

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA MECÂNICA**

**Estudo numérico do transporte de bolhas/gotas em fluidos
newtonianos através das metodologias VOF e S-CLSVOF
utilizando o OpenFOAM**

Autor:

Angel Edecio Malaguera Mora

Itajubá, 15 de Fevereiro de 2018

Minas Gerais – Brasil

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA MECÂNICA**

Angel Edecio Malaguera Mora

**Estudo numérico do transporte de bolhas/gotas em fluidos
newtonianos através das metodologias VOF e S-CLSVOF
utilizando o OpenFOAM**

**Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica como
parte dos requisitos para obtenção do Título de
Doutor em Ciências em Engenharia Mecânica.**

Área de Concentração: Térmica, Fluidos e Máquinas de Fluxo

Orientadora: Profa. Dra. Ana Lúcia Fernandes de Lima e Silva

Coorientador: Prof. Dr. Sandro Metrevelle Marcondes de Lima e Silva

Itajubá, 15 de Fevereiro de 2018

Minas Gerais – Brasil

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA MECÂNICA

Angel Edecio Malaguera Mora

Estudo numérico do transporte de bolhas/gotas em fluidos newtonianos através das metodologias VOF e S-CLSVOF utilizando o OpenFOAM

Tese aprovada pela banca examinadora em 15 de Fevereiro de 2018, conferindo ao autor o título de *Doutor em Ciências em Engenharia Mecânica*.

Banca Examinadora:

Prof. Dr. André Luiz Seixlack - UNESP (Ilha Solteira)

Prof. Dr. João Batista Campos Silva - UNESP (Ilha Solteira)

Prof. Dr. Genésio José Menon - UNIFEI

Prof. Dr. Marcelo José Pirani - UNIFEI

Prof. Dra. Ana Lúcia Fernandes de Lima e Silva - UNIFEI

Prof. Dr. Sandro Metrevelle Marcondes de Lima e Silva - UNIFEI

Itajubá, 15 de Fevereiro de 2018

Minas Gerais – Brasil

Dedicatória

Dedico estas linhas à memória de minha cunhada:

Inelda Caridad Utrera de Malaguera

Que Deus a tenha na sua glória.

Agradecimento

Agradeço a Deus acima de tudo, ele é o meu guia.

À minha orientadora, Profa. Dra. Ana Lúcia Fernandes de Lima e Silva, pela competência, dedicação, paciência e amizade. Obrigado pela confiança no meu trabalho, pelo respeito, por me ensinar, pela compreensão e pelos sábios conselhos sempre que a procurei para conversar.

Ao meu coorientador, Prof. Dr. Sandro Metrevelle Marcondes de Lima e Silva também pela confiança, pela orientação e por me receber como aluno de Doutorado no seu grupo de pesquisa.

Ao Dr. Hektor Monteiro do Instituto de Física e Química da UNIFEI, por facilitar o uso do *Cluster* do Laboratório de Astrofísica Computacional (LAC-UNIFEI). O LAC-UNIFEI é mantido com subsídios da CAPES, CNPq e FAPEMIG. Igualmente agradeço ao Eng. José Renato Castro, analista de tecnologia da informação da UNIFEI, por seu apoio na instalação e manipulação dos programas do *Cluster*.

Aos amigos e companheiros do laboratório LabTC da UNIFEI, especialmente meu amigo Cristiano Pedro da Silva por seu apoio incondicional. Obrigado pelos seus ensinamentos na parte computacional, isso foi crucial nos primeiros anos desta pesquisa.

Aos professores do Instituto de Engenharia Mecânica da Universidade Federal de Itajubá, que têm sido parte da minha formação.

À CAPES pela bolsa de Doutorado e ao CNPq e FAPEMIG, pelo apoio financeiro.

À minha mãe Alejandra, suas orações foram escutadas. Obrigado mãe, herdei de você a coragem e o espírito de luta.

À minha esposa Belkys Cecilia e meus filhos Angel Josué e Maria de los Angeles, minha razão de luta.

Aos meus irmãos Santos, Hilda, Lourdes, José e Oscar, pelo apoio e sábios conselhos no momento de sair da casa e buscar novos horizontes.

Contudo, a minha fé não me permite deixar de agradecer a Deus. Pois “tudo posso naquele que me fortalece (Fl 4.13)”.

“Deus me permita falar com conhecimento, e meus pensamentos sejam dignos dos dons que me foram concedidos. Ele é o guia da Sabedoria e o condutor dos sábios”.

Livro da Sabedoria, 7,15

Resumo

MALAGUERA, M. A. E. (2018), *Estudo numérico do transporte de bolhas/gotas em fluidos newtonianos através das metodologias VOF e S-CLSVOF utilizando o OpenFOAM*, 125p. Tese de Doutorado (Térmica, Fluidos e Máquinas de Fluxo) - Instituto de Engenharia Mecânica, Universidade Federal de Itajubá.

Escoamentos bifásicos são encontrados comumente em muitas aplicações industriais tais como no transporte de óleo, em reatores químicos, em resfriamento de reatores nucleares, em processos de aeração e outros. O estudo numérico das bolhas ascendendo em meios viscosos e gotas escoando em microcanais com junções em T são alguns destes exemplos e ainda existem muitos desafios em seus estudos numéricos, especialmente os relacionados com a definição correta da forma da interface entre os dois fluidos. No caso das bolhas ascendendo em meios líquidos, sua forma final tem dependência direta com os números adimensionais Eötvös, Morton e Reynolds. Em gotas escoando em microcanais através de junções em T , o regime de ruptura é fortemente influenciado pelo número de Capilaridade e por fatores geométricos como a razão entre o comprimento inicial da gota (l_o) e a dimensão transversal do microcanal (w). Várias técnicas numéricas de modelagem destes tipos de escoamentos bifásicos estão disponíveis na literatura e as mais comuns são os métodos de captura da interface tais como o método de Volume de Fluido (VOF) e método *Level Set* (LS). O objetivo do presente trabalho é estudar numericamente a dinâmica de uma bolha de gás ascendendo em um fluido estacionário e uma gota escoando em um microcanal com junção em T , por meio das metodologias VOF e o acoplamento VOF e LS (S-CLSVOF). As simulações foram realizadas com o software aberto OpenFOAM, no qual a metodologia VOF está implementada no módulo *interFoam*, e o acoplamento S-CLSVOF que foi implementado durante o desenvolvimento deste trabalho. Os resultados obtidos permitiram comparar as vantagens e desvantagens das duas metodologias. Uma das grandes dificuldades na simulação de gotas em microcanais é o fato de que, com baixos números de capilaridade, as correntes parasitárias tendem a deformar a interface. Nas simulações realizadas estes erros numéricos afetaram o regime de ruptura da gota através da junção em T . Para contornar este problema foi escolhido um número de Courant adequado para cada simulação, porém, com a implementação da metodologia S-CLSVOF conseguiu-se reduzir a formação destas correntes parasitárias. Os resultados das simulações tanto de bolhas ascendendo em meios líquidos quanto de gotas escoando em microcanais tiveram boa concordância com a literatura consultada.

Palavras-chave

Escoamento bifásico, Volume de Fluido, S-CLSVOF, OpenFOAM.

Abstract

MALAGUERA, M. A. E. (2018), *Numerical study of the bubble/droplets transport in newtonian fluids using the VOF and S-CLSVOF methodologies using OpenFOAM software*, 125 p. Doctoral thesis – Intitute of Mechanical Engineering, Federal University of Itajubá.

Two-phase flow is commonly found in many industrial applications such as oil transportation, chemical reactors, cooling of nuclear reactors, aeration processes, etc. The numerical study of bubbles ascending in viscous media and droplets flowing into microchannels with T-junctions are some of these examples, and there are still many challenges in their numerical studies, especially those related to the correct definition of the shape of the interface between the two fluids. In the case of bubbles ascending in liquid media, its final form has direct dependence on the dimensionless numbers Eötvös, Morton and Reynolds. In droplets flowing in microchannels through T-junctions, the rupture regime is strongly influenced by the capillarity number and by geometric factors such as the ratio of the initial drop length (l_o) to the transverse dimension of the microchannel (w). The techniques of modeling these types of two-phase flows are diverse, and the most common are the interface capture methods such as Volume of Fluid (VOF) and Level Set (LS). In the present work the objective was to study the dynamics of a bubble of gas ascending in viscous media and the dynamic of a drop flowing into microchannels with T-junction using VOF methodologies and the coupling VOF-Level Set (S-CLSVOF). The simulations were performed with OpenFOAM in which the VOF methodology is implemented in the interFoam module, and the S-CLSVOF methodology, which was implemented in its source code. The results obtained allowed comparing the advantages and disadvantages of the two methodologies. One of the great difficulties in the simulation of droplets in microchannels is the fact that, with low capillarity numbers, parasitic currents tend to deform the interface. In our simulations these numerical errors affected the regime of rupture of the drop through the T-junction. Assigning a suitable number of Courant we find to reduce them, however, with the S-CLSVOF methodology we obtained a lower tendency to the formation of these parasitic currents. The results of the simulations of both bubbles ascending in liquid media and droplets flowing in microchannels have good agreement with the literature consulted.

