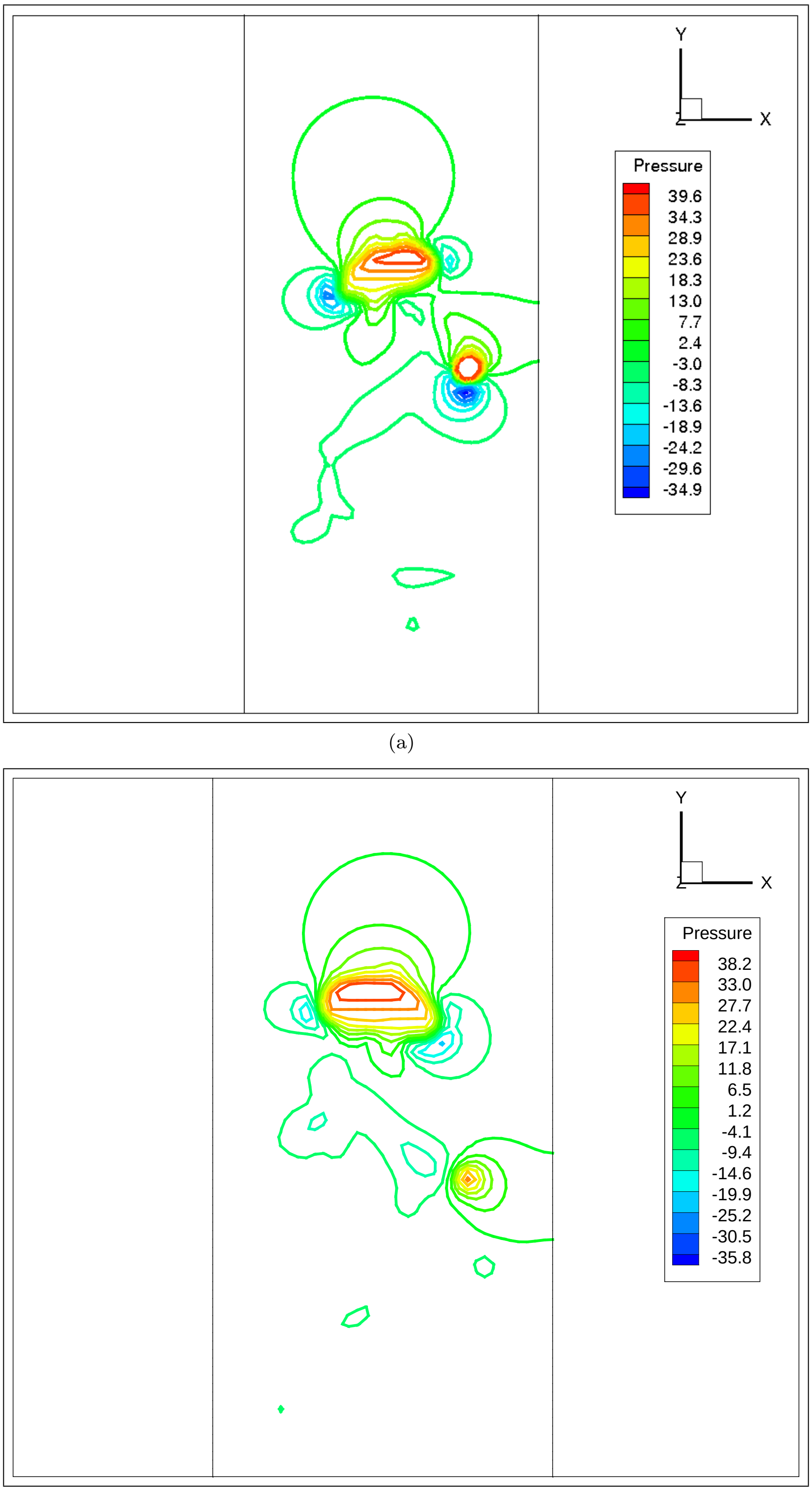


(a)



(b)

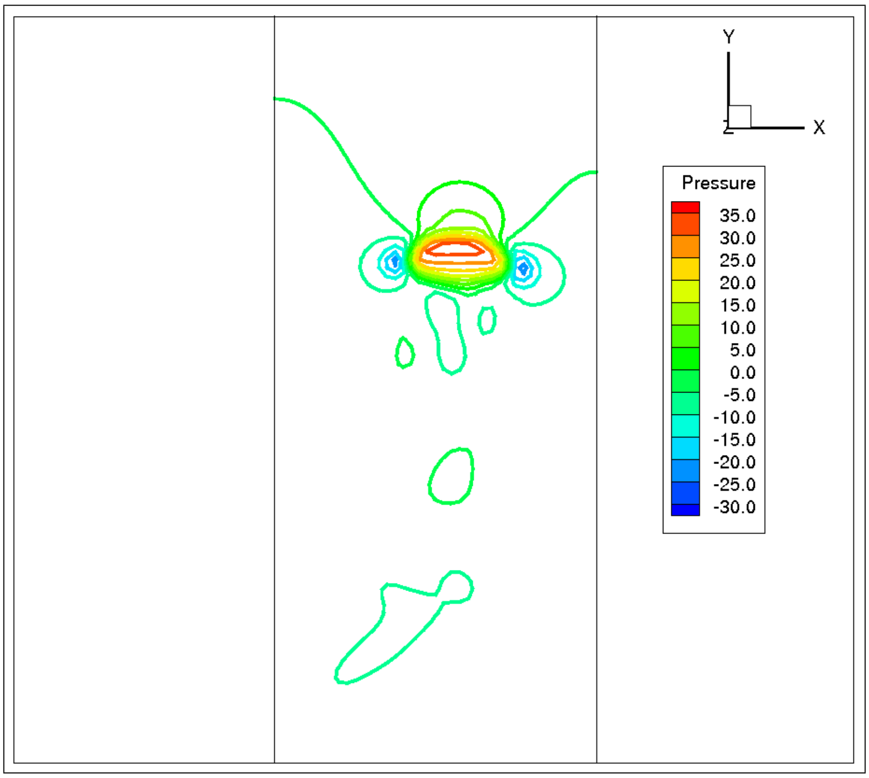
Figure 5.26: Pressure contour for 5mm bubble with VOF, time=0.35s, 0.37s.



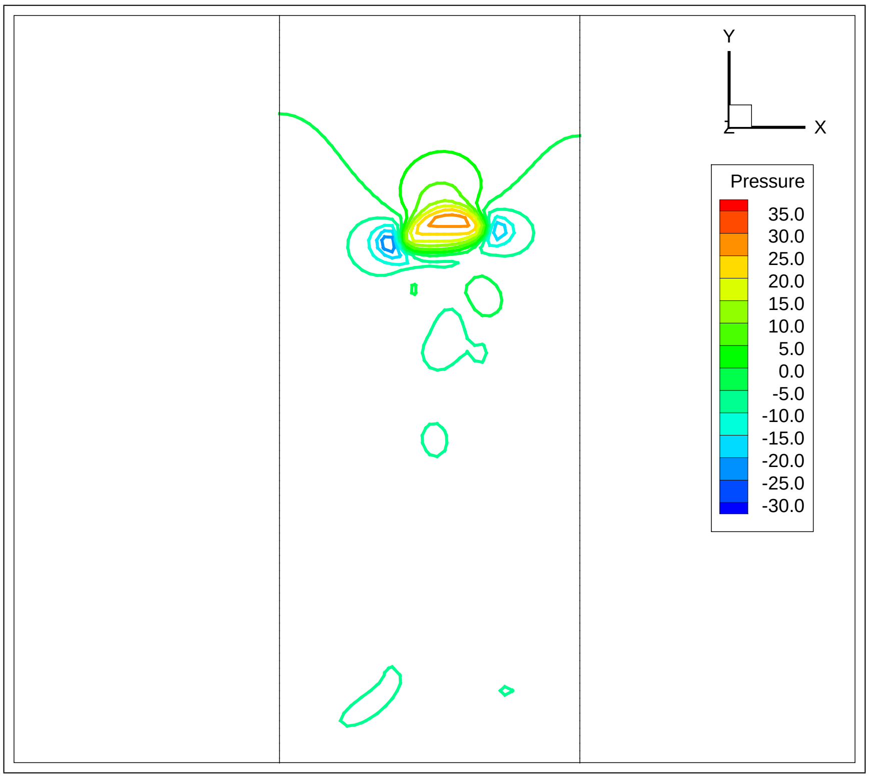
(a)

(b)

Figure 5.27: Pressure contour for 5mm bubble with CLSVOF, time=0.455s, 0.5s.

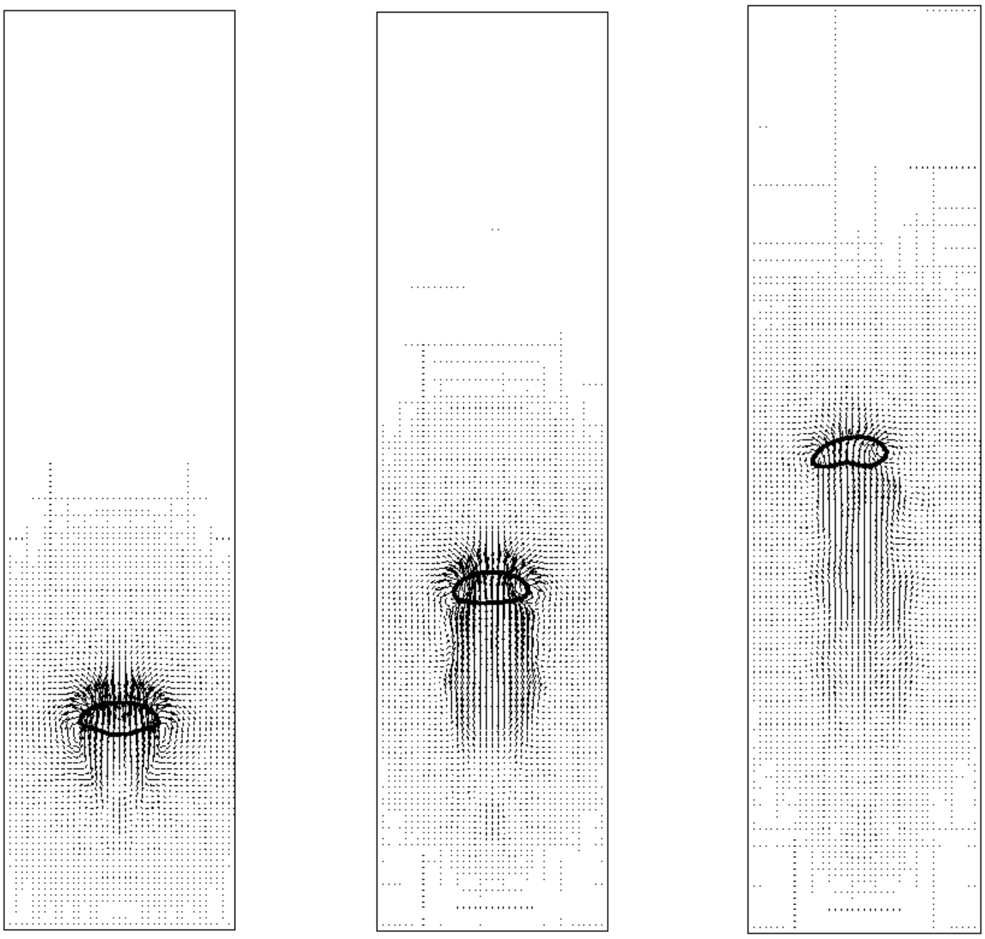


(a)



(b)

Figure 5.28: Pressure contour for 5mm bubble with VOF, time=0.455s, 0.5s.



(a) t=0.05s (b) t=0.1s (c) t=0.15s

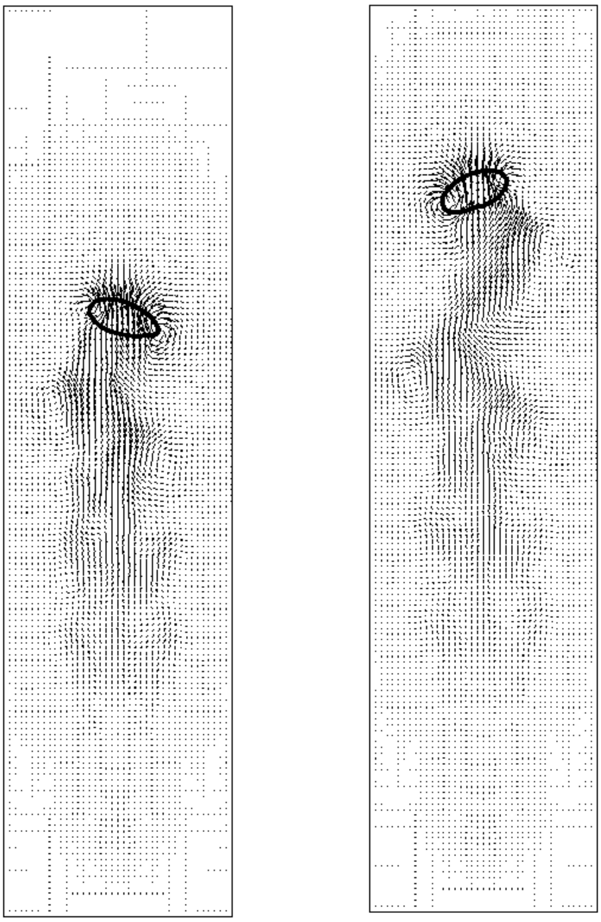
Figure 5.29: Interface position and velocity field with 3D CLSVOF for 5mm bubble.

Table 5.11: Bubble rise velocities vs. data of Zun [5].

Diameter	v_{exp}	$v_{num,CLSVOF}$	$v_{num,VOF}$	ϵ_{CLSVOF}	ϵ_{VOF}
mm	m/s	m/s	m/s	%	%
2	0.310	0.311	0.2904	0.2258	6.32
3	0.28	0.278	0.2707	0.7143	3.32
4	0.25	0.252	0.2444	0.8000	2.24
5	0.235	0.223	0.218	5.1064	7.23

Table 5.12: Bubble rise velocities vs. data of Peebles and Garber [11].

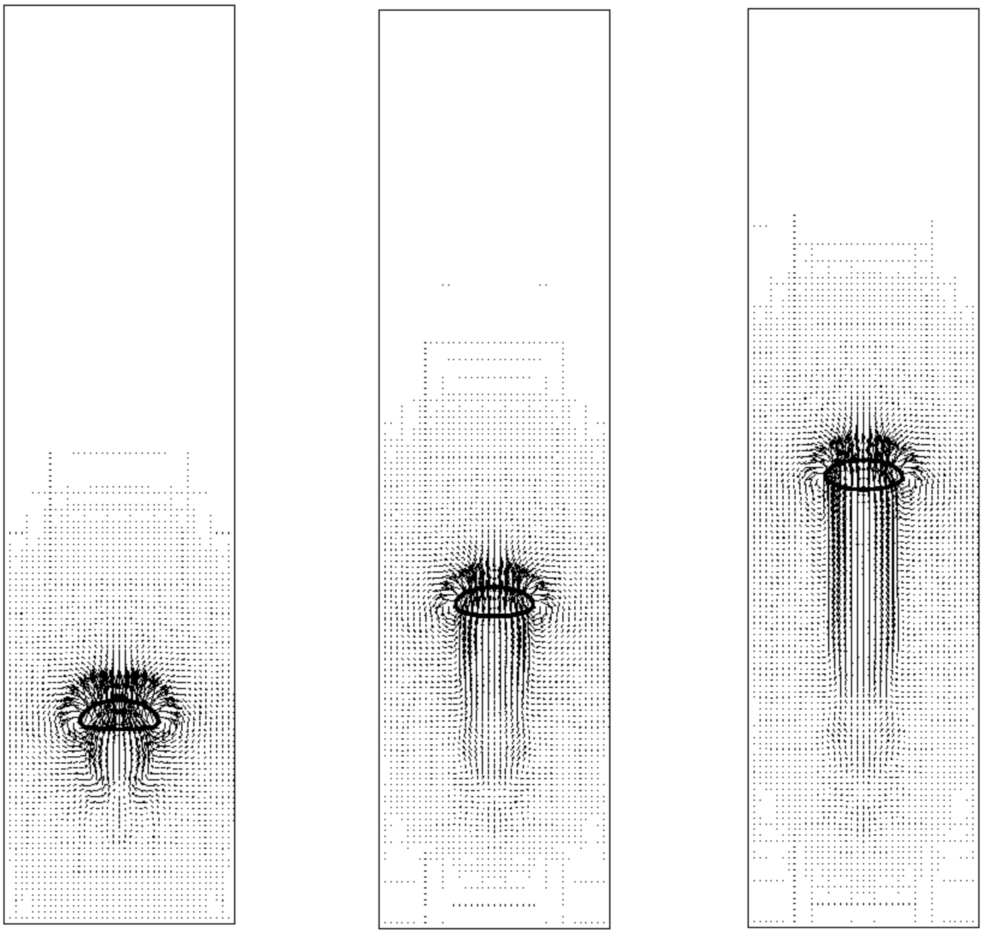
Diameter	v_{exp}	$v_{num,CLSVOF}$	$v_{num,VOF}$	ϵ_{CLSVOF}	ϵ_{VOF}
mm	m/s	m/s	m/s	%	%
2	0.289	0.311	0.2904	7.07	0.48
3	0.296	0.278	0.2707	6.08	8.54
4	0.256	0.252	0.2444	1.56	4.53
5	0.229	0.223	0.218	2.62	4.80



(d) t=0.2s (e) t=0.25s

Figure 5.29: Interface position and velocity field with 3D CLSVOF for 5mm bubble, cont.

We also compared the bubble rise velocity with the predictions given by Wallis [6]. The first three cases fall into the transition zone between region 3 and 4, while the 5mm case falls into region 4. This can be seen in Fig. (5.32). According to a recent finding by Tomiyama *et al.* [131], the terminal speed of free rising bubbles is apt to widely scatter between region 3 and 4 in low viscosities systems, depending upon the way of bubble is released, or in other words, due to the initial conditions. A drastic increase in bubble drag, that is usually referred in the literature as a main cause to reduce the bubble free rising speed, can be also caused by a small initial shape deformation. When a bubble is released from a nozzle with small initial shape deformation, the bubble terminal speed takes a low value, (close to the extrapolated region 4 according to Wallis) and bubble motion is apt to be either rectilinear or zigzag. To the contrary, when a bubble is released with a large initial shape deformation, the bubble terminal speed increases and the motion is likely to be either helical or rectilinear. Initial conditions in our numerical studies certainly did not cover this variety of possibilities that could be tested in this respect. A maximum error of 18% obtained in our simulation may therefore be regarded as satisfactory, see Table (5.13).

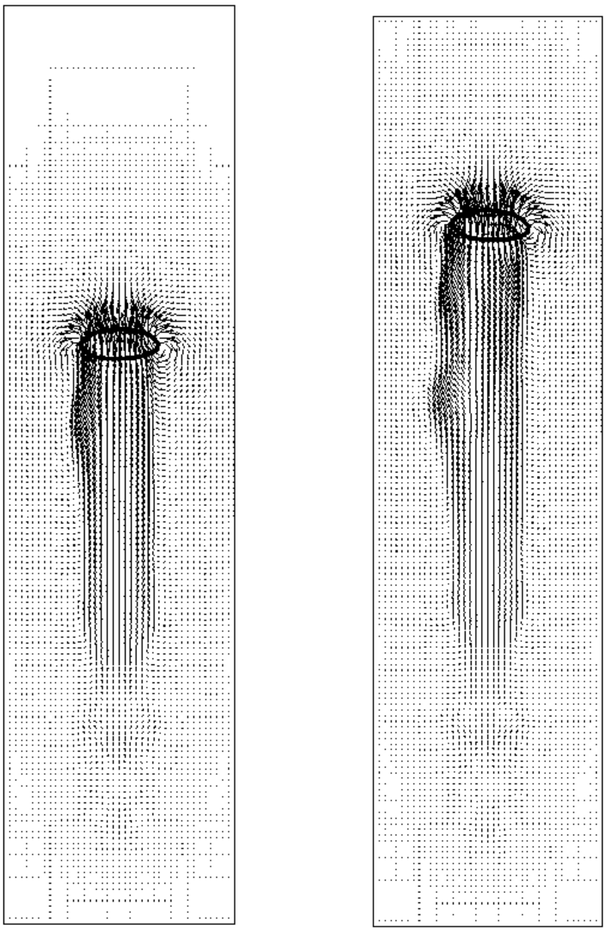


(a) t=0.05s (b) t=0.1s (c) t=0.15s

Figure 5.30: Interface position and velocity field with 3D VOF for 5mm bubble.