Keywords

Two-phase flow, Volume of Fluid, S-CLSVOF, OpenFOAM.

LISTA DE FIGURAS

Figura 1.1: Exemplos de escoamentos bifásicos..	21
Figura 1.2: Dispositivos com escoamentos bifásicos.	22
Figura 2.1: Fragmentação de uma gota em uma superfície livre.	28
Figura 2.2: Diagrama de Clift.	30
Figura 2.3: Ruptura de uma gota em um microcanal com junção em T .	32
Figura 2.4: Diagrama representativo da ruptura de gotas em um microcanal com junção em T .	32
Figura 2.5: Diagramas que representam os comportamentos de gotas movimentando-se através de uma junção em T para duas razões de viscosidades.	33
Figura 2.6: Diagramas de mecanismo de ruptura em junções em T .	34
Figura 2.7: Diagrama representativo do mecanismo de ruptura de gotas em junções em T .	34
Figura 2.8: Técnicas de Modelagem de escoamentos multifásicos.	36
Figura 2.9: Formas de representar superfícies livres e interfaces entre fluidos.	37
Figura 2.10: Função volumétrica α no método VOF.	38
Figura 2.11: Comparação das diferentes técnicas para reconstrução da interface.	39
Figura 2.12: Função Level Set.	42
Figura 3.1: Fluxograma do acoplamento S-CLSVOF.	51
Figura 4.1: Discretização nos domínios de espaço e tempo.	60
Figura 4.2: Esquema de pastas que formam um caso no OpenFOAM.	66
Figura 4.3: Comparação da forma de uma bolha ascendendo num meio líquido com os <i>solvers</i> (a) <i>interFoam</i> e (b) <i>interDyMFoam</i> .	68
Figura 4.4: Tela de cálculo da velocidade baricêntrica utilizando <i>ParaView</i> .	70
Figura 5.1: Vista da bolha no instante inicial dentro do domínio computacional em corte no plano xy .	72
Figura 5.2: Influência da relação do refinamento sobre a velocidade da bolha.	73
Figura 5.3: Influência da relação do refinamento sobre a forma da bolha.	74
Figura 5.4: Refinamento dinâmico sobre a interface da bolha.	74
Figura 5.5: Influência do número $nBufferLayers$ na velocidade da bolha.	75

Figura 5.6: Influência do número $nBufferLayers$ na forma final da bolha.	75
Figura 5.7: Domínio computacional em três dimensões para o Caso 3.	76
Figura 5.8: Evolução temporal da velocidade baricêntrica para os casos simulados com condição de não deslizamento e livre deslizamento com o solver <i>interDyMFoam</i>	77
Figura 5.9: Dimensões do domínio computacional adotado neste trabalho.	78
Figura 5.10: Influência do tamanho da malha na velocidade terminal para o caso 3.	78
Figura 5.11: Influência do tamanho da malha na forma final da bolha.	79
Figura 5.12: Evolução temporal da velocidade terminal adimensional da bolha com as três metodologias.	79
Figura 5.13: Comparação das formas das bolhas com as três metodologias estudadas e à consultada na bibliografia.	80
Figura 5.14: Comparação da forma final e números adimensionais reportados na literatura e na presente simulação.	82
Figura 5.15: Número de Reynolds da bolha em função do tempo adimensional para $Eo=339$, $Mo=43,1$ e $\mu_l/\mu_g = 100$	83
Figura 5.16: Número de Reynolds da bolha em função do tempo adimensional para $Eo=339$, $Mo=43,1$ e $\rho_l/\rho_g = 1000$	84
Figura 5.17: Campo de velocidades sobre a bolha para $Eo=339$, $Mo=43,1$, $\rho_l/\rho_g = 100$ com (a) $\mu_l/\mu_g = 5$ e (b) $\mu_l/\mu_g = 100$	84
Figura 5.18: Dimensões principais do domínio computacional e da gota.	86
Figura 5.19: Domínio computacional gerado com o arquivo <i>blockMeshDict</i> e a ferramenta <i>snappyHexMesh</i>	87
Figura 5.20: Forma da interface no tempo $t = 0,02$ s com valores do fator de compressão: (a) $C_\alpha = 1$ e (b) $C_\alpha = 3$	88
Figura 5.21: Vetores velocidade sobre a zona de distorção da interface para os três valores de C_α estudados.	89
Figura 5.22: Campo de velocidades sobre a gota para um caso com $Ca = 0,04$, $l_o/w =$ $1,87$ para (a) $Co = 0,2$ e (b) $Co = 0,5$	90
Figura 5.23: Campo dos vetores velocidade no momento da ruptura com a metodologia S- CLSVOF	91
Figura 5.24: Campo dos vetores velocidade no momento da ruptura com a metodologia VOF.	91
Figura 5.25: Regimes de ruptura de gotas em microcanais com junção em T	92

Figura 5.26: Vetores velocidade instantâneos para $Ca = 0,01$ e $l_o/w = 2,3$.-----	93
Figura 5.27: Espessura da gota (δ) durante a ruptura no regime RNO.-----	93
Figura 5.28: Evolução temporal da espessura da gota durante sua ruptura em comparação com um caso similar do trabalho de Fu et al. (2011).-----	94
Figura 5.29: Evolução temporal da espessura da gota δ/w , para $Ca = 0.09$ no regime RO para diferentes relações de l_o/w .-----	94
Figura 5.30: Evolução temporal da espessura da gota δ/w , para $Ca = 0.09$ no regime RNO para diferentes relações de l_o/w .-----	95
Figura 5.31: Modelo matemático obtido neste trabalho e modelo proposto por Leshansky et al. (2012) com $Ca = 0,09$ e $l_o/w = 2,9$.-----	96

LISTA DE TABELAS

Tabela 4.1: Solvers multifásicos disponíveis no OpenFOAM.-----	65
Tabela 5.1: Propriedades físicas dos fluidos para a primeira etapa da simulação. -----	72
Tabela 5.2: Tamanho dos domínios computacionais.-----	76
Tabela 5.3: Valores obtidos da velocidade terminal.-----	77
Tabela 5.4: Propriedades dos fluidos para a segunda e terceira etapa da simulação.-----	81
Tabela 5.5: Propriedades físicas dos fluidos.-----	87
Tabela 5.6: Tempo de processamento para os diferentes números de Capilaridade.-----	87

Lista de Símbolos

Letras Latinas

- \mathcal{A} Matriz com coeficientes diagonais, sem dimensões.
- \mathbf{b} Vetor do termo fonte referente ao centro dos volumes, sem dimensões.
- C_α Constante na Eq. (3.11), sem dimensões.
- d Distância entre o centro de dois volumes vizinhos (m).
- D Diâmetro da bolha (m).
- \mathcal{D} Matriz com coeficientes diagonais, sem dimensões.
- E Variável tensorial genérica, sem dimensões.
- F Variável tensorial genérica, sem dimensões.
- F_s Força de tensão superficial por unidade de volume (Pa/m).
- g Módulo do vetor gravidade (m/s²).
- \mathbf{g} Vetor gravidade (m/s²).
- h Altura do líquido (m).
- I Variável tensorial genérica, sem dimensões.
- H Função de *Heaviside*, sem dimensões.
- k Curvatura da interface (m)
- l_o Comprimento da gota no momento em que ela chega na junção (m).
- L_o Comprimento inicial da gota (m).
- M Variável tensorial genérica, sem dimensões.
- \mathbf{n} Vetor unitário normal à interface, sem dimensões.
- N Ponto central de um volume de controle, sem dimensões.
- \mathcal{N} Matriz com coeficientes não-diagonais, sem dimensões.
- p Pressão (Pa).
- p_{rgh} Pressão dinâmica (Pa).

P	Ponto central de um volume de controle, sem dimensões.
Q_g	Vazão do gás ($\mu\text{L}/\text{min}$).
S	Vetor elemento de área (m^2).
$S(\phi_0)$	Função distância, sem dimensões.
$S(\varphi)$	Termo fonte na Eq. (4.1), sem dimensões.
t	Tempo (s).
\mathbf{u}	Vetor velocidade (m/s).
\mathbf{u}_c	Vetor velocidade da fase contínua (m/s).
\mathbf{u}_d	Vetor velocidade da fase dispersa (m/s).
\mathbf{u}_r	Vetor velocidade relativa (m/s).
U_b	Velocidade baricêntrica da bolha (m/s).
U_b^*	Velocidade adimensional baricêntrica da bolha, sem dimensões.
w	Largura e altura do microcanal (m).
W	Fator na Eq. (4.9), sem dimensões.
X_1, X_2, X_3, X_4, X_5 e X_6	Constantes na Eq. (5.1), sem dimensões.

Letras Gregas

α	Fração volumétrica, sem dimensões.
α'	Constante na Eq. (5.3), sem dimensões.
ϵ	Espessura da interface (m).
ε	Diferença porcentual entre o Reynolds optido e o consultado na literatura (%).
μ	Viscosidade absoluta ($\text{kg}/\text{m s}$).
μ_c	Viscosidade da fase contínua ($\text{kg}/\text{m s}$).
μ_d	Viscosidade da fase dispersa ($\text{kg}/\text{m s}$).
ρ	Massa específica, (kg/m^3).

ρ_c	Massa específica da fase contínua, (kg/m ³).
ρ_d	Massa específica da fase dispersa, (kg/m ³).
$\boldsymbol{\tau}$	Tensor das tensões viscosas, (N/m ²).
$\Delta\tau$	Intervalo de tempo artificial, (s).
τ^*	Tempo adimensional, sem dimensões.
σ	Coefficiente da tensão superficial, (Pa m).
δ	Espessura mínima da gota (m).
δt^n	Novo intervalo do tempo (s).
δt^0	Tempo antigo (s).
$\delta(x)$	Função delta de Dirac, sem dimensões
ϕ	Função <i>Level Set</i> (curva de nível), sem dimensões.
ϕ_0	Função <i>Level Set</i> inicial, sem dimensões.
ϕ_{corr}	Número de iterações, sem dimensões.
γ	Variável tensorial genérica, sem dimensões.
φ	Variável tensorial genérica, sem dimensões.
$\boldsymbol{\varphi}$	Vetor das variáveis referente ao centro dos volumes, sem dimensões.
Γ	Constante na Eq. (3.16), sem dimensões.
Ψ	Coefficiente de difusão (m ² /s).
Δx	Tamanho da malha, (m).

Índices

b	Baricêntrica
c	Fase contínua
d	Fase dispersa
l	Líquido
g	Gás

o Inicial
T Transposto

Números Adimensionais

Bo Número de Bond.
Ca Número de capilaridade.
Co Número de Courant.
*Co*⁰ Número de Courant adaptativo.
*Co*_{max} Número de Courant máximo.
Eo Número de Eötvös.
Mo Número de Morton.
Re Número de Reynolds.

Lista de Abreviaturas

ACLSVOF Acoplamento VOF e LS (Algoritmo de Yang *et al.* (2006)).
ASM *Algebraic Slip Model.*
CFD *Computational Fluid Dynamics* (Dinâmica dos Fluidos Computacional).
CLSVOF Acoplamento VOF e LS (Algoritmo de Sussman e Puckett (2000)).
CSF Continuum Surface Force.
CSS Continuous Surface Stress.
CST *Continuum Surface Tension.*
DPD *Dissipative Particle Dynamics.*
DPM *Dispersed Phase Model.*
EGMM *Eulerian Granular Multiphase Model*
EMM *Eulerian Multiphase Model.*
FI Fronteira imersa.
FVC *Finite Volume Calculus.*
FVM *Finite Volume Method.*
FOAM *Field Operation and Manipulation.*

LAC-UNIFEI	Laboratório de Astrofísica Computacional – UNIFEI.
LBM	<i>Lattice Boltzmann Model</i>
LS	<i>Level Set</i> (Curva de nível).
LWENO	<i>Linear weighted essentially non-oscillatory scheme</i>
MD	<i>Molecular Dynamics.</i>
NR	Não ruptura.
PIMPLE	Acoplamento PISO e SIMPLE
PISO	<i>Pressure-Implicit with Splitting of Operators.</i>
PLIC	<i>Piecewise Linear Interface Calculation.</i>
PROST	<i>Parabolic Reconstruction of Surface Tension.</i>
RO	Ruptura obstruída.
RNO	Ruptura não obstruída.
SAMR	<i>Structured Adaptative Mesh Refinement).</i>
S-CLSVOF	Acoplamento VOF e LS (Algoritmo de Albadawi <i>et al.</i> (2013))
SIMPLE	<i>Semi-Implicit Method for Pressure-Linked Equations.</i>
SLIC	<i>Simple Line Interface Calculation.</i>
SPH	<i>Smoothed Particle Hydrodynamics.</i>
VOF	<i>Volume of Fluid</i> (Volume de Fluido).
VTK	<i>Visualization toolkit.</i>

Sumário

CAPÍTULO 1 - INTRODUÇÃO	21
1.1 Contexto e Motivação	21
1.2 Objetivos	23
1.3 Justificativa e Relevância	23
1.4 Desenvolvimento do trabalho	25
CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA	26
2.1 Introdução aos escoamentos bifásicos	26
2.1.1 Ascensão de bolhas através de líquidos	28
2.1.2 Escoamentos bifásicos em Microcanais	31
2.2 Modelagem de escoamentos Multifásicos.	35
2.3.1 Método Volume de Fluido (VOF)	38
2.3.2 Método Level Set (LS)	41
2.3.3 Acoplamento dos Métodos Level Set - VOF (CLSVOF)	44
CAPÍTULO 3 – FUNDAMENTOS MATEMÁTICOS	46
3.1 Introdução	46
3.2 Equações de Navier-Stokes	46
3.2.1 Formulação da metodologia VOF	47
3.2.2 Tensão Superficial	48
3.2.3 Acoplamento VOF com LS (CLSVOF)	49
3.3 Condições iniciais e de contorno.	52
3.4 Números adimensionais relevantes	53
CAPÍTULO 4 – MÉTODO NUMÉRICO	55
4.1 Fluidodinâmica Computacional	55
4.2 Principais características do OpenFOAM	57
4.3 Discretização por Volumes Finitos	59
4.4 Acoplamento pressão-velocidade	63
4.5 Solvers disponíveis no OpenFOAM	64
4.6 Estrutura de um caso no OpenFOAM	66
4.7 Descrição dos solvers interFoam e interDyMFoam	67
4.8 Implementação do acoplamento VOF-LS no OpenFOAM	69

4.9 Pós-processador ParaView-----	69
CAPÍTULO 5 – RESULTADOS E DISCUSSÕES _____	71
5.1 Ascensão de uma bolha-----	71
5.1.1 Efeito dos parâmetros de refinamento. _____	72
5.1.2 Efeito das dimensões do domínio computacional, do tamanho da malha e das condições de contorno nas paredes. _____	75
5.1.3 Efeito da variação dos números adimensionais _____	80
5.1.4 Efeito das razões de massa específica e viscosidade _____	82
5.2 Transporte de uma gota em um microcanal -----	85
5.2.1 Definição da geometria e malha. _____	85
5.2.2 Estudos dos parâmetros da simulação. _____	88
5.2.3 Mecanismos de ruptura/não ruptura de gotas com junções em T. _____	92
CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS _____	98
6.1 Conclusões -----	98
6.2 Trabalhos Futuros. -----	99
REFERÊNCIAS BIBLIOGRÁFICAS _____	100
APÊNDICES. _____	105
Apêndice 1-----	105
1.1 Arquivos que compõem cada pasta de um caso em OpenFOAM -----	105
Apêndice 2-----	107
2.1 Acesso ao código fonte -----	107
Apêndice 3-----	110
3.1 Refinamento dinâmico com o Solver interDyMFoam e arquivos do código fonte relacionados com sua funcionalidade -----	110
Apêndice 4-----	112
4.1 Explicação das funcionalidades dos arquivos do código fonte do solver interFoam-----	112
4.2 Arquivos adicionais no caso do solver S-CLSVOF. -----	112
ANEXOS _____	114
ANEXO A-----	114
A.1 Arquivo dynamicMeshDict -----	114
ANEXO B-----	116
A.1 Fotografia do Cluster do Laboratório de Astrofísica Computacional da UNIFEI-	116
ANEXO C -----	117

C.1 Arquivo controlDict da pasta system.-----	117
ANEXO D-----	118
D.1 Arquivo snappyHexMesh -----	118
ANEXO E -----	125
E.1 Resultados apresentados por Hoang <i>et al.</i> (2013)-----	125

Capítulo 1

Introdução

1.1 Contexto e Motivação

As aplicações envolvendo escoamentos multifásicos estão presentes em muitos ramos da engenharia. Entre estes podem ser citados: a indústria de petróleo, a indústria química, equipamentos de refrigeração como condensadores, evaporadores e torres de resfriamento, misturadores siderúrgicos, bombeamento de água por processo de elevação por ar, plantas de potência, reatores nucleares, entre outros.