Table 5.13: Bubble rise velocities vs. theory of Wallis [6].

Radius	v	r^*	$v^*_{num,CLSVOF}$	$v^*_{num,VOF}$	v^*_3	v^*_4	ϵ_{CLSVOF}	ϵ_{VOF}
mm	m/s	-	-	-	-	-	%	%
1.0	0.311	21.32	14.5	13.55	17.72	10.75	18	23.5
1.5	0.278	31.99	12.97	12.63	14.47	10.75	10.3	12.72
2.0	0.252	42.66	11.75	11.40	12.53	10.75	6.2	9.02
2.5	0.223	53.32	10.4	10.17	11.2	10.75	3.2	5.42



(d) t=0.2s (e) t=0.25s

Figure 5.30: Interface position and velocity field with 3D VOF for 5mm bubble, cont.

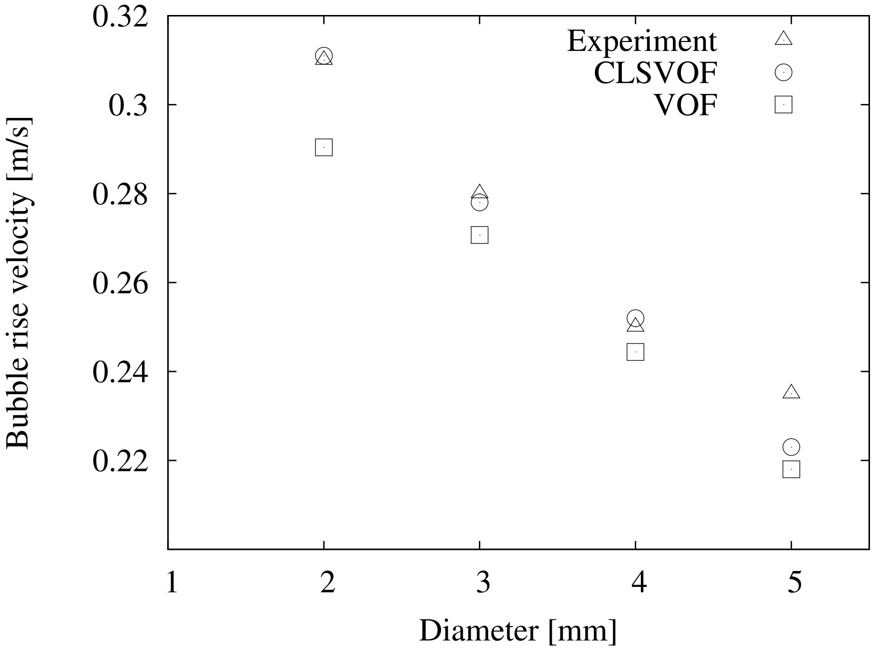


Figure 5.31: A comparison between bubble rise velocity obtained experimentally by Zun and Groselj [5] and bubble rise velocity obtained with CLSVOF and VOF.

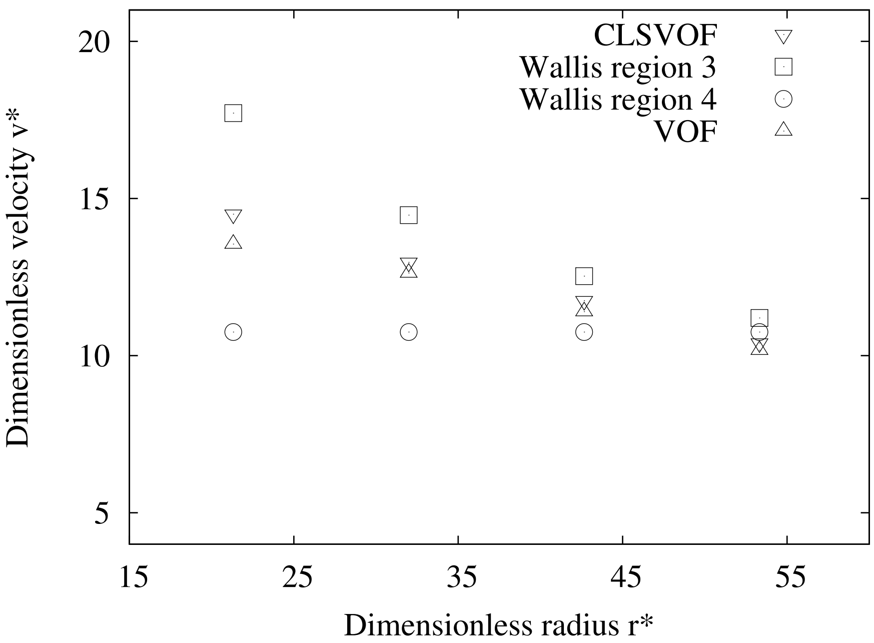


Figure 5.32: A comparison between bubble rise velocity obtained with CLSVOF, VOF and bubble rise velocity predicted by Wallis [6].

5.2 Numerical validations with dynamic contact angle model

In this section several 2D and 3D simulations are presented for air-water and a coated silicon wafer surface.

5.2.1 2D examples

In the present study, a semi-circular droplet of 2cm diameter is placed on a wall of a rectangular domain of 8x2.5cm at (4.0,0.5) cm with a gravity vector $(0.0, -9.81)$ m/s^2. The fluids used are air and water, see Table (5.8) for their physical properties. Fang *et al.* [116] have measured the contact angle for air-water on a coated silicon wafer surface and found that the advancing contact angle is around 105° and that the receding contact angle is around 65°. Consequently, in our simulations we used the above mentioned values as limiters for the dynamic contact angle. A no slip wall boundary condition was used for all boundaries. Beside this, for the bottom boundary the dynamic contact angle was computed and prescribed. Three different grids were used: 80x25, 160x50 and 320x100. Fig. (5.33) shows the bubble contour at different times using a grid of 80x25. Fig. (5.34) and Fig. (5.35) show the bubble contour on a grid of 160x50 and on a grid of 320x100, respectively. Similar shapes were found numerically by Manservisi and Scardovelli [119] and experimentally by Lim *et al.* [132] and Grundke *et al.* [118]. Grundke *et al.* [118] measured a contact angle between a sessile water drop and a silicone surface between 70° and 120°.The error E_1^i between succesive mesh refinements is computed using the L_1^i norm, which is well suited for the interfacial flow problems:

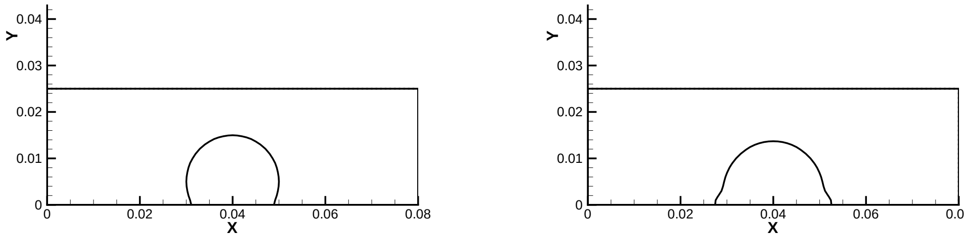
$$E_1^i = \int_\Omega |H(F_r) - H(F_c)| d\Omega \tag{5.5}$$

where $H(F)$ is the discontinuous Heaviside function, and F_r and F_c are solutions from, respectively, refined and coarse grids. For the numerical integration of Eq. (5.5) a domain of 1000x1000 mesh points was used and the solution was interpolated from the actual domain to the computational one. The results obtained are shown in Table (5.14). As can be seen, increasing the resolution of the grid leads to a decrease in computation error.

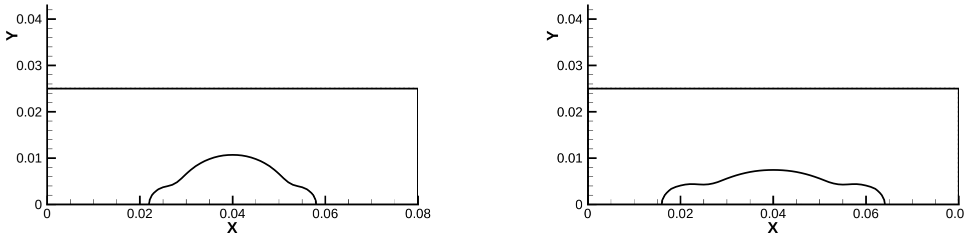
Table 5.14: Error between successive mesh refinements for 2D air-water simulations with gravity vector (0,-9.81) and (-9.81,0).

Grid	$E_1^i, (0, -9.81)$	$E_1^i, (-9.81, 0)$
80x25	N.A.	N.A.
160x50	1.22e-2	7.049e-3
320x100	3.359e-3	2.829e-3

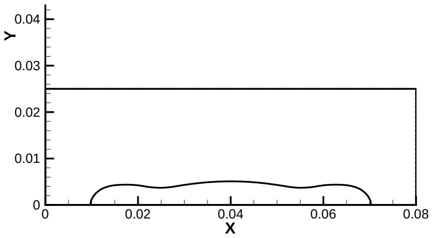
The second simulated case was similar to the above, however the gravity vector is (-9.81,0)m/s^2. Thus we expect now to see different contact angles at the two droplet extremities, using the



(a) t=0s (b) t=0.02s



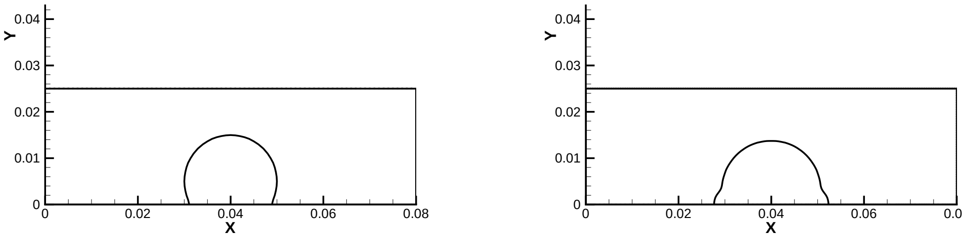
(c) t=0.04s (d) t=0.06s



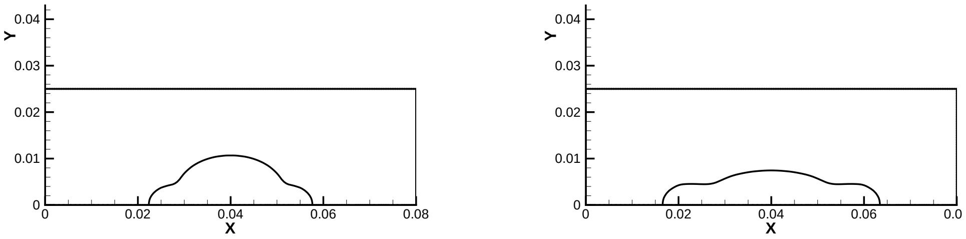
(e) t=0.08s

Figure 5.33: Snapshots of the 2D water drop for a grid of 80x25 and gravity vector (0.0,-9.81).

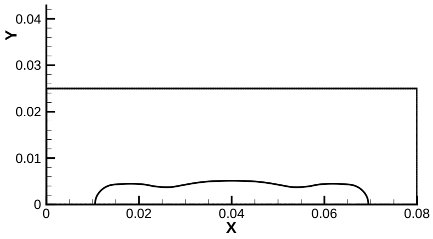
same boundary conditions as in the previous case and three different grids: 80x25, 160x50 and 320x100. Once again the error decreased with increasing mesh densities. Fig. (5.36) shows the droplet contour at different times using a grid of 80x25. As can be seen, the drop moves from right to left under the action of gravity. Fig. (5.37) and Fig. (5.38) show the bubble contour on grids of 160x50 and 320x100, respectively. The error E_1^i between successive mesh refinements is computed as is explained above, and the results are shown in Table (5.14). Fig. (5.39) depicts the velocity profile near the contact line for the grid of 320x100 and gravity vector of (0.0,-9.81).



(a) t=0s (b) t=0.02s

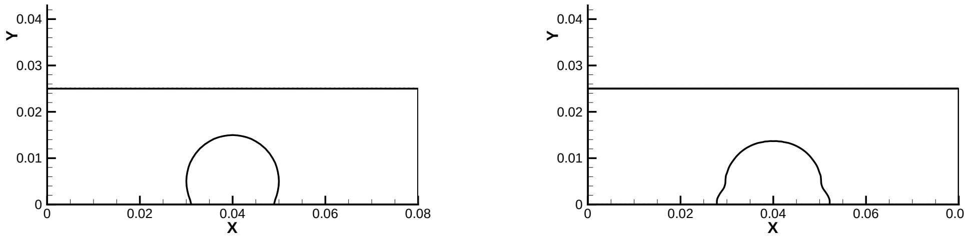


(c) t=0.04s (d) t=0.06s

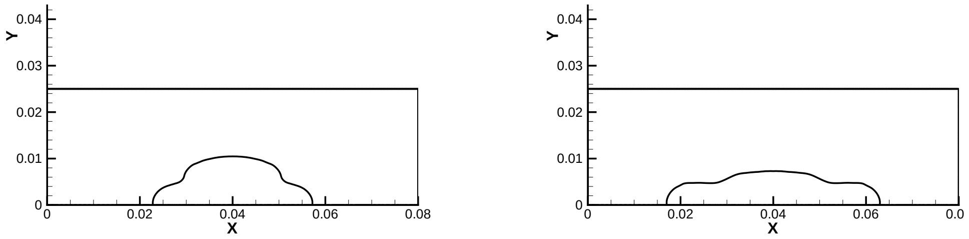


(e) t=0.08s

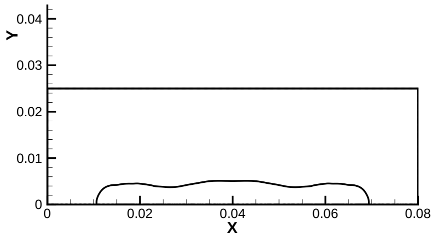
Figure 5.34: Snapshots of the 2D water drop for a grid of 160x50 and gravity vector (0.0,-9.81).



(a) t=0s (b) t=0.02s

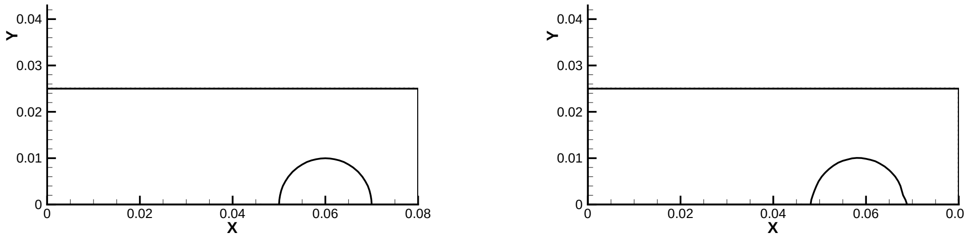


(c) t=0.04s (d) t=0.06s

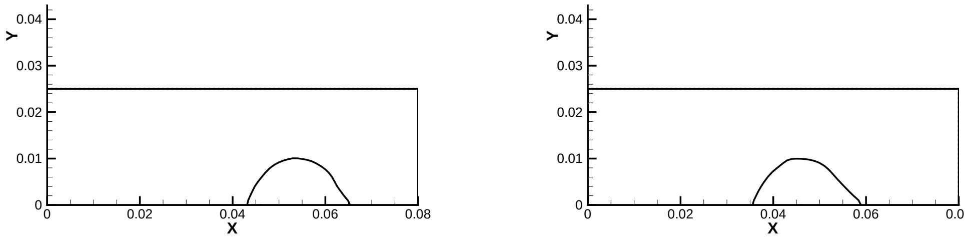


(e) t=0.08s

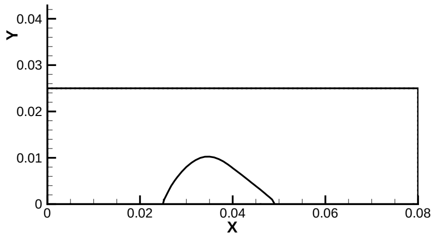
Figure 5.35: Snapshots of the 2D water drop for a grid of 320x100 and gravity vector (0.0,-9.81).



(a) t=0s (b) t=0.02s

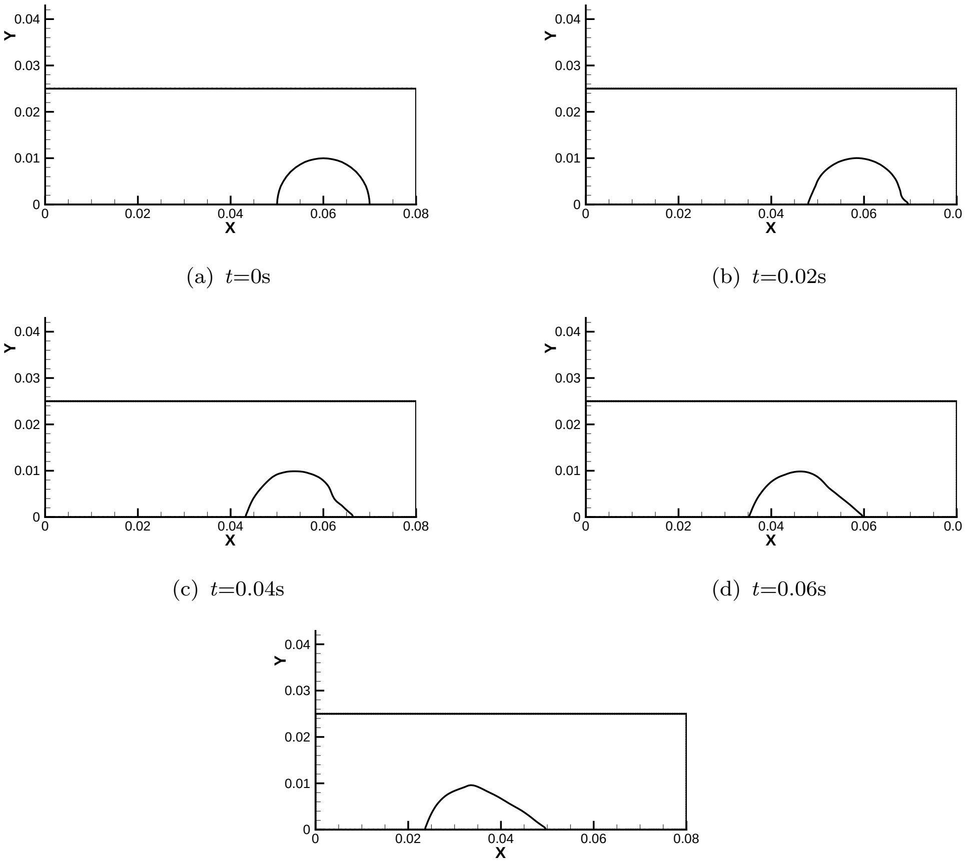


(c) t=0.04s (d) t=0.06s



(e) t=0.08s

Figure 5.36: Snapshots of the 2D bubble contour for a grid of 80x25 and gravity vector (-9.81,0).