Bolhas de um gás ascendendo num meio líquido (Fig. 1.1a) e gotas caindo ou a colisão entre elas (Fig. 1.1b), são exemplos deste tipo de escoamento. A ascensão de uma bolha de gás em um líquido viscoso é geralmente acompanhada por deformações significativas, indicando uma interação complexa e uma forte dependência das propriedades dos fluidos. Vários estudos numéricos e experimentais abordam este problema e são ainda muito utilizados como referência, o que mostra que a ciência ainda tem muito a evoluir.

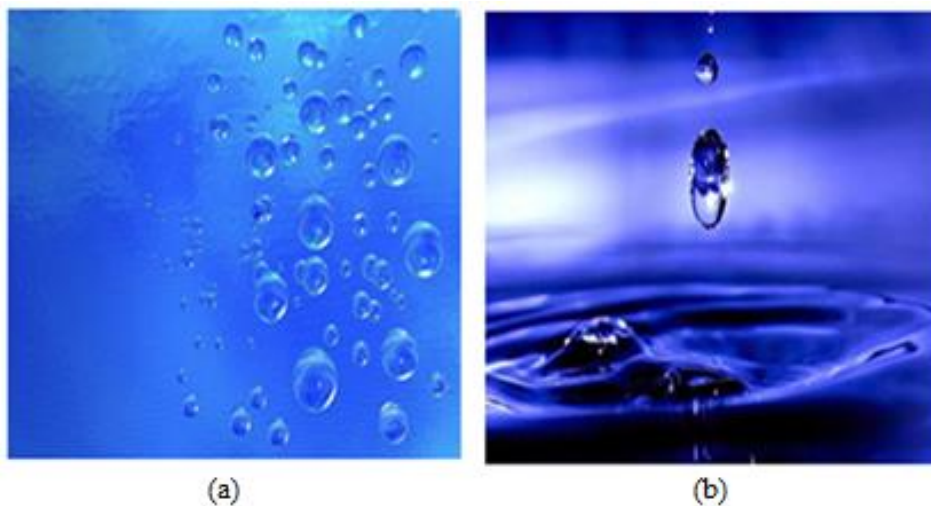


Figura 1.1: Exemplos de escoamentos bifásicos. (a) bolhas de ar ascendendo num líquido, (b) colisão entre duas gotas. Fonte: <https://www.google.com.br/search?q=imagenes+de+escoamentos+multifasico>.

Outras aplicações são encontradas em tecnologias de microsistemas como equipamentos biomédicos, bioquímicos e em microreatores de escoamento segmentado, como mostrado na Figura 1.2. No dispositivo microfluidico apresentado na Figura 1.2(a) as gotas escoam através de uma junção em T, após um canal lateral contendo uma segunda solução aquosa ao ser injetada nas gotas. Na Figura 1.2(b) é apresentado um processo de separação de bolhas em uma junção em T. Estes dispositivos de dimensões micro e até nanométricas podem

ser usados para diversas aplicações como na síntese química fina e em microrreatores para a indústria química e bioquímica. O projeto desses microsistemas requer uma compreensão completa e uma visão mecanicista do fenômeno em questão.

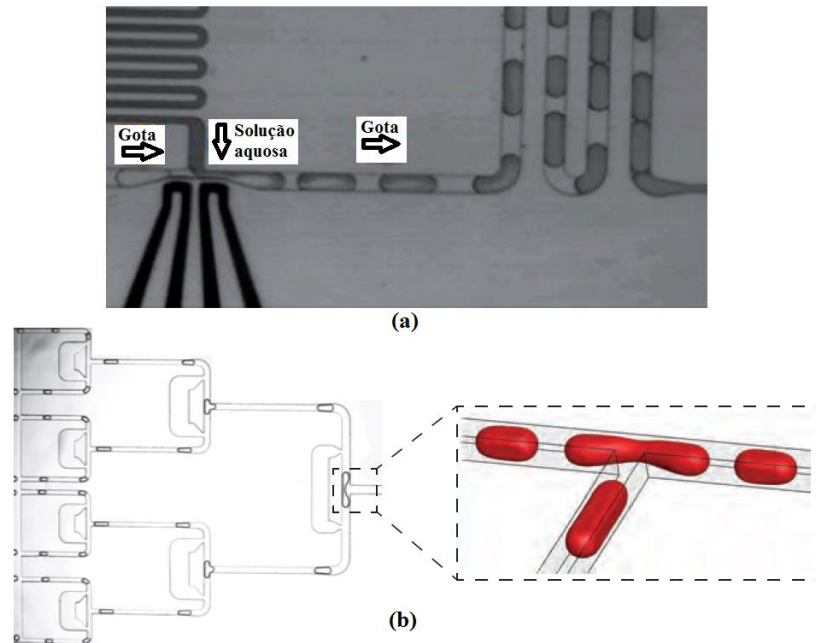


Figura 1.2: Dispositivos com escoamentos bifásicos. (a) Ilustração de um dispositivo microfluídico (GUO *et al.* 2012). (b) Ilustração de um distribuidor de bolhas (HOANG, 2012).

A Dinâmica dos Fluidos Computacional (CFD) como ferramenta para estudar o comportamento desses tipos de escoamentos bifásicos (bolhas ascendendo em meios líquidos e gotas escoando em microcanais) pode fornecer informações detalhadas sobre a física desses fenômenos. Em ambos os tipos de escoamentos bifásicos é preciso modelar com precisão a interface entre os fluidos.

Dentre as metodologias numéricas existentes, o presente trabalho deu enfoque ao uso das técnicas VOF (Volume de Fluido) e CLSVOF (acoplamento do *Level Set* e VOF). A primeira, VOF, já implementada no OpenFOAM baseia-se no cálculo da fração volumétrica de uma das fases presentes no escoamento. Em alguns problemas nos quais a interface se alonga, como no processo de ruptura de gotas/bolhas, esta metodologia pode gerar resultados não físicos devido à dificuldade de representar corretamente a interface. Desta forma, a principal motivação do trabalho foi testar a técnica CLSVOF para os mesmos problemas, buscando identificar as vantagens e desvantagens de cada uma das duas metodologias.

As técnicas VOF e *Level Set* (LS) foram revisadas para o estudo de escoamentos bifásicos e foi feita uma ampla revisão bibliográfica do estado da arte envolvendo as duas

técnicas. Além disso, é feita uma breve introdução sobre Fluidodinâmica Computacional e sobre o *software* livre, OpenFOAM.

1.2 Objetivos

O objetivo geral deste trabalho foi estudar problemas envolvendo o transporte de bolhas/gotas em fluidos newtonianos por meio das metodologias VOF e CLSVOF usadas para o estudo de escoamentos bifásicos com a utilização do código aberto OpenFOAM.

Dentre os objetivos específicos podem ser citados:

- Realizar uma ampla revisão das metodologias numéricas VOF e LS empregadas para analisar escoamentos envolvendo o transporte de bolhas e gotas.
- Entender a estrutura e as funcionalidades já disponíveis no código OpenFOAM com a metodologia VOF afim de obter dados para a implementação do acoplamento VOF e LS no código OpenFOAM.
- Implementar no código computacional, modelos ou outras funcionalidades necessárias para representar o problema completo. Definir o tipo de malha numérica, condições iniciais e de contorno, escolher os métodos de discretização espacial e temporal dentre outros.
- Validar a metodologia VOF implementada no OpenFOAM com simulações da ascensão de uma bolha num meio líquido e comparar os resultados com a literatura.
- Implementar no OpenFOAM a metodologia CLSVOF com o objetivo de estudar o transporte de bolhas em microcanais e estudar os mecanismos de ruptura.
- Analisar os resultados das simulações numéricas e comparar com resultados da literatura.

1.3 Justificativa e Relevância

Os escoamentos bifásicos ocorrem em uma ampla gama de situações, como processos biológicos e industriais, e nas últimas décadas numerosos pesquisadores abordaram estes estudos tanto teórica quanto experimentalmente. Um exemplo fundamental é o caso de uma bolha de gás isolada que se eleva e se deforma em um líquido viscoso. Um conhecimento deste sistema básico é de grande importância para entender escoamentos mais complexos, como os escoamentos borbulhantes. Outro exemplo de interesse são as gotas confinadas no interior de microcanais escoando através de junções em T, dispersas em outra fase contínua. O interesse em manipular gotas em microcanais surgiu de duas motivações distintas mas complementares. A primeira é o desejo de produzir gotas calibradas para aplicações em ciência de materiais, nas

indústrias farmacêuticas e alimentares. A segunda tem relação com aplicações em microfluídica digital, um tema inteiramente dedicado ao tratamento de gotas em microsistemas. Do ponto de vista fluidodinâmico, a microfluídica digital trata de interfaces móveis que adquirem dinâmicas muitas vezes complexas complexas.

Nas últimas décadas as técnicas em CFD (*Computational Fluid Dynamics*) com novas ferramentas numéricas e computacionais, tornaram possível ampliar o estudo do comportamento dinâmico desses tipos de escoamentos bifásicos. A escolha correta de um código CFD como ferramenta para analisar um determinado fenômeno requer muitas considerações tais como a facilidade de uso, a capacidade de representar a geometria, a eficiência e robustez das técnicas numéricas disponíveis e a diversidade dos problemas físicos que o código pode resolver. Atualmente diferentes *softwares* em CFD estão disponíveis, como por exemplo os comerciais CFX e FLUENT, ambos da ANSYS e PHOENICS, da CHAM. Contudo, em muitos casos, o custo de adquirir uma licença inviabiliza sua escolha, principalmente para a realização de trabalhos de pesquisa acadêmicos. Outras limitações dos *softwares* comerciais refere-se à falta de detalhes dos módulos existentes e a impossibilidade de efetuar novas implementações.

Existem também códigos em CFD gratuitos disponíveis, cuja maioria foi desenvolvida no meio acadêmico. O *software* de código livre OpenFOAM utilizado neste trabalho, surgiu como uma opção viável e muito interessante para o presente estudo. Esse *software* reúne características como a possibilidade de implementar uma nova metodologia para estudar classes de escoamentos, diferentes métodos de discretização temporal e espacial disponíveis, assim como a existência de ferramentas de geração de malhas, refinamento local e dinâmico. Além disso, modelos de turbulência podem ser testados e adequados para cada estudo. Neste contexto, além do fato do código ser aberto/livre, todos estes fatores contribuíram para sua escolha no presente trabalho.

As primeiras simulações feitas no presente estudo foram conduzidas com a metodologia Volume de Fluido que já estava implementada no código. O objetivo dessas primeiras simulações foi validar o código com a bibliografia consultada. Em seguida foi implementada a metodologia S-CLSVOF de Albadawi *et al.* (2013) e foram posteriormente comparadas ambas metodologias especialmente no estudo dos mecanismos de ruptura/não ruptura de gotas escoando através de microcanais com junções em T.

1.4 Desenvolvimento do trabalho

No capítulo 2 são citados trabalhos numéricos desenvolvidos para o estudo de escoamentos envolvendo principalmente o transporte de bolhas/gotas. A revisão bibliográfica foi feita com o intuito de se conhecer as diferentes metodologias mais utilizadas atualmente, assim como para coletar dados para comparação com o presente trabalho.

No capítulo 3 a modelagem matemática utilizada no presente estudo, as equações da metodologia VOF assim como a formulação do acoplamento Volume de Fluido e *Level Set* são apresentadas.

No capítulo 4 a metodologia numérica utilizada no presente trabalho é apresentada, destacando o uso do OpenFOAM e às técnicas de discretização, de geração de malha, acompanhamento e reconstrução da interface e aos procedimentos de solução das equações de conservação.

No capítulo 5 são apresentados os resultados numéricos obtidos neste trabalho, relacionados com a ascensão de uma bolha em um meio viscoso e a dinâmica de uma gota escoando através de um microcanal com junção em T . Ambos estudos numéricos usaram as metodologias VOF e S-CLSVOF, estabelecendo comparações entre as duas técnicas e com resultados numéricos e experimentais da literatura consultada.

Finalmente, são discutidas as principais conclusões do presente trabalho e são propostas possibilidades para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, apresenta-se uma revisão da literatura sobre os fundamentos de escoamentos bifásicos e as técnicas de modelagem utilizadas neste tipo de problema. Os temas abordados tratam, da ascensão de bolhas em meios líquidos e os escoamentos bifásicos em microcanais. É apresentada uma breve descrição das principais técnicas de modelagem, com ênfase nas baseadas em malhas, com captura da interface e os métodos Volume de Fluido (VOF) e *Level Set* (LS).

2.1 Introdução aos escoamentos bifásicos

Os escoamentos bifásicos são fontes de inúmeros processos não-lineares de relevância científica e tecnológica. Esses escoamentos são observados tanto em processos industriais quanto na natureza. São citados como exemplos a cavitação em turbinas e bombas hidráulicas, processos de fabricação de papel e plásticos, extração de petróleo, aspersão de líquidos no ar como venenos e aerossóis, fluxo de sangue e dispersão de poluentes na atmosfera.

Em diversas indústrias, incluindo de processos, energia, aeroespacial e transporte, os escoamentos em duas ou mais fases são práticas comuns. Em outros casos eles são indesejados, mas simplesmente podem não ser possíveis ou economicamente viáveis de evitar. Portanto, a análise e compreensão dos escoamentos em duas fases são de suma importância e os processos envolvendo tais escoamentos devem ser projetados e controlados de forma segura.

Estes escoamentos são caracterizados por um movimento complexo da interface que separa as massas dos fluidos presentes e pode sofrer deformações significativas e transições topológicas. Geralmente, são amplamente categorizados pelos estados físicos dos componentes constituintes e pela topologia das interfaces. Assim, um escoamento de duas fases pode ser classificado como gás-sólido, gás-líquido, líquido-sólido ou no caso de dois líquidos não miscíveis, líquido-líquido. Da mesma forma, um escoamento pode ser amplamente classificado topologicamente como separado, disperso ou de transição (BERGLES *et al.* 1981 e HEWITT, 1982). Esta variedade de combinações torna o projeto de equipamentos industriais para aplicações de escoamento em duas fases uma tarefa complexa.

A previsão do comportamento de um escoamento bifásico é extremamente complexa, especialmente devido às interações entre as fases. As fases podem se arranjar em várias configurações ou padrões. A ocorrência de um determinado padrão de escoamento depende das

características geométricas do sistema de transporte de fluido, de variáveis operacionais e das propriedades físicas das substâncias que compõem a mistura bifásica.

A necessidade de modelar e prever o comportamento destes escoamentos é um tema persistente em muitos trabalhos de pesquisa. Existem três maneiras pelas quais esses modelos são explorados (BRENNEN, 2005): experimentalmente, através de modelos de tamanho laboratorial equipados com instrumentação, analiticamente usando equações matemáticas e modelos de escoamentos, ou computacionalmente, usando as técnicas CFD e modelos numéricos para prever a posição e forma da interface nos fluidos. Muitos projetos de equipamentos sujeitos a escoamentos bifásicos ainda dependem, em grande parte, de correlações empíricas deduzidas a partir de resultados de medições experimentais realizadas em condições mais próximas possíveis das operacionais (FACCINI, 2008). Por outro lado, como as correlações empíricas foram obtidas em laboratório, têm a desvantagem de que essa informação experimental possa esconder informações relacionadas ao processo industrial.

Conseqüentemente, a capacidade preditiva e a compreensão física dependem fortemente de modelos teóricos sendo a complexidade da maioria dos escoamentos bifásicos, um grande obstáculo. É possível codificar as equações da quantidade de movimento para cada uma das fases ou componentes e calcular todos os detalhes de cada uma das fases, mas a capacidade de processamento e a velocidade dos computadores necessários são muito elevadas e nem sempre estão disponíveis para o estudo da maioria dos escoamentos que são comumente experimentados. Quando o escoamento de uma ou ambas as fases se torna turbulento o que é bastante comum, a magnitude do desafio aumenta ainda mais. Portanto, as simplificações são muitas vezes também necessárias nas soluções tanto analíticas quanto numéricas.