(a) t=0s

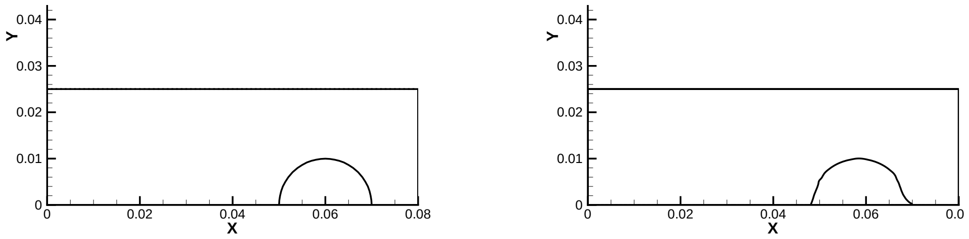
(b) t=0.02s

(c) t=0.04s

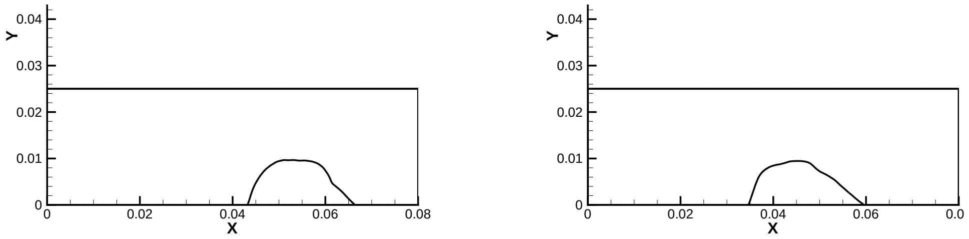
(d) t=0.06s

(e) t=0.08s

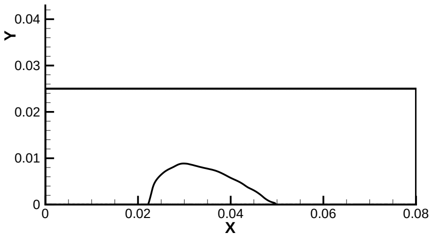
Figure 5.37: Snapshots of the 2D bubble contour for a grid of 160x50 and gravity vector (-9.81,0).



(a) t=0s (b) t=0.02s

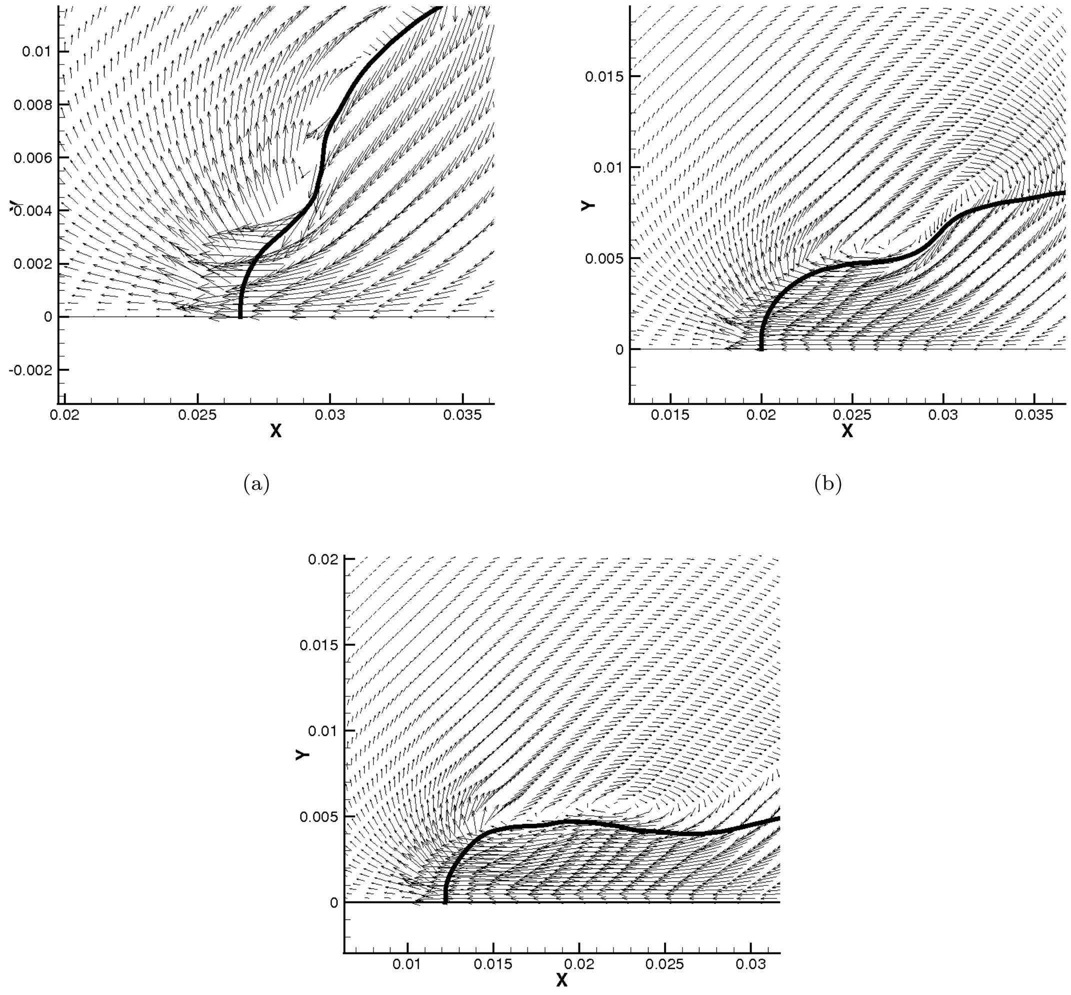


(c) t=0.04s (d) t=0.06s



(e) t=0.08s

Figure 5.38: Snapshots of the 2D bubble contour for a grid of 320x100 and gravity vector (-9.81,0).



(a)

(b)

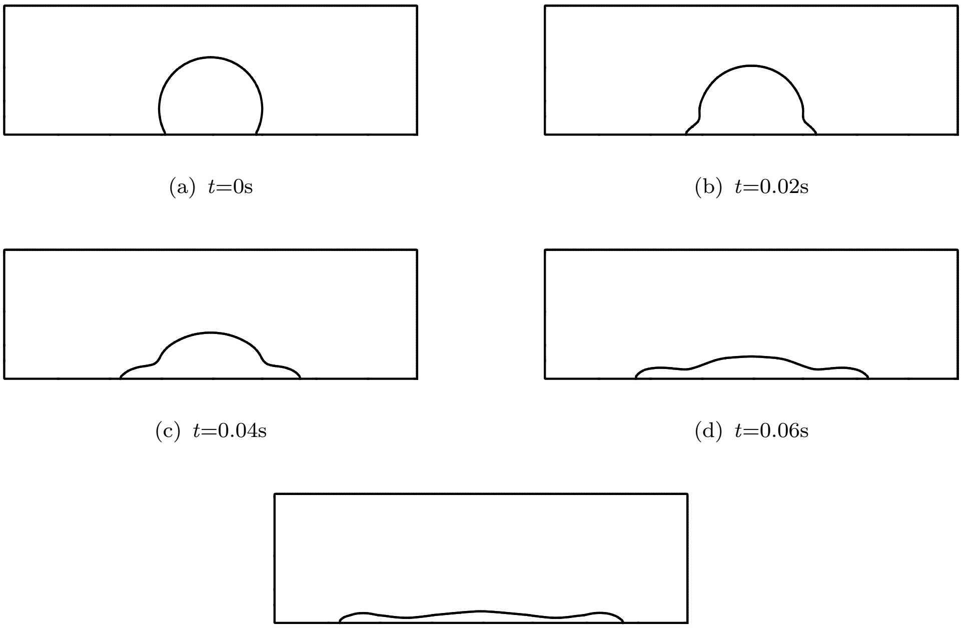
(c)

Figure 5.39: Velocity field near the contact line.

5.2.2 3D examples

In this section we extend the 2D examples shown above to the 3D case. For the first case a semi-spherical droplet of 2cm diameter is placed on a wall of a parallelipipedic domain of 8x2.5x8 cm at (4.0,0.5,4.0) cm. Based on the convergence study carried out for the 2D case, the grid chosen was 160x50x160. A no slip wall boundary condition was used for all boundaries. The dynamic contact angle, limited by 65° and 105°, was computed and prescribed at the bottom boundary. Fig. (5.40) shows the bubble contour in the XY plane for Z=0.04m at different times. Again good agreement was found when compared with numerical results of Manservisi and Scardovelli [119], and when compared with experimental data of Lim *et al.* [132] and Grundke *et al.* [118].

As for the 2D case, the gravity vector was then changed from (0.0,-9.81,0.0) to (-9.81,0.0,0.0). A semi-spherical droplet of 2cm diameter was placed on a wall of a parallelipipedic domain of 8x2.5x8 cm at (4.0,0.0,4.0) cm using a grid of 160x50x160. A no slip wall boundary condition was used for all boundaries. The dynamic contact angle, limited by 65° and 105°, was prescribed at the bottom boundary. Again different contact angles at the droplet extremities were expected. Fig. (5.41) shows how the shape and position of the droplet changed with time.



(a) t=0s

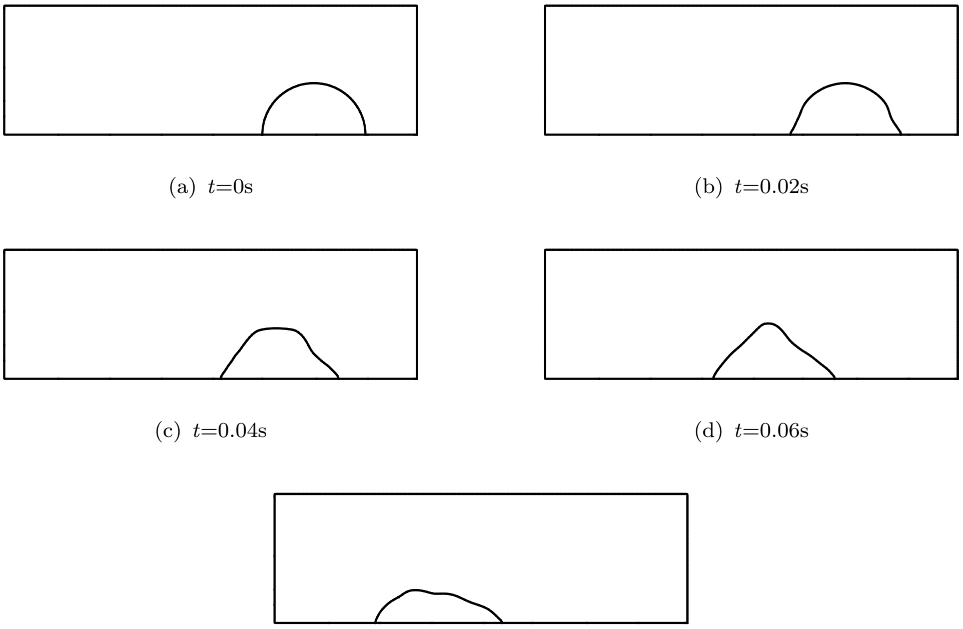
(b) t=0.02s

(c) t=0.04s

(d) t=0.06s

(e) t=0.08s

Figure 5.40: Snapshots of the 3D bubble contour for a grid of 160x50x160 and gravity vector (0,-9.81,0.0) in plane z=0.04m.



(a) t=0s

(b) t=0.02s

(c) t=0.04s

(d) t=0.06s

(e) t=0.08s

Figure 5.41: Snapshots of the 3D bubble contour for a grid of 160x50x160 and gravity vector (-9.81,0.0,0.0) in plane z=0.04m.

5.3 Heat transfer simulations

In this section, simulations of boiling inside microchannels and small geometries will be presented using water and R134a as working fluids. A 3D computational domain of 800x200x200 μm was used with a grid of 320x80x80. The physical properties are given in Table (5.15).

Table 5.15: Physical properties of water and R134a

Fluid	ρ_l kg/m^3	ρ_v kg/m^3	μ_l μPas	μ_v μPas	c_l J/kgK	c_v J/kgK	k_l W/mK	k_v W/mK	$sigma$ N/m
R134a	1187	37.54	185.4	12.04	1446	1065	0.079	0.0173	0.0074
Water	958.4	0.5976	281.8	12.27	4216	2080	0.679	0.0251	0.0589

A small spherical bubble of 20μm was placed at (0.0002, 0.00002, 0.0001). Water at 102°C enters the domain at the left side (see Fig. (5.42)) with a uniform velocity of 0.146 m/s. The initial temperature of the vapor phase is 100°C. All the physical properties are taken at 100°C. A wall superheat by 8K is applied to the bottom wall and a static contact angle of 54° was assumed. A pressure outlet boundary condition was used at the right side of the domain and the other three walls were considered adiabatic. Fig. (5.43) shows the bubble contour at different time slots as it grows on the superheated liquid while attached to the superheated wall. Although gravity is present, due to the contact angle and surface tension force,the bubble remains on the heated wall, grows and eventually occupies the entire cross section of the channel (see Fig. (5.44)) and creates dry patches on the lateral walls.

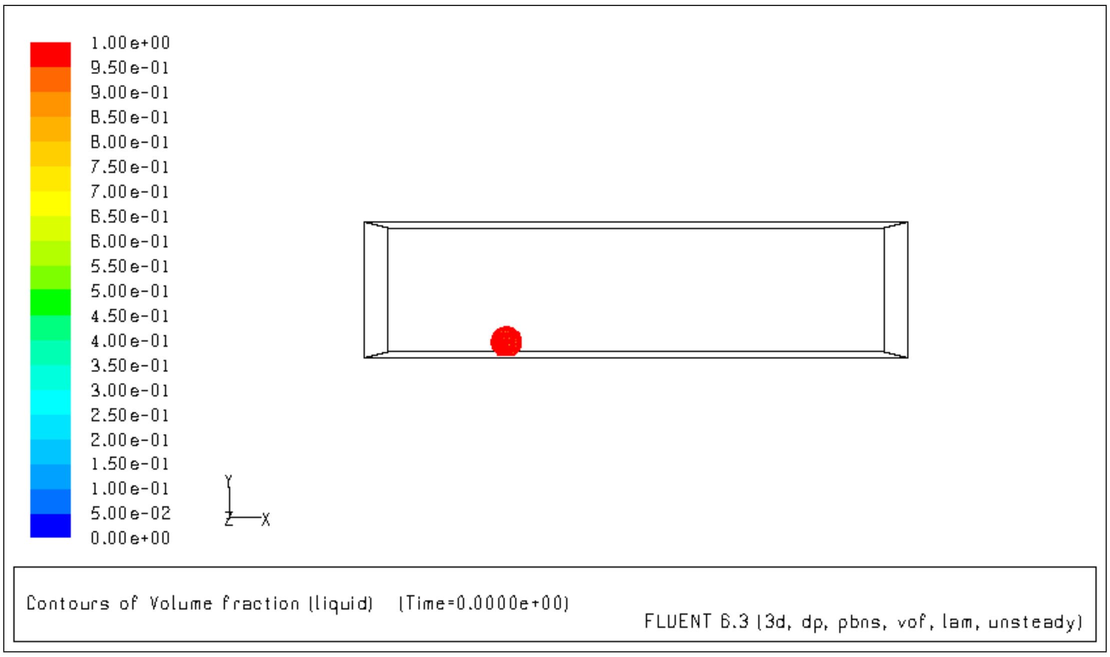


Figure 5.42: Initial condition for boiling inside microchannels.

The same computational domain was used for refrigerant R134a. The liquid enters the domain

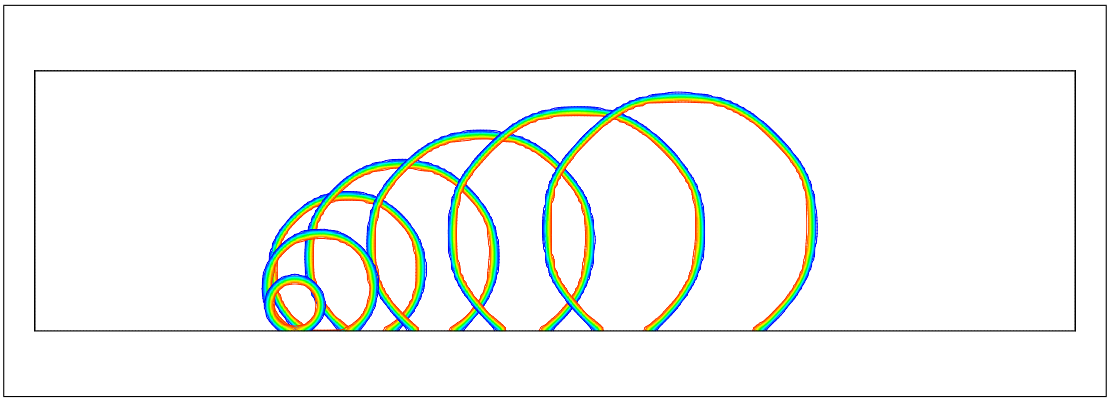


Figure 5.43: Bubble contour for boiling of water in microchannels.

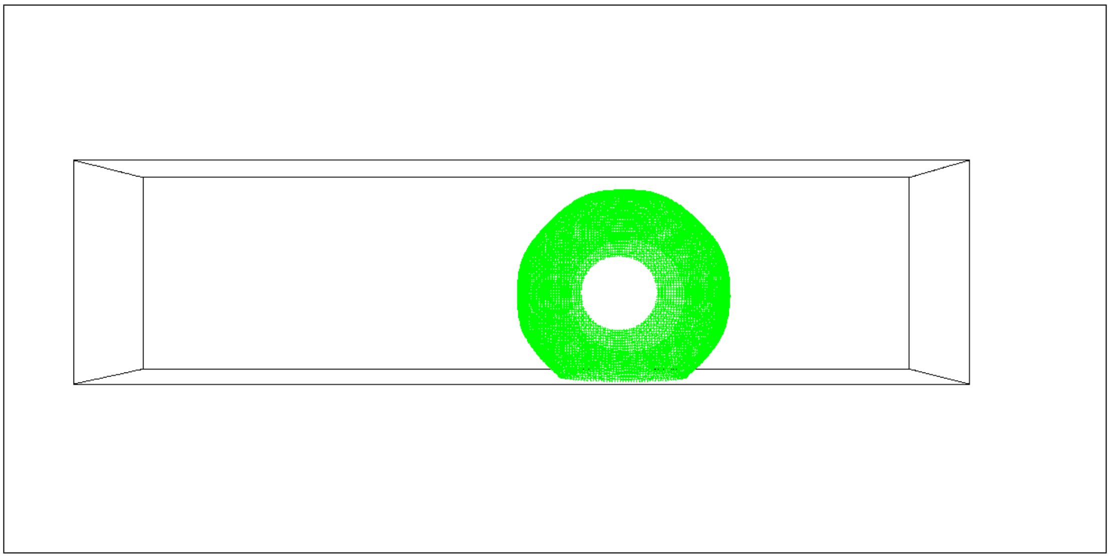


Figure 5.44: 3D bubble contour with dry patches at the lateral walls.

at 32°C at the left side with a uniform velocity of 0.05 m/s. The initial vapor phase temperature is 30°C. All the physical properties are taken at this temperature. A wall superheat by 8K is applied to the bottom wall and a static contact angle of 8° was used. A pressure outlet boundary condition was used at the right side of the domain and the other three walls were considered adiabatic. Fig. (5.45) shows the bubble contour at different time slots. Due to the low contact angle, the bubble detaches from the heated wall and therefore the heat and mass transfer process is lower than in the case with water, resulting in a smaller bubble size at the end of the simulation.