A turbulência em escoamentos bifásicos chama a atenção não somente pelo fenômeno de formação de esteiras turbilhonares, mas também pelo fato de que a turbulência na fase contínua afeta a dinâmica da fase dispersa (bolhas/gotas). A turbulência também pode levar à deformação por cisalhamento, fragmentando assim uma bolha/gota e criando várias outras bolhas/gotas satélites como mostrado na Figura 2.1. Outro fenômeno observado em escoamentos bifásicos turbulentos é por exemplo, um jato de líquido que é quebrado em gotas aparentemente microscópicas, conhecido como atomização.



Figura 2.1: Fragmentação de uma gota em uma superfície livre. Fonte: Tryggvason, Scardovelli e Zaleski (2011).

Os escoamentos bifásicos que se desenvolvem em regime laminar são mais comuns em dispositivos microfluídicos. O estudo do comportamento do escoamento bifásico em dispositivos microfluídicos ganhou destaque, visto que o entendimento de suas características é fundamental para o projeto e aplicação de aquecimento/resfriamento, mistura, reações e emulsões. Essa área de pesquisa é um campo interdisciplinar de rápido crescimento com várias aplicações em bioquímica, biotecnologia e processos químicos em geral. Atualmente, já estão disponíveis sistemas capazes de separar, detectar ou filtrar substâncias, misturadores de soluções, microválvulas, microbombas e sistemas para obtenção de partículas e outros produtos em escala reduzida (WHITESIDES, 2006).

Neste trabalho foram simulados escoamentos laminares bifásicos gás-líquido e líquido-líquido e em específico estudou-se a ascensão de bolhas num meio líquido estagnado e limitado por paredes e o transporte de gotas dentro de microcanais. A seguir são descritos alguns trabalhos utilizados como referência que também auxiliaram na compreensão do estado da arte deste tema.

2.1.1 Ascensão de bolhas através de líquidos

Um exemplo clássico de escoamento bifásico é o de uma bolha de gás isolada ascendendo e deformando-se num líquido estacionário. As colunas de bolhas são reatores

multifásicos onde o gás, que constitui a fase dispersa, é distribuído no fundo da coluna e sobe através do líquido que constitui a fase contínua (estacionária). Estes equipamentos são utilizados em muitos processos industriais importantes que envolvem sistemas gás-líquido nas áreas químicas (por exemplo, fabricação de combustíveis sintéticos), biológicas (por exemplo fermentadores) e petroquímicas (por exemplo, liquefação com carvão). As colunas de bolhas apresentam vantagens sobre outros reatores multifásicos, que incluem baixo preço, construção simples, baixos custos operacionais, mistura eficiente e alta eficiência de transferência de massa e calor (DIAZ *et al.*, 2008). Apesar das extensas aplicações industriais das colunas de bolhas, ainda existem algumas questões não respondidas, basicamente devido ao conhecimento parcial da dinâmica do escoamento gás-líquido.

Existem trabalhos experimentais e numéricos envolvendo bolhas ascendendo num líquido (BHAGA e WEBER, 1981; RAYMOND e ROSANT, 2000; TOMIYAMA *et al.*, 2002) desenvolvidos há mais de 15 anos, que ainda são tomados como referência, porém, existem aspectos importantes que ainda permanecem incertos no comportamento de uma única bolha ascendendo em meios líquidos.

Em geral a ascensão de uma bolha no líquido é uma função de vários parâmetros tais como a característica geométrica da bolha (tamanho e forma), as propriedades dos sistemas gás-líquido (viscosidade, tensão superficial, concentração do soluto, diferença de massa específica entre gás e líquido), movimento do líquido (direção) e condições de operação (temperatura, pressão, gravidade). As paredes do recipiente que as contém pode também afetar a dinâmica da bolha assim como sua forma final (CLIFT e GRACE 1978). A Figura 2.2 é conhecida como diagrama de Clift e mostra a forma de bolhas e gotas movendo-se em meios líquidos estacionários, em função dos números adimensionais de Reynolds (Re), Eötvös (Eo) e Morton (Mo) (definidos no Cap. 3). Um dos objetivos deste trabalho foi também verificar a forma final de uma bolha em ascensão, utilizando as duas metodologias numéricas mencionadas, VOF e S-CLSVOF, e comparar os resultados com o diagrama de Clift.

Observa-se na Figura 2.2 que, quando as bolhas são pequenas e a tensão superficial é alta ou seja, baixos números de Reynolds e Eötvös ou também baixos Re e Eo altos, as bolhas permanecem esféricas e a dinâmica da bolha se torna independente da tensão superficial (ESMAEELI E TRYGGVASON, 1998). Para altos valores de Morton as bolhas se tornam elipsoidais à medida que Reynolds aumenta. Um aumento também de Eo faz com que as bolhas se deformem e assumam a forma de “calotas esféricas”. Para Morton moderados, as bolhas apresentam forma helicoidal. Para altos valores de Eo e Mo elas podem sofrer grandes

deformações e até se fragmentar. O número de Reynolds é geralmente calculado com base na velocidade terminal da bolha e desta forma só pode ser determinado após o regime terminal ser atingido, no qual a aceleração é nula.

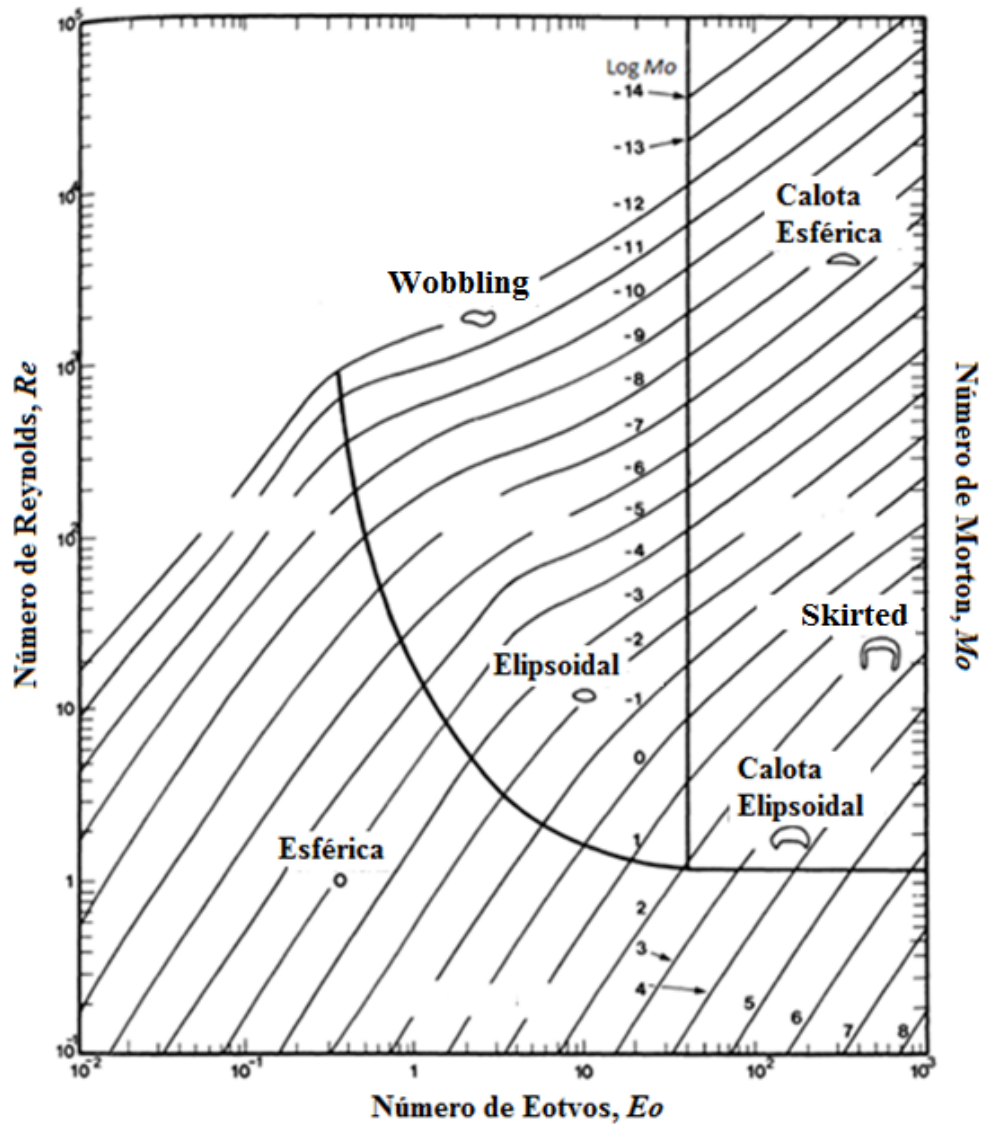


Figura 2.2: Diagrama de Clift. Forma das bolhas e gotas movendo-se através de líquidos. Fonte: adaptado de Clift e Grace (1978).