A bubble rising from a heated wall was used as an another example. A 2D axial symmetric domain of 5x1.25mm was used with a grid of 288x72. A small circular bubble of water at 100°C of 25 μm radius was placed in the domain at (0.0,0.0). The bottom wall was superheated 10K above the saturation temperature, and the surrounding liquid was set initially to 102°C. Since the contact angle is an important parameter in these types of simulations, three contact angles were used: 40°, 54° (the real value) and 80°. Fig. (5.46) depicts the bubble contour at two

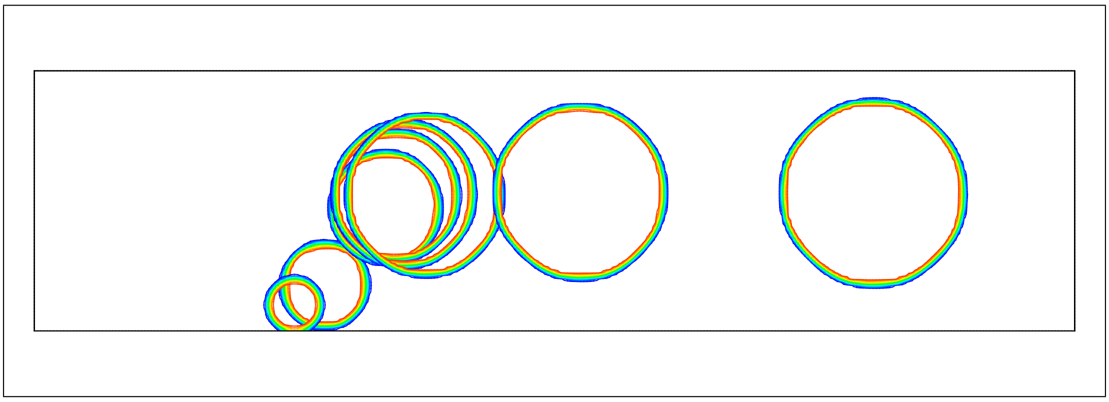


Figure 5.45: Bubble contour for boiling of R134a in microchannels

different time slots when using a contact angle of 40° where due to the low contact angle, the bubble detaches and the growing rate decreases because the bubble is not anymore in contact with the superheated wall. Fig. (5.47) shows the bubble contour when using the real (measured) contact angle of 54°. In this case the bubble remains in contact with the heated wall, and thus the heat and mass transfer are high and the bubble reaches the largest size out of the three simulations. Fig. (5.48) depicts the bubble contour for the contact angle of 80°. The highest heat and mass transfer occur near the triple line point. In this case the interface between liquid and vapor is almost perpendicular on the heated wall and so the heat and mass transfer are lower than the case of 54°, yielding a smaller bubble than the previous case.

The influence of the wall superheat was also studied. Fig. (5.49) shows the bubble contour when the bottom wall temperature is 15K above the saturation temperature and Fig. (5.50) shows the bubble contour when the bottom wall temperature is 20K above the saturation temperature. As expected, when the wall temperature is increased, the bubble grows faster and bigger.

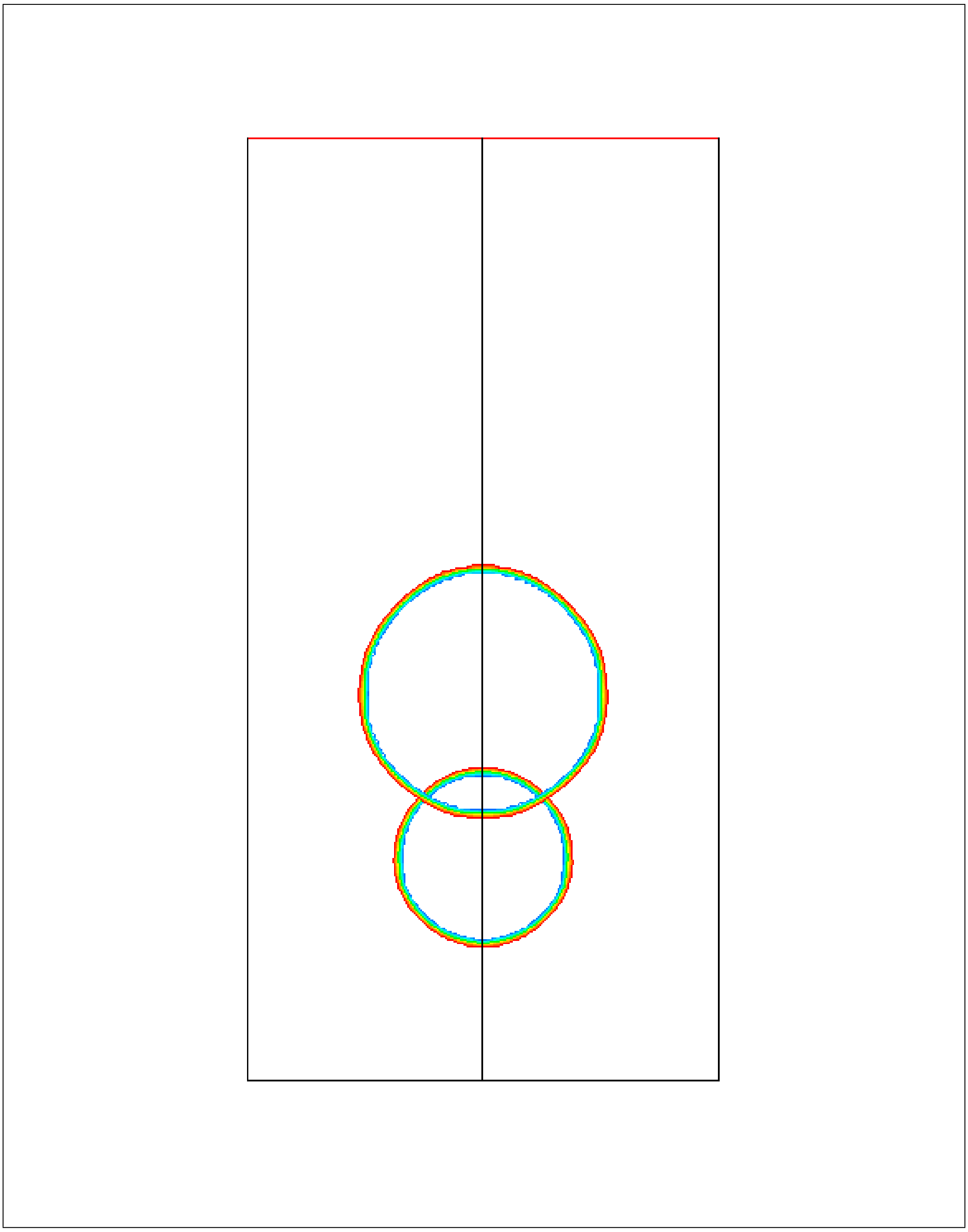


Figure 5.46: Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 40°.

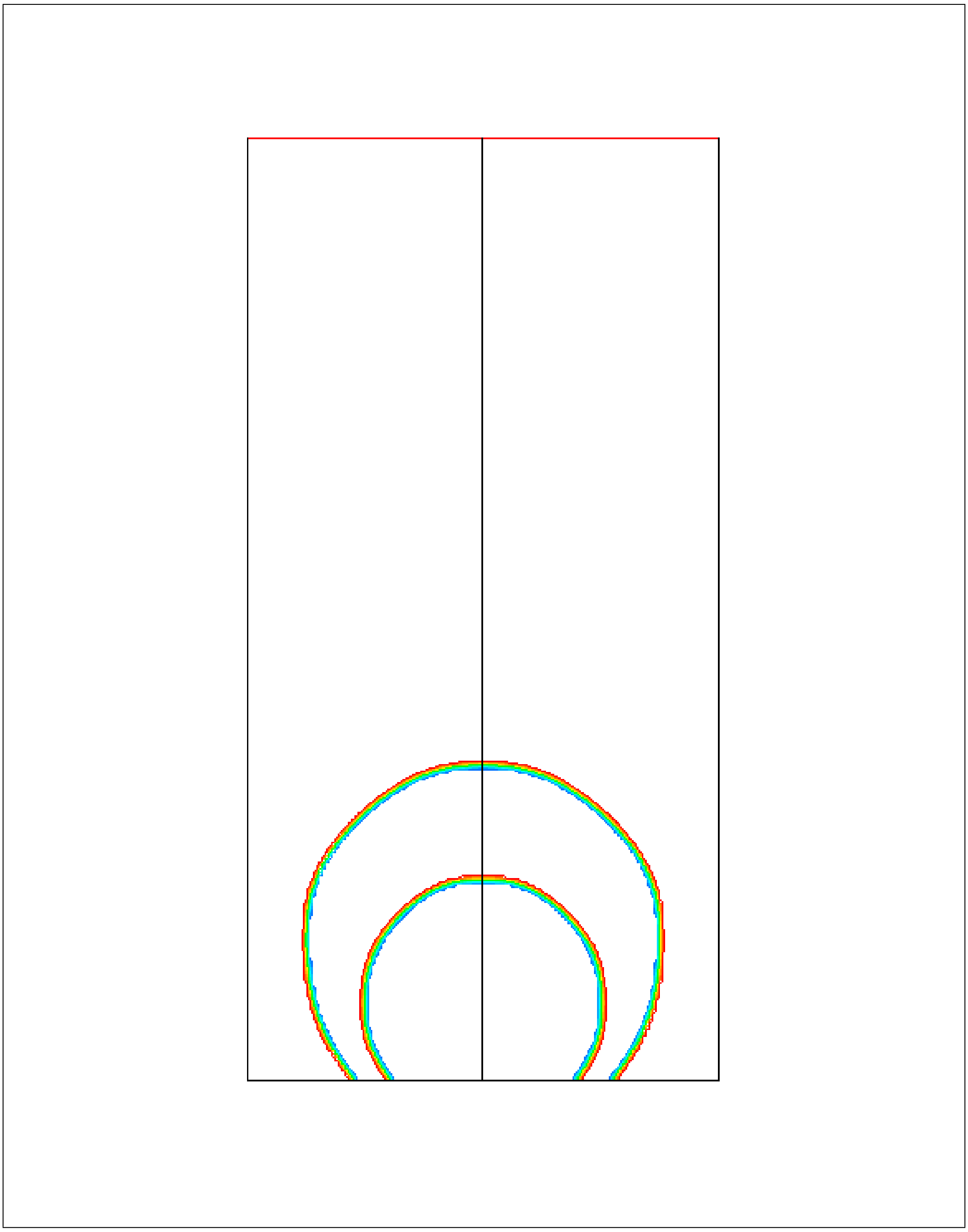


Figure 5.47: Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 54°.

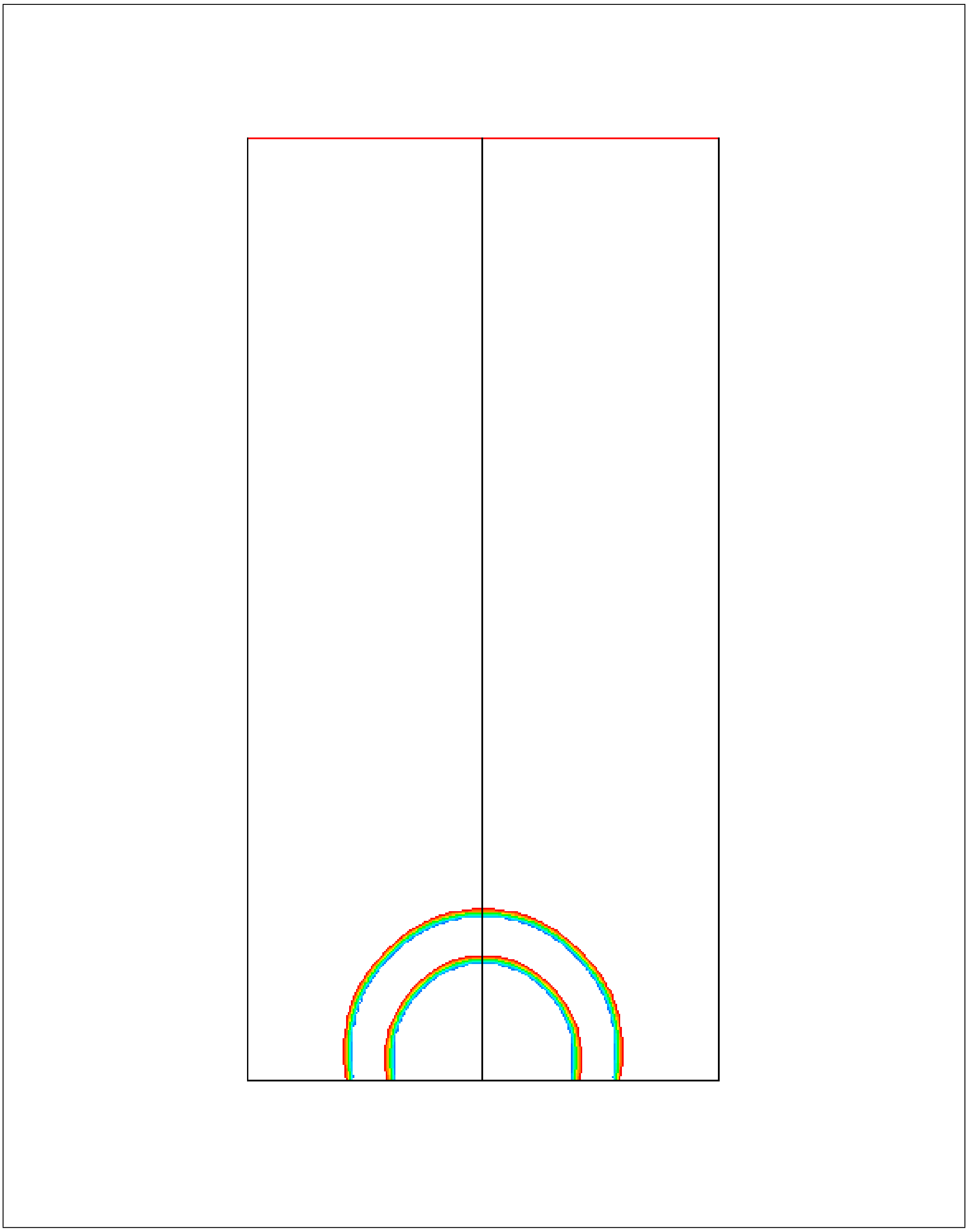


Figure 5.48: Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with a contact angle of 80°.

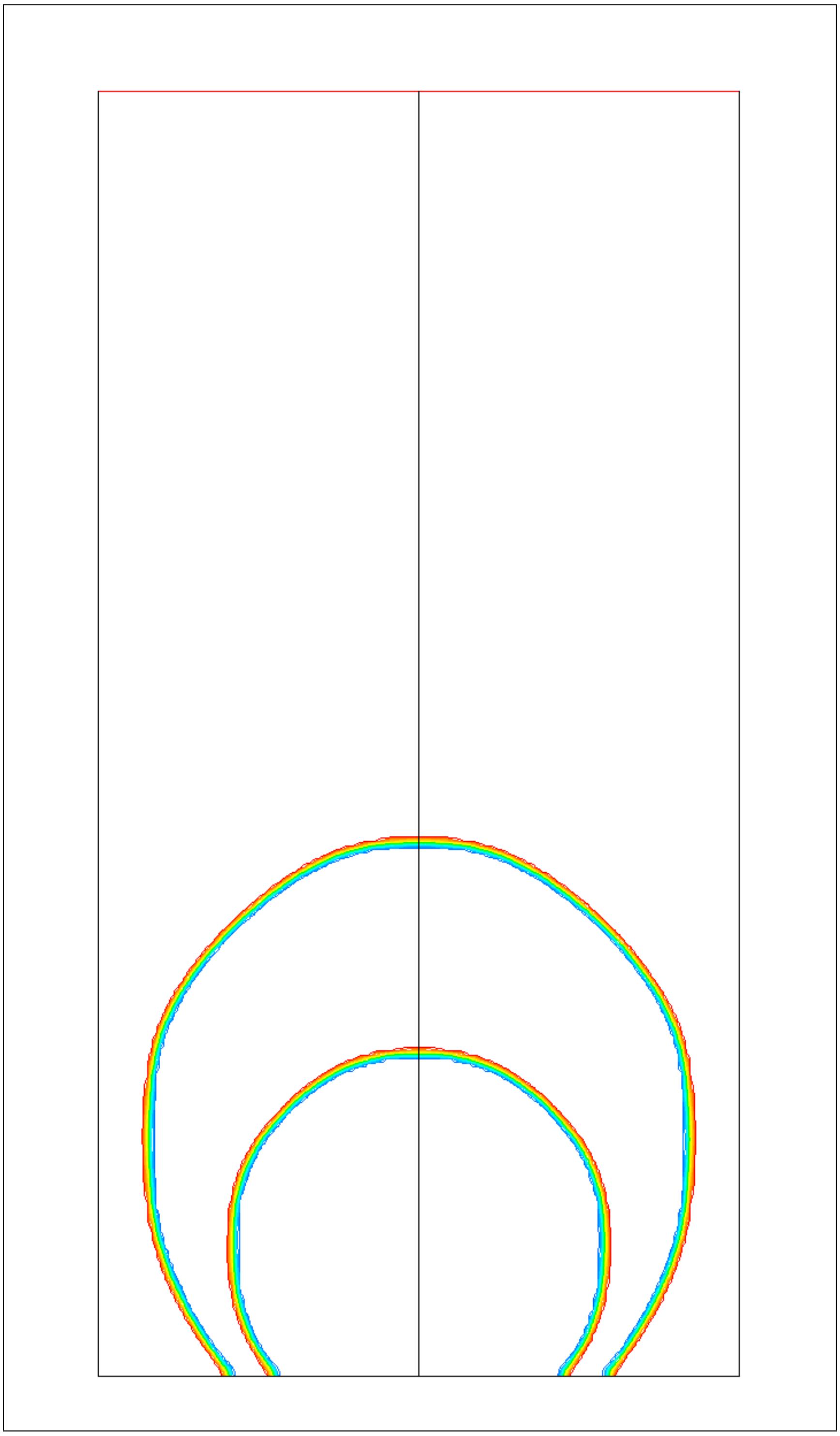


Figure 5.49: Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with 15K wall superheat with a contact angle of 54°.

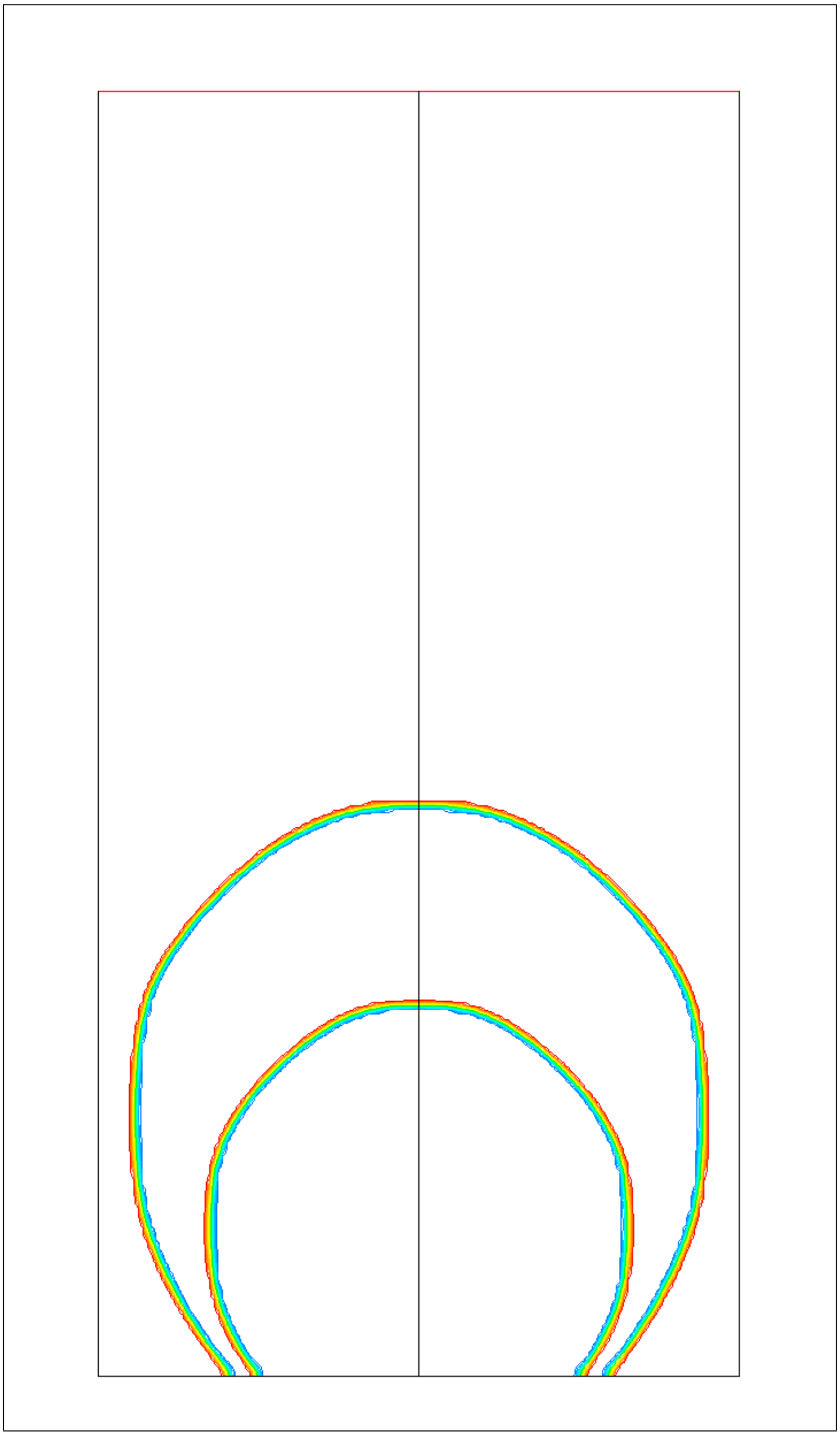


Figure 5.50: Bubble contour at $t = 0.0625s$ and $t = 0.125s$ with 20K wall superheat with a contact angle of 54°.