Algumas limitações nos métodos experimentais tornam necessária a investigação numérica, por exemplo, é difícil obter informações completas sobre o campo de velocidade e a distribuição de pressão em todo o domínio. Em geral, os métodos experimentais estão limitados aos casos com formas de bolhas simples. As simulações numéricas permitem estudar bolhas com formas diversas. Por exemplo, o efeito da forma inicial da bolha em sua ascensão, bem como o efeito das razões de baixa massa específica e viscosidade, ou regimes de números de Reynolds baixos são difíceis de se estudar experimentalmente.

Neste contexto, o uso da dinâmica computacional de fluidos (CFD) torna-se uma ferramenta essencial, pois pode prever e descrever a hidrodinâmica em escala local e global. Nesse sentido, vários estudos de escoamentos multifásicos utilizando ferramentas CFD foram publicados nos últimos 10 anos. No entanto, vários problemas de modelagem ainda precisam ser resolvidos devido às complexas características hidrodinâmicas relacionadas com as interações interfaciais.

2.1.2 Escoamentos bifásicos em Microcanais

Escoamentos bifásicos em microcanais recentemente têm atraído a atenção de muitos pesquisadores devido à sua ampla aplicabilidade nas modernas e avançadas tecnologias tais como micro-eletrônica, engenharia de processos químicos, engenharia genética, bioengenharia, miniaturização de dispositivos em outros campos como na indústria aeroespacial e nuclear. Trocadores de calor implementados em equipamentos como supercomputadores, lasers e reatores nucleares também estão entre as aplicações atuais (PARK e PUNCH, 2008).

Várias características de escoamentos bifásicos em microcanais têm sido pesquisadas nas últimas décadas, visando por exemplo, analisar os padrões de escoamento (FUKANO e KARIYASAKI, 1993, TRIPLETT *et al.*, 1999a e HETSRONI *et al.*, 2003) e a queda de pressão (TRIPLETT *et al.*, 1999b). Trabalhos numéricos e experimentais sobre a dinâmica de bolhas e gotas em microcanais têm sido realizados por vários pesquisadores com o objetivo de estudar a condição de ruptura e não-ruptura (FU *et al.*, 2011; AFKHAMI, LESHANSKY e RENARDY, 2011 e HOANG *et al.*, 2013). O comportamento de ruptura e de não ruptura de bolhas é muito semelhante ao observado em gotas nos escoamentos em microcanais, segundo dados da literatura consultada (JULLIEN *et al.*, 2009).

Recentemente a tecnologia de escoamentos em microfluidos tem avançado como um novo campo de pesquisa e fornece uma nova abordagem para a preparação de emulsões micro dispersas controladas, que têm sido amplamente utilizadas nas reações químicas, bioensaios, processos de separação e de síntese de materiais. Emulsões monodispersas podem ser preparadas controladamente em microcanais com junções em T com dimensões entre 10 e 500 μm . O número de capilaridade Ca , o qual representa a relação entre os efeitos viscosos e as forças de tensão superficial, é o parâmetro mais importante nesse estudo. Estudos mostram que em microcanais com junções em T, a condição de ruptura/não-ruptura de gotas/bolhas é influenciada principalmente pelo número de capilaridade Ca e pela razão entre o comprimento

da bolha ou gota na junção (l_0) e a dimensão transversal do microcanal (w), ou seja l_0/w (HOANG *et al.* 2013). Na Figura 2.3 é ilustrado o processo de ruptura simétrica de uma gota num microcanal de seção retangular, com junção em T .

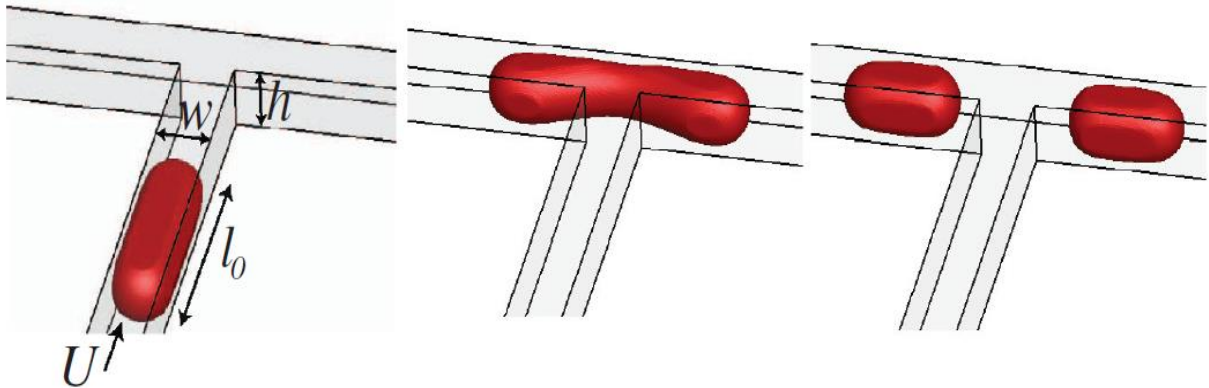


Figura 2.3: Ruptura de uma gota em um microcanal com junção em T . Fonte: Hoang *et al.* 2013.

Apesar de um alto interesse pelas técnicas microfluídicas baseadas em gotas, a compreensão teórica desta operação unitária é bastante limitada. Link *et al.* 2004 descrevem em seus estudos experimentais o processo de ruptura de gotas movimentando-se através de uma junção em T de seção transversal quadrada de $30 \times 30 \mu\text{m}$. Nesses experimentos os autores constataram que o mecanismo de ruptura/não ruptura é governado pela relação entre o número de capilaridade e o comprimento adimensional da gota definido pela relação $l_0/\pi w$. Na Figura 2.4 apresenta-se a linha de transição entre os mecanismos de ruptura/não ruptura.

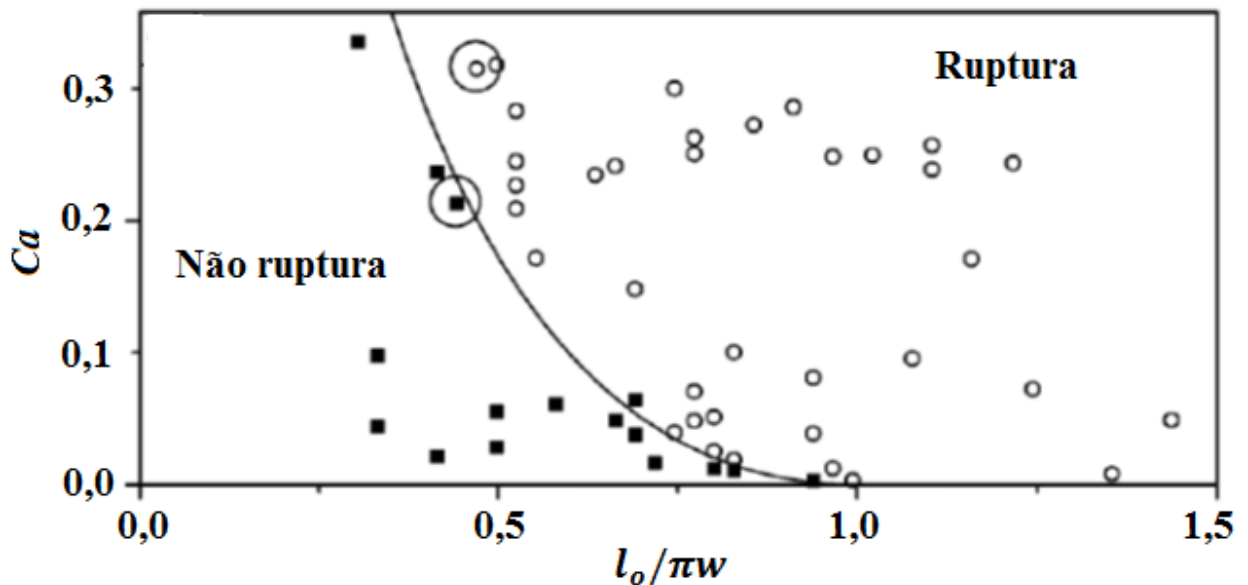


Figura 2.4: Diagrama representativo da ruptura de gotas em um microcanal com junção em T . A linha contínua indica a transição entre os mecanismos de ruptura/não-ruptura. Fonte: adaptado de Link *et al.* 2004.