5.4 Film thickness simulations

The estimation of the liquid film thickness inside small channels is a key factor for the prediction of the flow boiling heat transfer. In this section, the results obtained with elongated air bubbles flowing inside a 1mm diameter tube filled with ethanol are presented, and compared with several correlations found in the literature. In one of these, Han and Shikazono [133] performed a series of film thickness measurements inside circular channels with tube diameters ranged from 0.3 to 1.3 mm with air, ethanol, water and FC-40 as working fluids. In order to have a similar set-up as in the experiments, three air bubbles were used in the present simulation. A 2D domain of 30x1mm was used with a mesh of 3000x100. The implemented CLSVOF method was used. At the inlet a velocity boundary condition was used with a developed velocity profile of 1m/s average velocity. At the outlet a pressure boundary condition was used. A no slip wall boundary condition was used for the other two boundaries. Table (5.16) shows the results of the film thickness as for the middle air bubble considering three initial film thickness: 40, 60 and 80 μm (cases I, II and III) showing some effect on the final value at the middle of the bubble, but no effect on the thinnest value.

Table 5.16: A comparison between film thickness predicted by correlations and film thickness obtained by simulations for air bubbles flowing in ethanol.

Correlation/Simulation	δ_{in} $[\mu m]$[a]	δ_0 $[\mu m]$[b]	δ_{min} $[\mu m]$[c]
Moriyama and Inoue	-	69	-
Park and Homsy	-	105	-
Aussillous and Quere	-	70	-
Han and Shikazono	-	64	-
Case I	80	135	65
Case II	60	125	65
Case III	40	115	65

---

[a] the initial film thickness assumed in our simulations

[b] film thickness given by correlation or obtained at the bubble center

[c] minimum film thickness obtained by simulations

In the first case, Moriyama and Inoue [134] measured the thickness of the liquid film formed by a growing flattened bubble in a narrow gap whose width ranged from 0.1 to 0.4 mm. They proposed the following correlation to predict the film thickness:

$$\frac{\delta}{d} = \begin{cases} 0.10(\delta^*)^{0.84}, & (Bo > 2) \\ 0.07Ca^{0.41}, & (Bo \leq 2) \end{cases} \tag{5.6}$$

where in Eq. (5.6) δ is the film thickness, d is the tube diameter, Ca is the capillary number ($Ca = \mu U/\sigma$), Bo is the Bond number ($Bo = \rho d^2 U/\sigma$) and δ^* is the ratio between the viscous layer and tube diameter.

$$\delta^* = \frac{\sqrt{\nu t_g}}{d}$$

where t_g is the time required for the bubble edge to reach the measuring point from the bubble initiation.

Han and Shikazono [133] proposed the following correlation for the prediction of the liquid film thickness:

$$\frac{\delta_0}{d} = \begin{cases} \dfrac{0.67Ca^{\frac{2}{3}}}{1+3.13Ca^{\frac{2}{3}}+0.504Ca^{0.672}Re^{0.589}-0.352We^{0.629}} & (Re < 2000) \\ \\ \dfrac{106.0\left(\frac{\mu^2}{\rho\sigma}\frac{1}{d}\right)^{\frac{2}{3}}}{1+497.0\left(\frac{\mu^2}{\rho\sigma}\frac{1}{d}\right)^{\frac{2}{3}}+7330\left(\frac{\mu^2}{\rho\sigma}\frac{1}{d}\right)^{0.672}-5000\left(\frac{\mu^2}{\rho\sigma}\frac{1}{d}\right)^{0.629}} & (Re > 2000) \end{cases} \tag{5.7}$$

where in Eq. (5.7) Ca is the capillary number, Re is the Reynolds number ($Re = \rho U d/\mu$) and We is the Weber number ($We = \rho U^2 d/\sigma$).

Park and Homsy [135] developed a theory describing two-phase displacement in the gap between closely spaced planes. They proposed the following correlation for the prediction of the liquid film thickness:

$$\frac{\delta}{d} = 0.669Ca^{2/3} \tag{5.8}$$

Aussilous and Quéré [136] performed a series of experiments at high speeds with liquids of low viscosity in order to determine the liquid film thickness. They proposed the following correlation for the prediction of the liquid film thickness:

$$\frac{\delta_0}{d} = \frac{0.67Ca^{\frac{2}{3}}}{1+2.5\cdot 1.34Ca^{\frac{2}{3}}} \tag{5.9}$$

Table (5.16) shows the prediction of the liquid film thickness using the above correlations. As can be seen, the experimentally based correlations propose thicknesses ranging from $64 - 105\mu m$, which fall between the minimum and middle values from the simulation. It is not clear experimentally which values were measured and predicted by their methods. Furthermore, it is expected that a 3D simulation will certainly improve the results (the assumption that the bubble cannot move in the perpendicular plane is eliminated), but nowadays for this case and the mesh used, a 3D simulation is quasi-impossible; the 2D simulation took about 60 days to finish on a 4 processor computer node. Fig. (5.51) depicts the bubble contours in a steady-state condition and Fig. (5.52) shows the presence of a small recirculating zone near the rear of the middle air bubble. Fig. (5.53) depicts the film thickness (top and bottom) along the middle air bubble. From Fig. (5.53) we can say that the buoyancy effect is minimal.

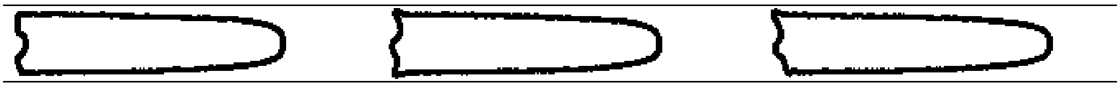


Figure 5.51: Bubble contour for the air bubble flowing inside 1mm tube filled with ethanol.

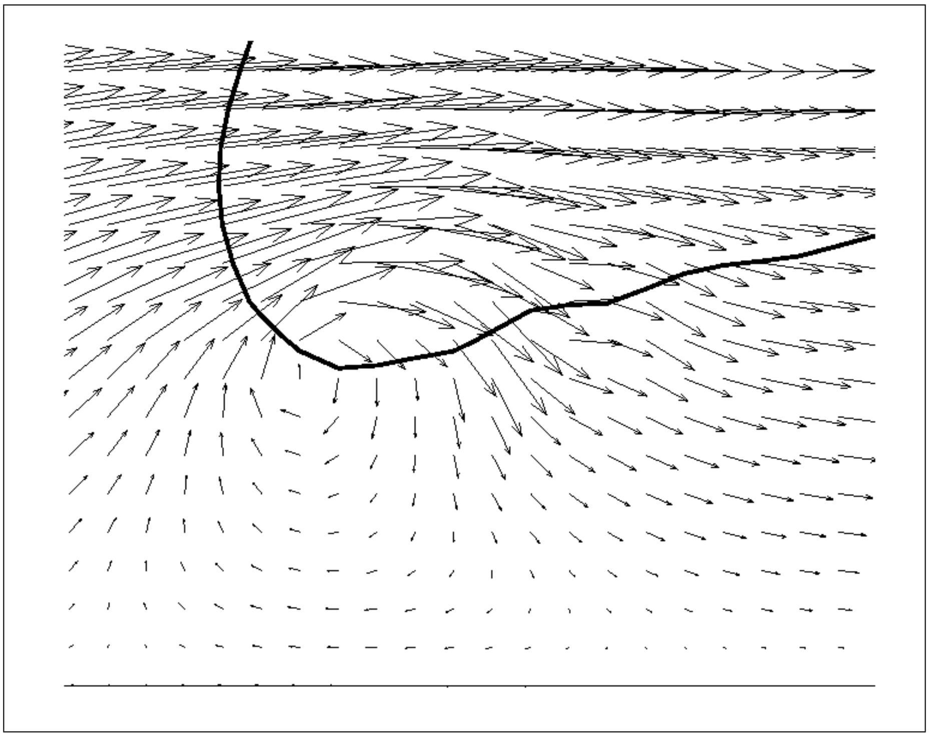


Figure 5.52: Velocity field near the rear of the middle air bubble.

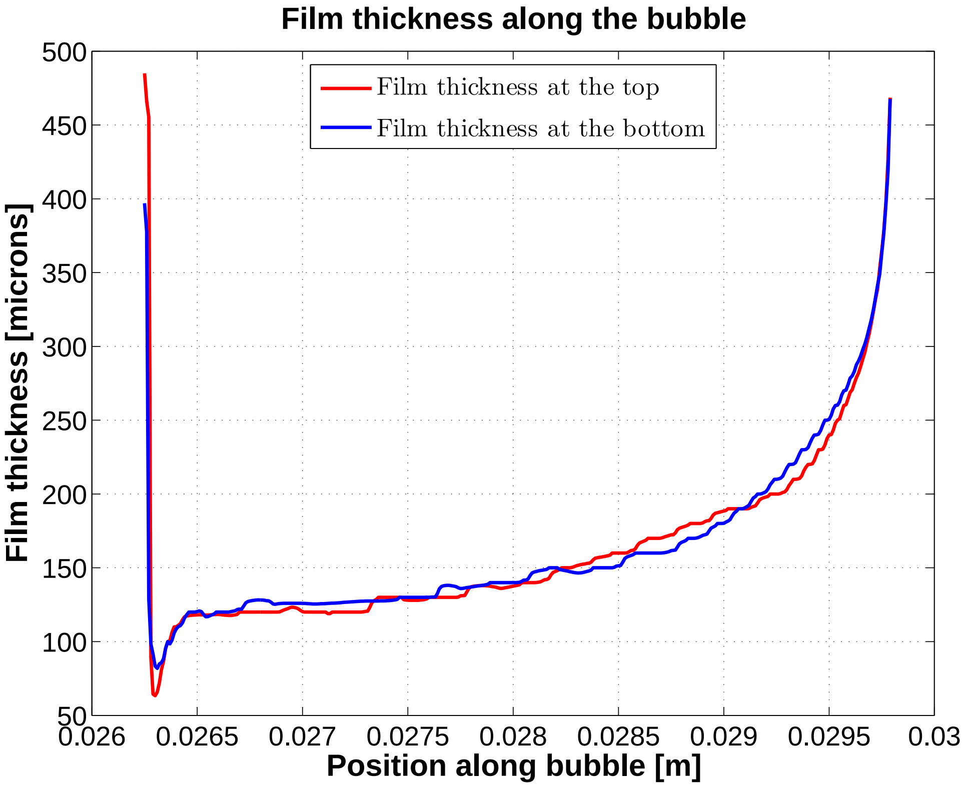


Figure 5.53: Film thickness along the middle air bubble.

Chapter 6

Conclusions

A fully 3D level set method was developed and coupled with the volume of fluid method within the commercial CFD code FLUENT. In this CLSVOF method, the level set function was used to compute the surface tension contribution to the Navier-Stokes equation more accurately than the VOF method by itself. The volume of fluid function was then used to capture the interface. By doing this, we overcame two drawbacks of LS and VOF methods: the mass conservation problem of the LS method, and the rather poor calculation of curvature and normal vector to the interface for the VOF method.

A re-initialization equation was solved after each time step for the level set function. This equation was discretized using a fifth order WENO scheme for spatial derivatives and a first order Euler explicit method for time integration. The method was implemented on both serial and parallel solvers.

The static bubble simulation we performed showed a reduction of the strength of the spurious currents around the bubble by approximately 51% using CLSVOF compared with PLIC VOF. This is explained by the fact that the level set function is a continuous function as opposed to the volume of fluid function, and thus the surface tension force is discretized more accurately.

We also simulated droplet deformation due to a vortex velocity field. The interface position, after undergoing severe deformation, was recovered at the initial position with a very small perturbation for the coarsest grid and insignificantly for the finest grid.

Our numerical validation was concluded for a series of bubbles rising, and sometimes coalescing, in a stagnant fluid. We achieved good agreement with the experimental data available from Bhaga and Weber [9], with a maximum relative error of approx 16% using CLSVOF; while with VOF the maximum relative error was approx 19%. Using a 3D-axisymmetric domain we achieved a better accuracy with a maximum relative error of approx 9%. A very good agreement was also obtained when we compared the results obtained with our CLSVOF code with the experimental data of Hnat and Buckmaster [10]. Although for the air-water sugar simulations the difference between CLSVOF and VOF was almost insignificant, for the air-water simulations CLSVOF performed better than VOF itself. Furthermore, vortex shedding was better observed with CLSVOF than with VOF. Although the relative error for the bubble mean lateral displacement was 55% using CLSVOF compared with the experiments of Zun and Groselj [5], CLSVOF (bubble mean lateral displacement of 0.885mm) generally, performs better than VOF (bubble mean lateral displacement of 1.7mm is attributed based on non-realistic bubble movement that does not comprise periodic nature at all). The bubble rise velocity obtained with CLSVOF and VOF was comparable with the experimental one, with slightly better accuracy achieved with the CLSVOF method.

Using the CLSVOF method the CPU time required is a minimum of two times higher than the corresponding CPU time required by the VOF method. The CPU time required by CLSVOF is highly influenced by the number of mesh points, parallel or serial solvers, number of processors etc. One can think that instead using the CLSVOF method to use the VOF method with a higher mesh resolution. For a 3D case, by doubling the mesh, the CPU time required is 64 times higher than the case with the original mesh which is considerably higher than using CLSVOF with the original mesh.

The commercial CFD code FLUENT allows one to use a static contact angle between the interface and the wall. In dynamic conditions (i.e. when the contact line starts to move), this choice of a static value is not anymore appropriate.

A dynamic contact angle model was developed and implemented via UDF into a commercial CFD code FLUENT for two phase flows with wetted boundaries. As a first step, the interface is reconstructed based on the volume fractions. After the interface position is determined, the contact angle is determined based on the volume fractions and the normal to the interface. The model has as its lower limit for the contact angle, the receding contact angle, and as its upper limit, the advancing contact angle. The advancing and receding contact angles are available experimentally for several pairs of fluids/surfaces, or can be estimated using several published correlations.

Several 2D and 3D test were performed for air/water on a coated silicon wafer surface with different gravity vectors, which proved the accuracy of our model when compared to both numerically and experimentally data available in the literature.

A heat and mass transfer model was also implemented into the commercial CFD code FLUENT for simulating of boiling (and condesation) heat transfer. Several 3D and 2D axis symmetric simulations were presented with water and R134a as working fluids. The influence of the contact angle and the wall superheat was also studied.

6.1 Future work

The LS method, the dynamic contact angle model and the heat transfer model with phase change presented here work for a Cartesian grid, which is inappropriate for complex geometries like t-junctions. Therefore a general curvilinear implementation of the LS method and the dynamic contact angle model is foreseen.

A more efficient parallel implementation of the LS method could also be done.

The development of the heat transfer model is needed by inclusion of the microlayer region underneath the bubble.

Appendix A

The WENO scheme

For ENO schemes, one should use three substencils as is shown in Fig. (A.1). In this case, the third order ENO scheme will choose one from the following possibilities:

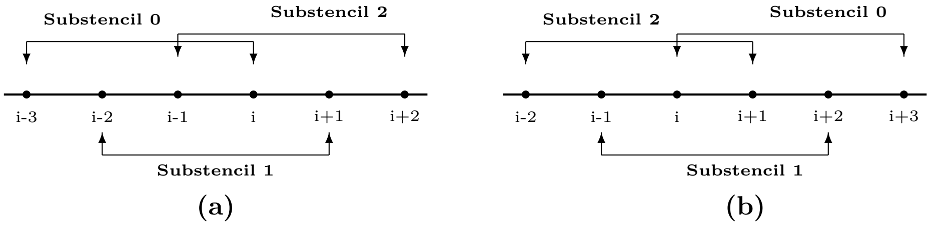


Figure A.1: The three substencils: (a) the left-biased stencil; (b) the right-biased stencil.

$$\begin{aligned}
\phi_{x,i}^{-,0} &= \frac{1}{3}\frac{\phi_{i-3}^{+}}{\Delta x} - \frac{7}{6}\frac{\phi_{i-2}^{+}}{\Delta x} + \frac{11}{6}\frac{\phi_{i-1}^{+}}{\Delta x} \\
\phi_{x,i}^{-,1} &= -\frac{1}{6}\frac{\phi_{i-2}^{+}}{\Delta x} + \frac{5}{6}\frac{\phi_{i-1}^{+}}{\Delta x} + \frac{1}{3}\frac{\phi_{i}^{+}}{\Delta x} \\
\phi_{x,i}^{-,2} &= \frac{1}{3}\frac{\phi_{i-1}^{+}}{\Delta x} + \frac{5}{6}\frac{\phi_{i}^{+}}{\Delta x} - \frac{1}{6}\frac{\phi_{i+1}^{+}}{\Delta x}
\end{aligned} \tag{A.1}$$

The best term of $\phi_{x,i}^{-,s}$ with $s = 0, 1, 2$ to choose is decided as follows:

$$\phi_{x,i}^{-} = \begin{cases} \phi_{x,i}^{-,0} & \text{if} \quad |\Delta^-\Delta^+\phi_{i-1}| < |\Delta^-\Delta^+\phi_i| \text{ and} \\ & \qquad |\Delta^-\Delta^-\Delta^+\phi_{i-1}| < |\Delta^+\Delta^-\Delta^+\phi_{i-1}|; \\ \phi_{x,i}^{-,2} & \text{if} \quad |\Delta^-\Delta^+\phi_{i-1}| > |\Delta^-\Delta^+\phi_i| \text{ and} \\ & \qquad |\Delta^-\Delta^-\Delta^+\phi_{i-1}| > |\Delta^+\Delta^-\Delta^+\phi_{i-1}|; \\ \phi_{x,i}^{-,1} & \text{if otherwise} \end{cases} \tag{A.2}$$

The WENO approximation of $\phi_x(x_i)$ is a combination of the weighted average of $\phi_{x,i}^{-,s}$ with $s = 0, 1, 2$:

$$\phi_{x,i}^{-} = \omega_0\phi_{x,i}^{-,0} + \omega_1\phi_{x,i}^{-,1} + \omega_2\phi_{x,i}^{-,2} \tag{A.3}$$

Here we know that $\omega_0 + \omega_1 + \omega_2 = 1$. Usually, these weights have the following values

$$\omega_0 = 0.1,\ \omega_1 = 0.6,\ \omega_2 = 0.3$$

and with definitions from Eq. (Eq. (A.1)), we get:

$$\phi_{x,i}^{-} = \frac{1}{30}\frac{\phi_{i-3}^{+}}{\Delta x} - \frac{13}{60}\frac{\phi_{i-2}^{+}}{\Delta x} + \frac{47}{60}\frac{\phi_{i-1}^{+}}{\Delta x} + \frac{9}{20}\frac{\phi_{i}^{+}}{\Delta x} - \frac{1}{20}\frac{\phi_{i+1}^{+}}{\Delta x} \tag{A.4}$$

Now substituting $\omega_1 = 1 - \omega_0 - \omega_2$ into Eq. (A.3) we get:

$$\phi_{x,i}^{-} = \frac{1}{2}(\phi_{x,i}^{-,1} + \phi_{x,i}^{-,2}) + \omega_0(\phi_{x,i}^{-,0} - \phi_{x,i}^{-,1}) + \left(\omega_2 - \frac{1}{2}\right)(\phi_{x,i}^{-,2} - \phi_{x,i}^{-,1}) \tag{A.5}$$

Now, substituting in the definitions given in Eq. (Eq. (A.1)) we obtain:

$$\begin{aligned} \phi_{x,i}^{-} &= \frac{1}{12}\left(-\frac{\phi_{i-2}^{+}}{\Delta x} + 7\frac{\phi_{i-1}^{+}}{\Delta x} + 7\frac{\phi_{i}^{+}}{\Delta x} - \frac{\phi_{i+1}^{+}}{\Delta x}\right) \\ &\quad - \Phi^{WENO}\left(\frac{\Delta^-\Delta^+\phi_{i-2}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i-1}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i+1}}{\Delta x}\right) \end{aligned} \tag{A.6}$$

where

$$\Phi^{WENO}(a,b,c,d) = \frac{1}{3}\omega_0(a - 2b + c) + \frac{1}{6}\left(\omega_2 - \frac{1}{2}\right)(b - 2c + d) \tag{A.7}$$

and the weights $\omega_0,\ \omega_2$ are defined as:

$$\omega_0 = \frac{\alpha_0}{\alpha_0 + \alpha_1 + \alpha_2}, \qquad \omega_2 = \frac{\alpha 2}{\alpha_0 + \alpha_1 + \alpha_2}$$

and

$$\alpha_0 = \frac{1}{(\epsilon + IS_0)^2}, \qquad \alpha_1 = \frac{6}{(\epsilon + IS_1)^2}, \qquad \alpha_2 = \frac{3}{(\epsilon + IS_2)^2}$$

where

$$\begin{aligned} IS_0 &= 13(a-b)^2 + 3(a-3b)^2 \\ IS_1 &= 13(b-c)^2 + 3(b+c)^2 \\ IS_2 &= 13(c-d)^2 + 3(3c-d)^2 \end{aligned}$$

In order to prevent the denominator from becoming zero, we set $\epsilon = 10^{-6}$.

By symmetry, we can find the approximation of $\phi_x(x_i)$ on the right-biased stencil (see Fig. (A.1)) to be:

$$\begin{aligned} \phi_{x,i}^{+} &= \frac{1}{12}\left(-\frac{\phi_{i-2}^{+}}{\Delta x} + 7\frac{\phi_{i-1}^{+}}{\Delta x} + 7\frac{\phi_{i}^{+}}{\Delta x} - \frac{\phi_{i+1}^{+}}{\Delta x}\right) \\ &\quad + \Phi^{WENO}\left(\frac{\Delta^-\Delta^+\phi_{i+2}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i+1}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i}}{\Delta x}, \frac{\Delta^-\Delta^+\phi_{i-1}}{\Delta x}\right) \end{aligned} \tag{A.8}$$

For a general 3D case, the fifth order HJ WENO solutions are:

$$\begin{aligned}\phi^{\pm}_{x,i,j,k} &= \frac{1}{12}\left(-\frac{\phi^{+}_{x,i-2,j,k}}{\Delta x}+7\frac{\phi^{+}_{x,i-1,j,k}}{\Delta x}+7\frac{\phi^{+}_{x,i,j,k}}{\Delta x}-\frac{\phi^{+}_{x,i+1,j,k}}{\Delta x}\right)\\ &\pm \Phi^{WENO}\left(\frac{\Delta^{-}_{x}\Delta^{+}_{x}\phi_{i\pm2,j,k}}{\Delta x},\frac{\Delta^{-}_{x}\Delta^{+}_{x}\phi_{i\pm1,j,k}}{\Delta x},\frac{\Delta^{-}_{x}\Delta^{+}_{x}\phi_{i,j,k}}{\Delta x},\frac{\Delta^{-}_{x}\Delta^{+}_{x}\phi_{i\mp1,j,k}}{\Delta x}\right)\end{aligned} \tag{A.9}$$

$$\begin{aligned}\phi^{\pm}_{y,i,j,k} &= \frac{1}{12}\left(-\frac{\phi^{+}_{y,i,j-2,k}}{\Delta y}+7\frac{\phi^{+}_{y,i,j-1,k}}{\Delta y}+7\frac{\phi^{+}_{y,i,j,k}}{\Delta y}-\frac{\phi^{+}_{y,i,j+1,k}}{\Delta y}\right)\\ &\pm \Phi^{WENO}\left(\frac{\Delta^{-}_{y}\Delta^{+}_{y}\phi_{i,j\pm2,k}}{\Delta y},\frac{\Delta^{-}_{y}\Delta^{+}_{y}\phi_{i,j\pm1,k}}{\Delta y},\frac{\Delta^{-}_{y}\Delta^{+}_{y}\phi_{i,j,k}}{\Delta y},\frac{\Delta^{-}_{y}\Delta^{+}_{y}\phi_{i,j\mp1,k}}{\Delta y}\right)\end{aligned} \tag{A.10}$$

$$\begin{aligned}\phi^{\pm}_{z,i,j,k} &= \frac{1}{12}\left(-\frac{\phi^{+}_{z,i,j,k-2}}{\Delta z}+7\frac{\phi^{+}_{z,i,j,k-1}}{\Delta z}+7\frac{\phi^{+}_{z,i,j,k}}{\Delta z}-\frac{\phi^{+}_{z,i,j,k+1}}{\Delta z}\right)\\ &\pm \Phi^{WENO}\left(\frac{\Delta^{-}_{z}\Delta^{+}_{z}\phi_{i,j,k\pm2}}{\Delta z},\frac{\Delta^{-}_{z}\Delta^{+}_{z}\phi_{i,j,k\pm1}}{\Delta z},\frac{\Delta^{-}_{z}\Delta^{+}_{z}\phi_{i,j,k}}{\Delta z},\frac{\Delta^{-}_{z}\Delta^{+}_{z}\phi_{i,j,k\mp1}}{\Delta z}\right)\end{aligned} \tag{A.11}$$

Eq. (A.9), Eq. (A.10) and Eq. (A.11) are the general 3D fifth order HJ WENO approximations for the first derivative of the level set function ϕ with respect to x, y and z.

It is clear that for a point i one needs three points behind him and three points after him in order to discretize the gradient of a function using a fifth order WENO scheme. This pose no problem for a point inside the domain, but when the point is within 3 points far from the wall, the scheme doesn't work anymore due to the lack of points. The trick used was to extend the domain by three points beyond the actual domain.

Appendix B

The computation of the plane (line) constant α

B.1 Three-dimensional case

In 3D space with Cartesian coordinates (x,y,z) a rectangular parallelepiped of sides Δx, Δy and Δz is considered and a plane with normal vector $\mathbf{m} = (m_1, m_2, m_3)$ given by the equation:

$$m_1 x + m_2 y + m_3 z = \alpha \tag{B.1}$$

In the standard problem, the three coefficients m_1, m_2 and m_3 are assumed to be all positive

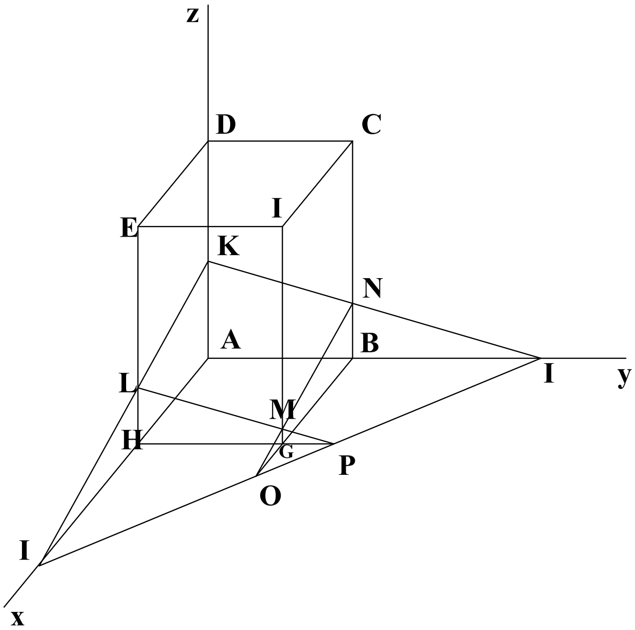


Figure B.1: The “cut volume” of a 3D rectangular cell.

and also $m_1 \leq m_2 \leq m_3$. The "cut volume" $ABCHKNML$ (see Fig. (B.1))of the rectangular cell need to be determined. This volume is given by the following relation [70]:

$$V = \frac{1}{6m_1m_2m_3}\left[\alpha^3 - \sum_{i=1}^{3} F_3\left(\alpha - m_i\Delta x_i\right) + \sum_{i=1}^{3} F_3\left(\alpha - \alpha_{\max} + m_i\Delta x_i\right)\right] \tag{B.2}$$

in Eq. (B.2) $\alpha_{\max} = \sum_{i=0}^{3} m_i\Delta x_i$ and function $F_3(y)$ is defined as:

$$F_3(y) = \begin{cases} y^3 & \text{for} \quad y > 0 \\ 0 & \text{for} \quad y \leq 0 \end{cases}$$

The volume fraction varies from zero, when $\alpha = 0$ to the volume of the parallelepiped $V_0 = \Delta x_1\Delta x_2\Delta x_3$ when $\alpha = \alpha_{\max}$. The analysis is first restricted to an unitary cube $\Delta x_i = 1$, and also the plane equation Eq. (B.1) is normalized by dividing it by $\sum_{i=1}^{3} x_i$. For more details and a thorough analysis see [125].

If we consider $m_{12} = m_1 + m_2$ and $m = \min(m_{12}, m3)$, then V varies cubically in the region $0 \leq \alpha \leq m_1$, quadratically in $m_1 \leq \alpha \leq m_2$, again cubically in $m_2 \leq \alpha \leq m$, and finally in $m \leq \alpha \leq 1/2$ cubically is $m = m_3$ and otherwise linearly if $m = m_{12}$.

For this kind of application we have the "forward" problem, which is to find the volume fraction occupied by one species given α, and the "inverse" problem, which is to find α given the volume fraction. Here we give the relations only for the "inverse" problem. A reader interested in the "forward" problem should see reference [125].

3D inverse problem:

$$\begin{array}{rcll} \alpha & = & \sqrt[3]{6m_1m_2m_3V}, & \text{for} \quad 0 \leq V < V_1 \\ \alpha & = & \frac{1}{2}(m_1 + \sqrt{m_1^2 + 8m_2m_3(V - V_1)}, & \text{for} \quad V_1 \leq V < V_2 \\ P(\alpha) & = & a_3'\alpha^3 + a_2'\alpha^2 + a_1'\alpha + a_0' = 0 & \end{array}$$

and we have two cases for the forth interval:

$$\begin{array}{rcll} P(\alpha) & = & a_3''\alpha^3 + a_2''\alpha^2 + a_1''\alpha + a_0'' = 0, & \text{for} \quad V_{31} \leq V \leq 1/2, \\ \alpha & = & m_3V + \frac{m_{12}}{2}, & \text{for} \quad V_{32} \leq V \leq 1/2. \end{array}$$

In the previous relations $V_1 = m_1^2/(\max(6m_2m_3, \epsilon))$, $V_2 = V_1 + (m_2 - m_1)/2m_3$, $V_3 = V_{31} = [m_3^2(3m_{12} - m_3) + m_1^2(m_1 - 3m_3) + m_2^2(m_2 - 3m_3)]/(6m_1m_2m_3)$ when $m = m3$ or $V_3 = V_{32} = m12/2m_3$ when $m = m_{12}$. For the coefficients of the two cubic polynomials we have $a_3' = -1$, $a_2' = 3m_{12}$, $a_1' = -3(m_1^2 + m_2^2)$, $a_0' = m_1^3 + m_2^3 - 6m_1m_2m_3V$, $a_3'' = -2$, $a_2'' = 3$, $a_1'' = -3(m_1^2 + m_2^2 + m_3^2)$ and $a_0'' = m_1^3 + m_2^3 + m_3^3 - 6m_1m_2m_3V$.

To find the roots of the cubic polynomial $P(\alpha)$, we first divide by a_3 and let

$$p_0 = \frac{a_1}{3} - \frac{a_2^2}{9}; \qquad q_0 = \frac{a_1a_2 - 3a_0}{6} - \frac{a_2^3}{27}$$

then we let $\cos(3\theta) = q_0/\sqrt{-p_0^3}$ and we find the root of interest

$$\alpha = \sqrt{-p_o}(\sqrt{3}\sin\theta - \cos\theta) - \frac{a_2}{3}.$$

B.2 Two-dimensional case

In 2D space with Cartesian coordinates (x,y) a rectangular parallelepiped of sides Δx and Δy is considered and a plane with normal vector $\mathbf{m} = (m_1, m_2)$ given by the equation:

$$m_1 x + m_2 y = \alpha \tag{B.3}$$

Again the coefficients m_1 and m_2 are assumed positive and $m_1 \leq m_2$. The "cut volume" *ABFGD* (see Fig. (B.2))is given by:

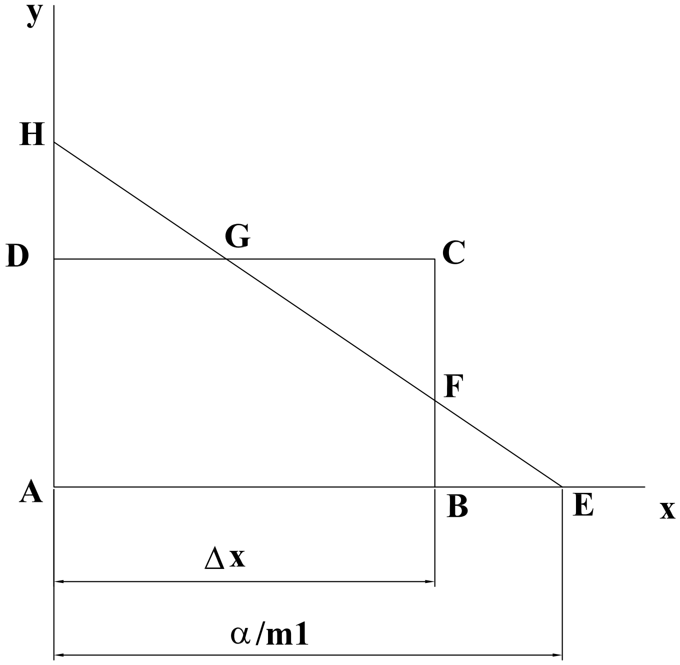


Figure B.2: The "cut volume" of a 2D rectangular cell.

$$V = \frac{1}{2m_1m_2}\left[\alpha^2 - \sum_{i=1}^{2} F_2(\alpha - m_i\Delta x_i)\right] \tag{B.4}$$

The analysis is first restricted to an unitary cube $\Delta x_i = 1$, and also the plane equation Eq. (B.3) is normalized by dividing it by $\sum_{i=1}^{2} x_i$. We let $m = m_1$.

2D inverse problem:

$$\begin{aligned} \alpha &= \sqrt{2m(1-m)V} \qquad \text{for} \quad 0 \le V < V_1, \\ \alpha &= V(1-m) + \tfrac{m}{2} \qquad \text{for} \quad V_1 \le V \le 1/2. \end{aligned}$$

with $V_1 = m/2(1-m)$.

If one or more of the m_i are negative, the geometry can be brought to the standard case with the linear transformation $x_i^{'} = \Delta x_i - x_i$.

Appendix C

Pressure-velocity coupling algorithms

C.1 The SIMPLE algorithm

The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) was originally developed by Patankar and Spalding [137] and is a guess and correct procedure for the calculation of pressure on the staggered grid arrangement. First a pressure field p^* is guessed. Discretized momentum equations are solved using the guessed pressure field to yields velocity components u^* and v^* as follows:

$$a_{i,J}u^*_{i,J} = \sum a_{nb}u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})A_{i,J} + b_{i,J} \tag{C.1}$$

$$a_{i,J}v^*_{i,J} = \sum a_{nb}v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J})A_{i,J} + b_{I,j} \tag{C.2}$$

A correction p' is then defined so that

$$p = p^* + p' \tag{C.3}$$

Similarly we define velocity corrections u' and v' to correlate the correct velocities u and v to the guessed velocities u^* and v^*:

$$u = u' + u^* \tag{C.4}$$

$$v = v' + v^* \tag{C.5}$$

Taking into account Eq. (C.3), Eq. (C.4) and Eq. (C.5) we can write now:

$$a_{i,J}u'_{i,J} = \sum a_{nb}u'_{nb} + (p'_{I-1,J} - p'_{I,J})A_{i,J} \tag{C.6}$$

$$a_{i,J}v'_{i,J} = \sum a_{nb}v'_{nb} + (p'_{I,J-1} - p'_{I,J})A_{I,j} \tag{C.7}$$

The SIMPLE algorithm then ignore the terms $\sum a_{nb}u'_{nb}$ and $\sum a_{nb}v'_{nb}$. We obtain then:

$$u_{i,J} = u^*_{i,J} + \frac{A_{i,J}}{a_{i,J}}(p'_{I-1,J} - p'I, J) \tag{C.8}$$

$$v_{I,j} = v^*_{I,j} + \frac{A_{I,j}}{a_{I,j}}(p'_{I,J-1} - p'I, J) \tag{C.9}$$

The sequence of operations described above is also presented in Fig. (C.1).

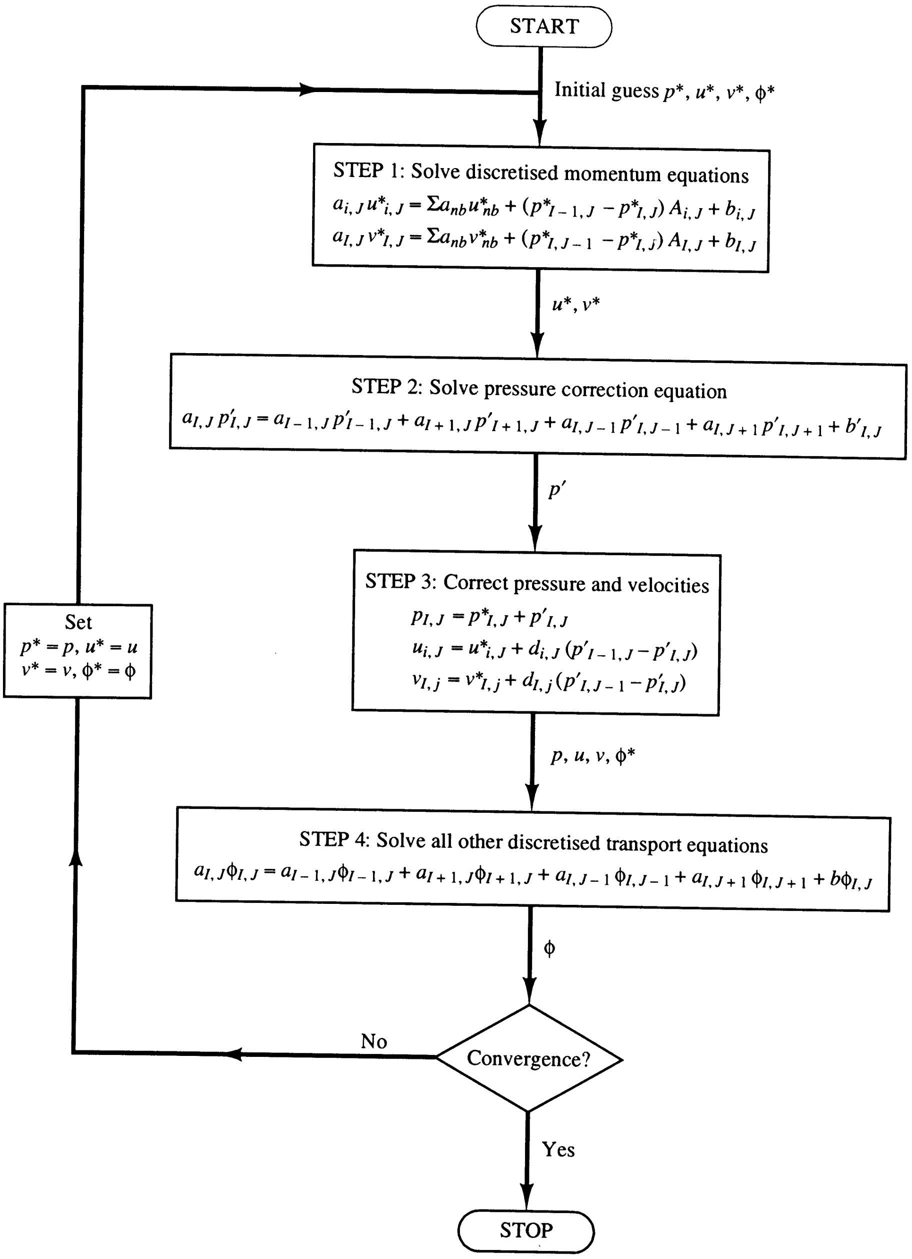

START
Initial guess p^*, u^*, v^*, ϕ^*
STEP 1: Solve discretised momentum equations
$a_{i,J}u^*_{i,J} = \Sigma a_{nb}u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})A_{i,J} + b_{i,J}$
$a_{I,J}v^*_{I,J} = \Sigma a_{nb}v^*_{nb} + (p^*_{I,J-1} - p^*_{I,j})A_{I,J} + b_{I,J}$
u^*, v^*
STEP 2: Solve pressure correction equation
$a_{I,J}p'_{I,J} = a_{I-1,J}p'_{I-1,J} + a_{I+1,J}p'_{I+1,J} + a_{I,J-1}p'_{I,J-1} + a_{I,J+1}p'_{I,J+1} + b'_{I,J}$
p'
STEP 3: Correct pressure and velocities
$p_{I,J} = p^*_{I,J} + p'_{I,J}$
$u_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J})$
$v_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J})$
p, u, v, ϕ^*
STEP 4: Solve all other discretised transport equations
$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\phi_{I,J-1} + a_{I,J+1}\phi_{I,J+1} + b_{\phi_{I,J}}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p, u^* = u$
$v^* = v, \phi^* = \phi$

Figure C.1: The SIMPLE algorithm from [7].

C.2 The SIMPLER algorithm

The SIMPLER (SIMPLE Revised) algorithm of Patankar [8] is an improved version of SIMPLE. In this algorithm, the continuity equation is used to derive an equation for pressure, instead of a pressure correction equation as in SIMPLE. The algorithm is presented in Fig. (C.2).

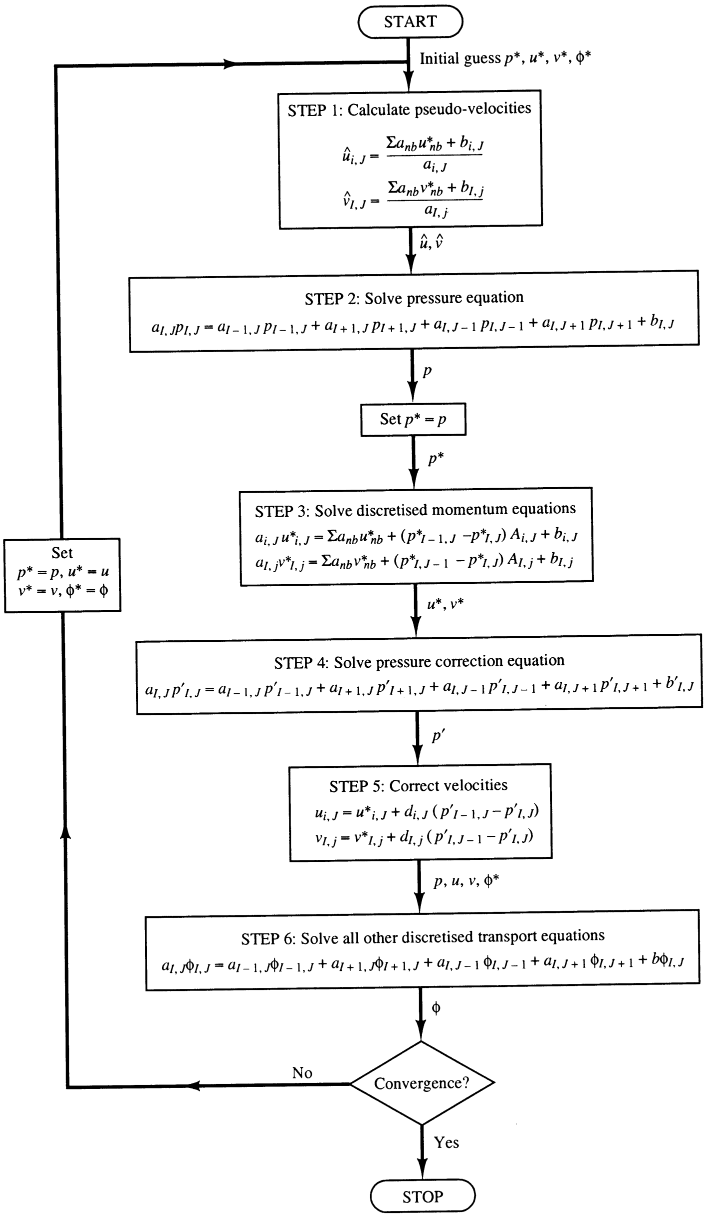

START
Initial guess p^*, u^*, v^*, ϕ^*
STEP 1: Calculate pseudo-velocities
$\hat{u}_{i,J} = \frac{\Sigma a_{nb}u^*_{nb} + b_{i,J}}{a_{i,J}}$
$\hat{v}_{I,J} = \frac{\Sigma a_{nb}v^*_{nb} + b_{I,j}}{a_{I,j}}$
$\hat{u}, \hat{v}$
STEP 2: Solve pressure equation
$a_{I,J}p_{I,J} = a_{I-1,J}p_{I-1,J} + a_{I+1,J}p_{I+1,J} + a_{I,J-1}p_{I,J-1} + a_{I,J+1}p_{I,J+1} + b_{I,J}$
p
Set $p^* = p$
p^*
STEP 3: Solve discretised momentum equations
$a_{i,J}u^*_{i,J} = \Sigma a_{nb}u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})A_{i,J} + b_{i,J}$
$a_{I,j}v^*_{I,j} = \Sigma a_{nb}v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J})A_{I,j} + b_{I,j}$
u^*, v^*
STEP 4: Solve pressure correction equation
$a_{I,J}p'_{I,J} = a_{I-1,J}p'_{I-1,J} + a_{I+1,J}p'_{I+1,J} + a_{I,J-1}p'_{I,J-1} + a_{I,J+1}p'_{I,J+1} + b'_{I,J}$
p'
STEP 5: Correct velocities
$u_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J})$
$v_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J})$
p, u, v, ϕ^*
STEP 6: Solve all other discretised transport equations
$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\phi_{I,J-1} + a_{I,J+1}\phi_{I,J+1} + b_{\phi I,J}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p, u^* = u$
$v^* = v, \phi^* = \phi$

Figure C.2: The SIMPLER algorithm of Patankar [8] form [7].

C.3 The SIMPLEC algorithm

The SIMPLEC (SIMPLE Consistent) algorithm follows the same steps as the SIMPLE algorithm, with the difference that the momentum equations are manipulated so that SIMPLEC velocities correction equations omit terms that are less significant that those omitted in SIMPLE. Thus the u velocity correction equation is given by:

$$u'_{i,J} = d_{i,J}(p'_{I-1,J} - p'_{I,J}) \tag{C.10}$$

where in Eq. (C.10)

$$d_{i,J} = \frac{A_{i,J}}{a_{i,J} - \sum a_{nb}} \tag{C.11}$$

Similarly the v correction equation is

$$v'_{I,j} = d_{I,j}(p'_{I,J-1} - p'_{I,J}) \tag{C.12}$$

where in Eq. (C.12)

$$d_{I,j} = \frac{A_{I,j}}{a_{I,j} - \sum a_{nb}} \tag{C.13}$$

Then the sequence of operations of the SIMPLEC algorithm is identical to that of SIMPLE.

C.4 The PISO algorithm

The PISO (Pressure Implicit with Splitting of Operators) algorithm involves one predictor step and two corrector steps and may be seen as an extension of SIMPLE with a further corrector step to enhance it.

Predictor step

Eq. (C.1) and Eq. (C.2) are solved with a guessed of intermediate pressure field p^* like in the SIMPLE algorithm.

Corrector step 1

The first corrector step is used to give a velocity field (u^{**}, v^{**}) which satisfies the discretized continuity equation. The following notations are used:

$$p^{**} = p^* + p'$$

$$u^{**} = u^* + u'$$

$$v^{**} = v^{*} + v'$$

then u^{**} and v^{**} are given by:

$$u^{**}_{i,J} = u^{*}_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J}) \tag{C.14}$$

$$v^{**}_{I,j} = v^{*}_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J}) \tag{C.15}$$

Corrector step 2

To improve the SIMPLE procedure PISO performs a second correction step. A twice corrected velocity field (u^{***}, v^{***}) may be obtained by solving the momentum equations once more:

$$u^{***}_{i,J} = u^{**}_{i,J} + \frac{\sum a_{nb}(u^{**}_{n}b - u^{*}_{n}b)}{a_{i,J}} + d_{i,J}(p''_{I-1,J} - p''_{I,J}) \tag{C.16}$$

$$v^{***}_{I,j} = v^{**}_{I,j} + \frac{\sum a_{nb}(v^{**}_{n}b - v^{*}_{n}b)}{a_{I,j}} + d_{I,j}(p''_{I,J-1} - p''_{I,J}) \tag{C.17}$$

The algorithm is presented in Fig. (C.3).

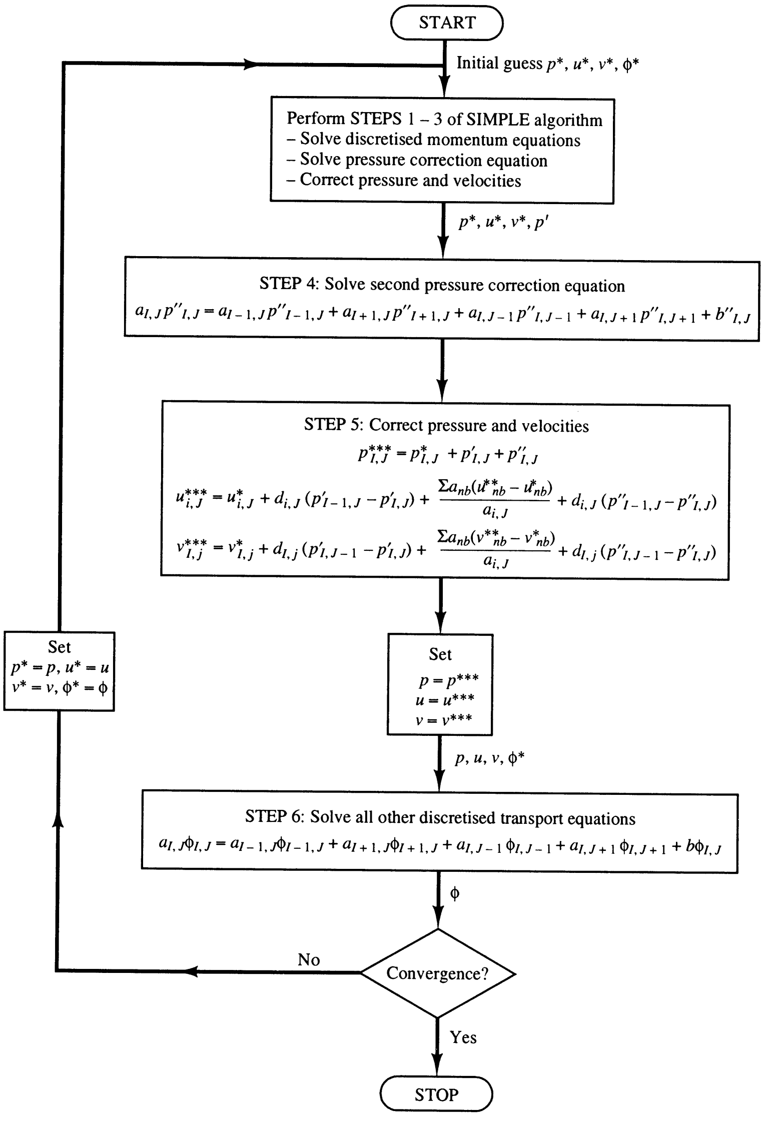

START
Initial guess p^*, u^*, v^*, ϕ^*
Perform STEPS 1 – 3 of SIMPLE algorithm
– Solve discretised momentum equations
– Solve pressure correction equation
– Correct pressure and velocities
p^*, u^*, v^*, p'
STEP 4: Solve second pressure correction equation
$a_{I,J}p''_{I,J} = a_{I-1,J}p''_{I-1,J} + a_{I+1,J}p''_{I+1,J} + a_{I,J-1}p''_{I,J-1} + a_{I,J+1}p''_{I,J+1} + b''_{I,J}$
STEP 5: Correct pressure and velocities
$p^{***}_{I,J} = p^*_{I,J} + p'_{I,J} + p''_{I,J}$
$u^{***}_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J}) + \frac{\Sigma a_{nb}(u^{**}_{nb} - u^*_{nb})}{a_{i,J}} + d_{i,J}(p''_{I-1,J} - p''_{I,J})$
$v^{***}_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J}) + \frac{\Sigma a_{nb}(v^{**}_{nb} - v^*_{nb})}{a_{i,J}} + d_{I,j}(p''_{I,J-1} - p''_{I,J})$
Set
$p = p^{***}$
$u = u^{***}$
$v = v^{***}$
p, u, v, ϕ^*
STEP 6: Solve all other discretised transport equations
$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\phi_{I,J-1} + a_{I,J+1}\phi_{I,J+1} + b_{\phi_{I,J}}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p, u^* = u$
$v^* = v, \phi^* = \phi$

Figure C.3: The PISO algorithm from [7].

Appendix D

List of scientific publications, presentations and conferences

1. “Implementation of a coupled level set volume of fluid method in FLUENT”–European Two-Phase Flow Group Meeting, Pisa, 28 – 30 May 2008;
2. “Development and validation of a CLSVOF method in FLUENT for two-phase flows” – Annual Meeting of the Leonhard Euler Center Swiss ERCOFTAC Pilot Center, 14th November 2008;
3. “A VOF method coupled with a dynamic contact angle model for simulation of two-phase flows with partial wetting”, 7th International Conference on Multiphase Flow, ICMF 2010, Tampa, FL, May 30 – June 4, 2010.
4. “A Level Set Method Coupled with a Volume of Fluid Method for Modeling of Gas-Liquid Interface in Bubbly Flow” accepted for publication in Journal of Fluid Engineering;
5. “A level set method and a heat transfer model implemented into FLUENT for modeling of two phase flows” NATO Research and Technology Organisation (RTO) Applied Vehicle Technology (AVT) Symposion on System Level Thermal Management for Enhanced Platform Efficiency, To be held in Bucharest, Romania

Bibliography

[1] G. Son, V.K Dhir, and N. Ramanujapu. Dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface. *Journal of Heat Transfer*, 121:623–631, 1999.

[2] Ruben Scardovelli and Stephane Zaleski. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41:251–274, 2003.

[3] Jure Mencinger and Iztok Zun. On the finite volume discretization of discontinuous body force field on collocated grid: Application to VOF method. *Journal of Computational Physics*, 221:524–538, 2006.

[4] Mark Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.

[5] I. Zun and J. Groselj. The structure of bubble non-equilibrum movement in free-rise and agitated-rise conditions. *Nuclear Engineering and Design*, 163:99–115, 1996.

[6] Graham B. Wallis. The terminal speed of single drops or bubbles in an infinite medium. *International Journal of Multiphase Flow*, 1:491–511, 1974.

[7] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics.* Pearson, 1995.

[8] Suhas V. Patankar. *Numerical heat transfer and fluid flow.* Hemisphere publishing corporation, 1980.

[9] D. Bhaga and M.E. Weber. Bubbles in Viscous Liquids: Shapes, Wakes and Velocities. *Journal of Fluid Mechanics*, 105:61–85, 1981.

[10] J.G Hnat and J.D. Buckmaster. Spherical cap bubbles and skirt formation. *The Physics of Fluids*, 19:182–194, 1976.

[11] F.N. Peebles and H.J. Garber. Studies of the motion of gas bubbles in liquids. *Chemical Engineering Progress*, 49:88–97, 1953.

[12] F. Harlow and J. Welch. Volume tracking methods for interfacial flow calculations. *Physics of Fluids*, 8:21–82, 1965.

[13] J. Donea. Arbitrary Lagrangian-Eulerian finite element methods. *Computational Methods for Transient Analisys*, 1:473–516, 1983.

[14] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135:203–216, 1997.

[15] T. Hughes, W. Liu, and T. Zimmermann. Lagrangian Eulerian finite element formulation for incompressible viscous flow. *Computer Methods in Applied Mechanics and Engineering*, 29:239–349, 1981.

[16] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A Front Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169:708–759, 2001.

[17] Salih Ozen Unverdi and Gretar Tryggvason. A Front Tracking Method for Viscous, Incompressible, Multi – fluid Flows. *Journal of Computational Physics*, 100:25–37, 1992.

[18] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[19] D.L. Youngs. Time dependent multimaterial flow with large fluid distortion. In K. Morton and M. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285. Academic Press, New York, 1982.

[20] Jie Li. Piecewise Linear Interface Calculation. *Comptes Rendus de l'Academie des Sciences Serie II. Fascicule B - Mecanique*, 320:391–396, 1995.

[21] Ruben Scardovelli and Stephane Zaleski. Direct numerical simulation of free surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:657–603, 1999.

[22] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[23] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5–6):663–680, 1998.

[24] Mark Sussman, Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, and Michael L. Welcome. An Adaptive Level Set Approach for Incompressible Two Phase Flows. *Journal of Computational Physics*, 148:81–124, 1999.

[25] Mark Sussman and Emad Fatemi. An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

[26] Sergey V. Shepel, Brian L. Smith, and Samuel Paolucci. Implementation of a Level Set Interface Tracking Method in the FIDAP and CFX-4 Codes. *Journal of Fluids Engineering*, 127:674–686, 2005.

[27] Mark Sussman and Elbridge Gerry Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162:301–337, 2000.

[28] Gihun Son and Nahmkeon Hur. A coupled level set and volume-of-fluid method for buoyancy-driven motion of fluid particles. *Numerical Heat Transfer, Part B*, 42:523–542, 2002.

[29] Gihun Son. Efficient implementation of a coupled level-set and volume of fluid method for three dimensional incompressible two-phase flow. *Numerical Heat Transfer, Part B*, 43:549–565, 2003.

[30] Xiaofeng Yang, Ashley J. James, John Lowengrub, Xiaoming Zheng, and Vittorio Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217:364–394, 2006.

[31] M. Sussuman, K.M. Smith, M.Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 221:469–505, 2006.

[32] Sergey V. Shepel and Brian L. Smith. New finite-element/finite-volume level set formulation for modelling two-phase incompressible flows. *Journal of Computational Physics*, 218:479–494, 2006.

[33] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dinamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences.* Springer-Verlag New York Inc., 2003. ISBN 0–387–95482–1.

[34] J. A. Sethian. *Level Set Methods and Fast Marching Methods*, volume 3 of *Cabridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, 1999. ISBN 0 521 64557 3.

[35] Y. C. Chang, T. Y. Hou, B. Marriman, and S. Osher. A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows. *Journal of Computational Physics*, 124:449–464, 1996.

[36] Mark Sussman and Peter Smereka. Axisymmetric free boundary problems. *Journal of Fluid Mechanics*, 341:269–294, 1997.

[37] M. Quecedo and M. Pastor. Application of the level set method to the finite element solution of two phase flows. *International Journal for Numerical Methods in Engineering*, 50:645–663, 2001.

[38] Wusi Yue, Ching-Long Lin, and Virendra C. Patel. Numerical simulation of unsteady multidimensional free surface motions by level set method. *International Journal for Numerical Methods in Fluids*, 42:853–884, 2003.

[39] Haruhiko Kohno and Takahiko Tanahashi. Numerical analysis of moving interfaces using a level set method coupled with adaptive mesh refinement. *International Journal for Numerical Methods in Fluids*, 45:921–944, 2004.

[40] Elin Olsson and Gunilla Kreiss. A conservative level set method for tho phase flow. *Journal of Computational Physics*, 210:225–246, 2005.

[41] Peter D. M. Spelt. A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. *Journal of Computational Physics*, 207:389–404, 2005.

[42] P. Gómez, J. Hernández, and J. López. On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows. *International Journal for Numerical Methods in Engineering*, 63:1478–1512, 2005.

[43] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[44] Sunitha Nagrath, Kenneth E. Jansen, and Richard T. Lahey Jr. Computation of incompressible bubble dynamics with a stabilized finite element level set method. *Computer Methods in Applied Mechanics and Engineering*, 194:4565–4587, 2005.

[45] J.U. Brackbill, D.B. Kothe, and C. Zemach. A Continuum Method for Modeling Surface Tension. *Journal of Computational Physics*, 100:335–354, 1992.

[46] Ching-Long Lin, Haegyun Lee, Taehun Lee, and Larry J. Weber. A level set characteristic Galerkin finite element method for free surface flows. *International Journal for Numerical Methods in Fluids*, 49:521–547, 2005.

[47] S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving Level-Set method for modelling of multiphase flows. *International Journal for Numerical Methods in Fluids*, 47:339–361, 2005.

[48] Anton Smolianski. Finite-element/level-set/operator splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces. *International Journal for Numerical Methods in Fluids*, 48:231–269, 2005.

[49] Leo Miguel Gonzáles Gutiérez and Rodolfo Bermejo. A semi-Lagrangian level set method for incompressible Navier-Stokes equations with free surface. *International Journal for Numerical Methods in Fluids*, 49:1111–1146, 2005.

[50] Xiaoming Zheng, John Lowengrub, Anthony Anderson, and Vittorio Cristini. Adaptive unstructured volume remeshing – II: Application to two- and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics*, 208:626–650, 2005.

[51] Yang Ge and L.-S. Fan. 3-D modelling of the dynamics and heat transfer characteristics of subcooled droplet impact on a surface with film boiling. *International Journal of Heat and Mass Transfer*, 49:4231–4249, 2006.

[52] Emilie Marchandise, Jean-François Remacle, and Nicolas Chevaugenon. A quadrature-free discontinuous Galerkin method for the level set equation. *Journal of Computational Physics*, 212:338–357, 2006.

[53] Emilie Marchandise and Jean-François Remacle. A stabilized finite element method using a discontinuous level set approach for solving two-phase incompressible flows. *Journal of Computational Physics*, 219:780–800, 2006.

[54] Kiran B. Deshpande and William B. Zimmerman. Simulation of interfacial mass transfer by droplet dynamics using the level set method. *Chemical Engineering Science*, 61:6486–6498, 2006.

[55] X.Y. Hu, B.C. Khoo, N.A. Adams, and F.L. Huang. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219:553–578, 2006.

[56] Sunitha Nagrath, Kenneth Jansen, Richard T. Lahey Jr., and Iskander Akhatov. Hydrodynamic simulation of air bubble implosion using a level set approach. *Journal of Computational Physics*, 215:98–132, 2006.

[57] W. G. Price and Y. G. Chen. A simulation of free surface waves for incompressible two phase flows using a curvilinear level set formulation. *International Journal for Numerical Methods in Fluids*, 51:305–330, 2006.

[58] Frank Losasso, Ronald Fedkiw, and Stanley Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids*, 35:995–1010, 2006.

[59] Koji Fukagata, Nobuhide Kasagi, Poychat Ua-arayaporn, and Takehiro Himeno. Numerical simulation of gas-liquid two-phase flow and convective heat transfer in a microtube. *International Journal of Heat and Fluid Flow*, 28:72–82, 2007.

[60] Frédéric Gibou, Liguo Chen, Duc Nguyen, and Sanjoy Banerjee. A level set based sharp interface method for the multiphase incompressible Navier-Stokes equations with phase change. *Journal of Computational Physics*, 222:536–555, 2007.

[61] Sébastien Tanguy, Thibaut Ménard, and Alain Berlemont. A Level Set Method for vaporizing two-phase flows. *Journal of Computational Physics*, 221:837–853, 2007.

[62] P. M. Carrica, R. V. Wilson, and F. Stern. An unsteady single-phase level set method for viscous free surface flows. *International Journal for Numerical Methods in Fluids*, 53:229–256, 2007.

[63] Akio Tomiyama, Iztok Zun, Akira Sou, and Tadashi Sakaguchi. Numerical analysis of bubble motion with the VOF method. *Nuclear Engineering and Design*, 141:69–82, 1993.

[64] D.B. Kothe, W.J. Rider, S.J. Mosso, and J.S. Brock. Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions. Technical report, Los Alamos, Los Alamos NM, 1996.

[65] Murray Rudman. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24:671–691, 1997.

[66] A. Tomiyama, I. Zun, H. Higaki, Y. Makino, and T. Sakaguchi. A three-dimensional particle tracking method for bubbly flow simulation. *Nuclear Engineering and Design*, 175:77–86, 1997.

[67] J.H Jeong and D.Y. Yang. Finite Element Analysis of Transient Fluid Flow with Free Surface using VOF (volume-of-fluid) Method and Adaptive Grid. *International Journal for Numerical Methods in Fluids*, 26:1127–1154, 1998.

[68] Li Chen and Yuguo Li. A numerical method for two phase flows with an interface. *Enviromental Modeling & Software*, 13:247–255, 1998.

[69] William J. Rider and Douglas B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141:112–152, 1998.

[70] Denis Gueyffier, Jie Li, Ali Nadim, Ruben Scardovelli, and Stephane Zaleski. Volume-of-fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows. *Journal of Computational Physics*, 152:423–456, 1999.

[71] Samuel W.J. Welch and John Wilson. A Volume of Fluid Based Method for Fluid Flows with Phase Change. *Journal of Computational Physics*, 60:662–682, 2000.

[72] Suho Shin and Woo Il Lee. Finite element analysis of incompressible viscous flow with moving free surface by selective volume of fluid method. *International Journal of Heat and Fluid Flow*, 21:197–206, 2000.

[73] I. Ginzburg and G Wittum. Two phase flow on interface refined grids modeled with VOF, Staggered finite volumes and spline interpolants. *Journal of Computational Physics*, 166:302–335, 2001.

[74] Gregor Cerne, Stojan Petelin, and Iztok Tiselj. Coupling of the Interface Tracking and the Two-Fluid Models for the Simulation of Incompressible Two-Phase Flow. *Journal of Computational Physics*, 171:776–804, 2001.

[75] M. Wörner. The influence of gas-liquid density ratio on shape and rise velocity of an ellipsiodal bubble: a numerical study by 3D volume-of-fluid computations. In *1st International Berlin Workshop (IBW1) on Transport Phenomena with Moving Boundaries*, pages 67–84, Berlin Germany, 2001.

[76] Markus Meier, George Yadigaroglu, and Brian L. Smith. A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics B/Fluids*, 21:61–73, 2002.

[77] S. Zaleski, J. Li, S. Succi, R. Scardovelli, and G. Zanetti. Direct numerical simulation of flows with interfaces. In *2nd International Conference on Multiphase Flow*, Kyoto, 1995.

[78] B.E. Ghidersa. *Finite Volume-based Volume of Fluid Method for the Simulation of Two Phase Flows in Small Rectangular Channels.* PhD thesis, TH, Karlsruhe, 2003.

[79] Eugenio Aulisa, Sandro Manservisi, and Ruben Scardovelli. A mixed markers and volume of fluid method for the reconstruction and advection of interfaces in two phase and free boundary flows. *Journal of Computational Physics*, 188:611–639, 2003.

[80] Min Soo Kim and Woo Il Lee. A new VOF - based numerical scheme for the simulation of fluid flow with free surface. Part I: New free surface - tracking algorithm and its verification. *International Journal for Numerical Methods in Fluids*, 42:765–790, 2003.

[81] Min Soo Kim, Jong Sun Park, and Woo Il Lee. A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part II: application to the cavity filling and sloshing problems. *International Journal for Numerical Methods in Fluids*, 42:791–812, 2003.

[82] Hui Gao, Han-Yang Gu, and Lie-Jin Guo. Numerical study of stratified oil-water two-phase turbulent flow in a horizontal tube. *International Journal of Heat and Mass Transfer*, 46:749–754, 2003.

[83] A. Theodorakakos and G. Bergeles. Simulation of sharp gas-liquid interface using VOF method and an adaptive grid local refinement around the interface. *International Journal for Numerical Methods in Fluids*, 45:421–439, 2004.

[84] Eugenio Aulisa, Sandro Manservisi, and Ruben Scardovelli. A surface marker algorithm coupled to an area-preserved marker redistribution method for three-dimensional interface tracking. *Journal of Computational Physics*, 197:555–584, 2004.

[85] Benjamin Biausser, Stephan Guignard, Richard Marcher, and Philippe Fraunie. 3D two phase flows numerical simulations by SL-VOF method. *International Journal for Numerical Methods in Fluids*, 45:581–604, 2004.

[86] Daniel Lorstad and Laszlo Fuchs. High order surface tension VOF model for 3D bubble flows with high density ratio. *Journal of Computational Physics*, 200:153–176, 2004.

[87] Daniel Lorstad, Marianne Francois, Wei Shyy, and Laszlo Fuchs. Assessment of volume of fluid and immersed boundary methods for droplet computations. *International Journal for Numerical Methods in Fluids*, 46:109–125, 2004.

[88] J.P. Wang, A.G.L. Borthwick, and R. Eatock Taylor. Finite-volume-type VOF method on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids*, 45:485–508, 2004.

[89] M. van Sint Annaland, N.G. Deen, and J.A.M. Kuipers. Numerical simulation of gas bubble behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science*, 60:2999–3011, 2005.

[90] J.R. Grace. Shapes and velocities of bubbles rising in infinite liquids. *Transactions of the Institution of Chemical Engineering*, 51:116–120, 1973.

[91] J. Lopez, J. Hernandez, P. Gomez, and F. Faura. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208:51–74, 2005.

[92] G. Yadigaroglu. Computational Fluid Dynamics for nuclear applications: from CFD to multi-scale CMFD. *Nuclear Engineering and Design*, 235:153–164, 2005.

[93] H. Tang and L.C. Wrobel. Modelling the interfacial flow of two immiscible liquids in mixing processes. *International Journal of Engineering Science*, 43:1234–1256, 2005.

[94] Jian Du, Brian Fix, James Glimm, Xicheng Jia, Xiaolin Li, Yuanhua Li, and Lingling Wu. A simple package for front tracking. *Journal of Computational Physics*, 213:613–628, 2006.

[95] George Yadigaroglu and Djamel Lakehal. New challenges in computational thermal hydraulics. *Nuclear Technology*, 152:239–251, 2005.

[96] Taha Taha and Z.F. Cui. CFD modeling of slug flow inside square capillaries. *Chemical Engineering Science*, 61:665–675, 2006.

[97] Taha Taha and Z.F. Cui. CFD modeling of slug flow in vertical tubes. *Chemical Engineering Science*, 61:676–687, 2006.

[98] Subhashini Ghorai and K.D.P. Nigam. CFD modeling of flow profiles and interfacial phenomena in two phase flow in pipes. *Chemical Engineering and Processing*, 45:55–65, 2006.

[99] K.D. Kim, D.Y. Yang, and J.H. Jeong. Adaptive refinement techniques based on tetrahedral and hexahedral grids for finite element analysis of mold filling in casting processes. *Computer Methods in Applied Mechanics and Engineering*, 195:6799–6821, 2006.

[100] D. Lakehal and C. Narayanan. Computational Heat Transfer and Two - phase Flow in Miniature Tubes. In *NSTI Nanotechnology Conference, Nanotech, Symp. Nano & Micro fluidics, paper 586*, page paper 586, 2006.

[101] S. Popinet and S. Zaleski. A front-tracking algorithm for the accurate representation of surface tension. *International Journal of Numerical Methods in Fluids*, 30:775–793, 1999.

[102] S. Shin, S.I. Abdel-Khalik, V. Daru, and D. Juric. Accurate representation of surface tension using the level set contour recontruction method. *Journal of Computational Physics*, 203:493–516, 2005.

[103] Albert Y. Tong and Zhaoyuan Wang. A numerical method for capillarity-dominat free surface flows. *Journal of Computational Physics*, 221:506–523, 2007.

[104] Dhir V. K. Nucleate and transition boiling heat transfer under pool and external flow conditions. *International Journal of Heat and Fluid Flow*, 21:290–314, 1991.

[105] Ramanujapu N. and V.K. Dhir. On the formation of vapor columns and mushroom type bubbles during nucleate boiling on a horizontal surface. In *NHTC 2000-12208*, 2000.

[106] A. Mukherjee and V.K. Dhir. Study of lateral merger of vapor bubbles during nucleate pool boiling. *Journal of Heat Transfer*, 126:1023–1039, 2004.

[107] Tsung-Shann Jiang, Soo-Gun Oh, and John C. Slattery. Correlation for Dynamic Contact Angle. *Journal of Colloid and lnterface Science*, 69:74–77, 1979.

[108] L.M. Hocking. On contact angles in evaporating liquids. *Physics of Fluids*, 7:2950–2955, 1995.

[109] Michael Renardy, Yuriko Renardy, and Jie Li. Numerical simulation of moving contact line problems using a volume of fluid method. *Journal of Computational Physics*, 171:243–263, 2001.

[110] Hansmartin Friess and Djamel Lakehal. A new method for including interfacial tension and wetting dynamics in the simulation of two phase flow. In *ICHMT International Symposium on Advances in Computational Heat Transfer*, Norway, 2004.

[111] A. Theodorakakos, T. Ous, M. Gavaises, J.M. Nouri, N. Nikolopoulos, and H. Yanagihara. Dynamics of water droplets detached from porous surfaces of relevance to pem fuel cells. *Journal of Colloids and Interface Science*, 300:673–687, 2006.

[112] C. Narayanan and D. Lakehal. Simulation of filling of microfluidic devices using a coarse-grained continuum contact angle model. In *NSTI Nanotechnology Conference, Nanotech, Symp. Nano & Micro fluidics, paper 770*, 2006.

[113] Abhijit Mukherjee and Satish G. Kandlikar. Numerical study of single bubbles with dynamic contact angle during nucleate pool boiling. *International Journal of Heat and Mass Transfer*, 50:127–138, 2007.

[114] Bhavin Vadgama and Daniel K. Harris. Measurements of the contact angle between r134a and both aluminium and copper surfaces. *Experimental Thermal and Fluid Science*, 31:979–984, 2007.

[115] Benjamin Y. Choi and Markus Bussmann. A piecewise linear approach to volume tracking a triple point. *International Journal for Numerical Methods in Fluids*, 53:1005–1018, 2007.

[116] Chen Fang, Carlos Hidrovo, Fu-min Wang, John Eaton, and Kenneth Goodson. 3-D numerical simulation of contact angle hysteresis for microscale two phase flow. *International Journal of Multiphase Flow*, 34:690–705, 2008.

[117] Taewoong Lim, Sewoon Han, Jaewon Chung, Jin Taek Chung, Seunghwan Ko, and Costas P. Grigoropoulos. Experimental study on spreading and evaporation of inkjet printed pico-liter droplet on a heated substrate. *International Journal of Heat and Mass Transfer*, 52:431–441, 2009.

[118] K. Grundke, S. Michel, G Knispel, and A. Grundler. Wettability of silicone and polyether impression materials: Characterization by surface tension and contact angles measurements. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 317:598–609, 2008.

[119] S. Manservisi and R. Scardovelli. A variational approach to the contact angle dynamics of spreading droplets. *Computers & Fluids*, 38:406–424, 2009.

[120] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[121] Guang-Shan Jiang and Danping Peng. Weighted ENO schemes for Hamilton–Jacobi equations. *Journal on Scientific Computing*, 21(6):2126–2143, 2000.

[122] S.K Godunov. Finite difference methods for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mathematics Sbornik*, 47:271–306, 1959.

[123] W. F. Noh and Paul Woodward. SLIC (Simple Line Interface Calculation). In *Proceeding of the Fifth International Conference on Numerical Methods on Fluid Dynamics*, volume 59 of *Lecture Notes in Physics*, pages 330–340. Springer, 1976.

[124] L H Tanner. The spreading of silicone oil drops on horizontal surfaces. *Journal of Physics D: Applied Physics*, 12:1473–1484, 1979.

[125] Ruben Scardovelli and Stephane Zaleski. Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids. *Journal of Computational Physics*, 164:228–237, 2000.

[126] *FLUENT 6.3 User's Guide.*

[127] J.H. Ferziger and M. Perić. *Computational methods for fluid dynamics.* Springer-Verlag, 2001.

[128] Anna-Karin Tornberg and Björn Engquist. A finite element based level-set method for multiphase flow applications. *Computing and Visualization in Science*, 3:93–101, 2000.

[129] Roland Clift, John R. Grace, and Martin E. Weber. *Bubbles, Drops and Particles.* Academic Press, New York, 1978.

[130] Akio Tomiyama. Struggle with computational bubble dynamics. In *Third International Conference on Multiphase Flow, ICMF'98, Lyon, France, June 8–12*, 1998.

[131] A. Tomiyama, Y Nakahara, and G. Morita. Rising velocities and shapes of single bubbles in vertical pipes. In *International Conference of Multiphase Flow*, pages 1–12, 2001.

[132] Taewoong Lim, Sewoon Han, Jaewon Chung, Jin Taek Chung, and Seunghwan Ko. Experimental study on spreading and evaporation of inkjet printed pico-liter droplet on a heated substrate. *International Journal of Heat and Mass Transfer*, 2008.

[133] Youngbae Han and Naoki Shikazono. Measurement of the liquid film thickness in micro tube slug flow. *International Journal of Heat and Fluid Flow*, 30:842–853, 2009.

[134] K. Moriyama and A. Inoue. Thickness of a liquid film formed by a growing bubble in a narrow gap between two horizontal plates. *Journal of Heat Transfer*, 118:132–139, 1996.

[135] C.-W. Park and G.M. Homsy. Two-phase displacement in hele shaw cells: theory. *Journal of Fluid Mechanics*, 139:291–308, 1984.

[136] Pascale Aussilous and David Quéré. Quick deposition of a fluid on the wall of the tube. *Physics of Fluids*, 12:2367–2371, 2000.

[137] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787–1806, 1972.

Curriculum Vitae

PERSONAL DATA

Name	**Bogdan Alexandru Nichita**
Date of birth	8th May, 1977
Birthplace	Bucharest, Romania
Nationality	Romanian

EDUCATION

2005–2009	**Doctoral Student and Assistant**, Laboratory of Heat and Mass Transfer, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
2000–2001	**Master studies in Thermal Engineering**, Technical University of Civil Engineering Bucharest, Romania
1995–2000	**Bachelor studies in Mechanical Engineering**, Technical University of Civil Engineering Bucharest, Romania
1991–1995	**High School Certificate** , Dante Alighieri, Bucharest, Romania

WORKING EXPERIENCE

2005–2010	**Swiss Federal Institute of Technology (EPFL), Laboratory of Heat and Mass Transfer, Lausanne** Work supervised by Prof. John R.Thome;
2003–2005	**Technical University for Civil Engineering Bucharest, Romania**, Assistant Professor Department of Heat and Mass Transfer.
2001–2003	**Technical University for Civil Engineering Bucharest, Romania**, Research Assistant Department of Heat and Mass Transfer.
2001–2004	**Forchungszentrum Karlsruhe, Germany, Institute for Reactor Safety** Three stages of 6 months each as a Research Assistant Topic: Numerical modeling of single phase flow in counter flow micro heat exchangers.