Jullien *et al.* (2009) apresentam estudos experimentais sobre a ruptura de gotas em microcanais com junções em T em uma faixa de número de capilaridade de 4×10^{-4} a 2×10^{-1} e duas razões de viscosidades dos fluidos (μ_d/μ_c) de 0,11 e 1,67. Duas formas de ruptura são apresentadas neste trabalho: ruptura obstruída (ou bloqueada) e não obstruída. No caso da ruptura com obstrução é mantida uma distância muito pequena entre a parede e a gota durante a ruptura, comumente inferior ao de uma resolução visual, e no caso não obstruída essa distância entre a parede e a gota é visível. Na Figura 2.5 são apresentados os resultados mais importantes desta pesquisa, mostrando as três regiões: A (não ruptura ou NR), B (ruptura não obstruída ou RNO) e C (ruptura obstruída ou RO).

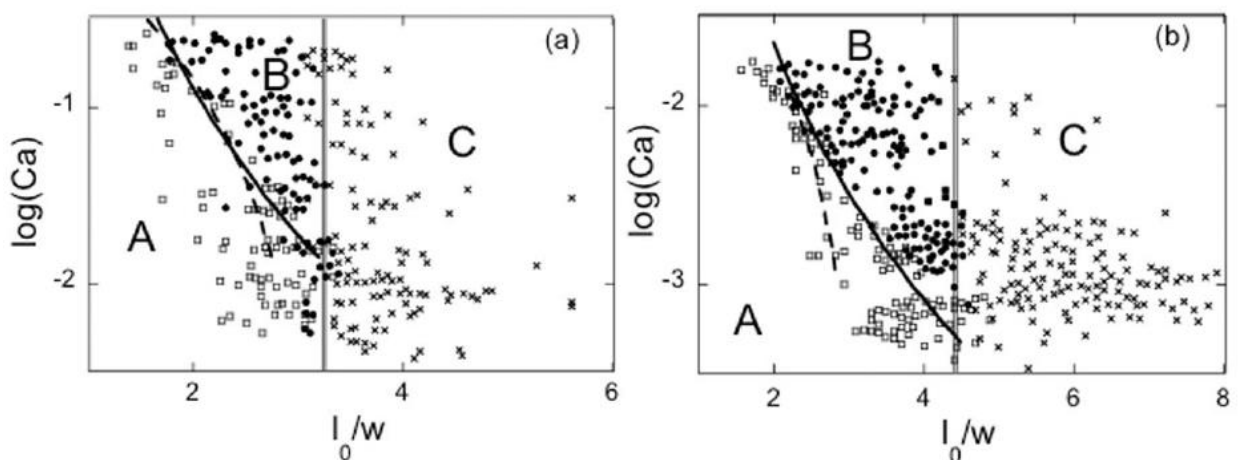


Figura 2.5: Diagramas que representam os comportamentos de gotas movimentando-se através de uma junção em T para duas razões de viscosidade: (a) $\mu_d/\mu_c = 0,11$ e (b) $\mu_d/\mu_c = 1,67$. Fonte: Jullien *et al.* (2009).

Fu *et al.* (2011), estudaram experimentalmente a ruptura de bolhas de Nitrogênio em misturas de glicerol-água com várias concentrações, dentro de um microcanal retangular de $400 \times 400 \mu\text{m}$ com junção em T , variando o número de capilaridade de 0,001 a 0,1. Quatro mecanismos de ruptura foram identificados, tipo I (RO), no qual a ruptura da bolha é controlada pelo aumento da pressão na fase líquida, no tipo II (RNO) é controlada tanto pelo aumento da pressão como pelas forças viscosas e no tipo III (RNO) no qual o mecanismo é controlado por uma lei com dependência entre a espessura da bolha durante a ruptura e o tempo restante durante o processo de ruptura. Este tipo de comportamento também foi relatado por Link *et al.* 2004 e eles argumentam que a ruptura pode ser atribuída à instabilidades de *Rayleigh-Plateau*. O quarto mecanismo é a condição de não ruptura (NR). Na Figura 2.6 é apresentado um gráfico que relaciona o número de capilaridade com a vazão do gás, onde pode-se apreciar estes quatro mecanismo de ruptura.

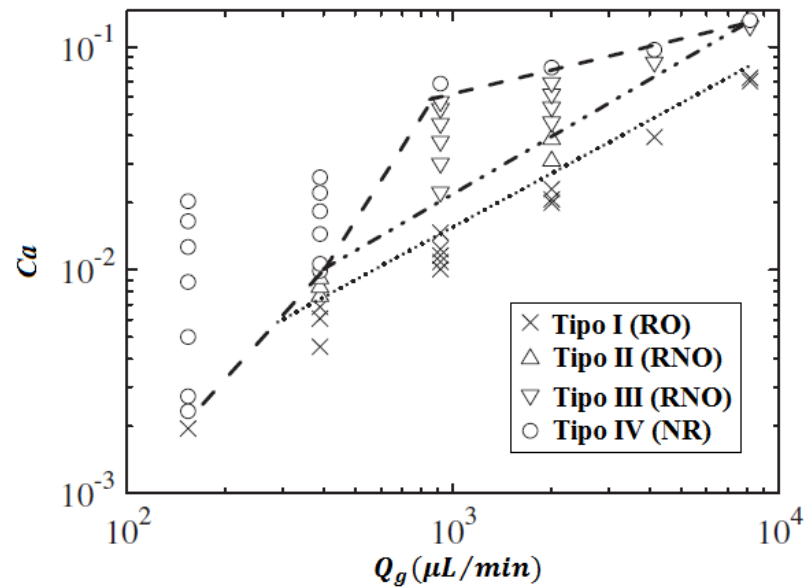


Figura 2.6: Diagramas de mecanismo de ruptura em junções em T. Fonte: Adaptado de Fu *et al.* (2011).

Leshansky *et al.* (2012) apresentam uma análise teórica e numérica da ruptura de gotas em junções em T. Com base em uma análise bidimensional são obtidas equações que relacionam os diferentes parâmetros geométricos com o número de capilaridade e foram traçadas a forma da interface durante a ruptura. Os resultados foram validados com simulações numéricas e com os resultados do trabalho de Jullien *et al.* (2009). Na Figura 2.7 são apresentados os resultados numéricos que predizem os mecanismo de ruptura/não ruptura de gotas através de junções em T em função do número de capilaridade e de l_0/w . Leshansky *et al.* (2012) apresentam nesse diagrama três zonas: NR, RNO e RO.

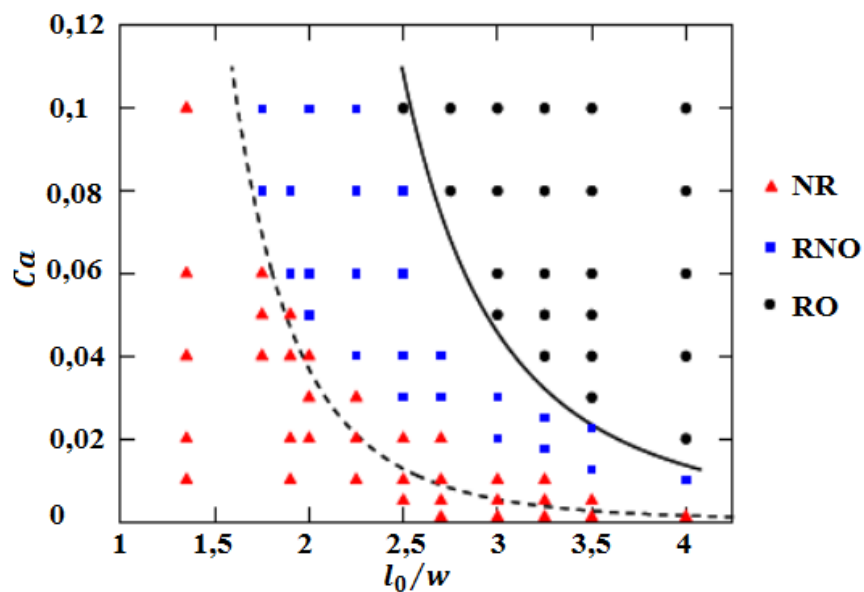


Figura 2.7: Diagrama representativo do mecanismo de ruptura de gotas em junções em T. Fonte: Adaptado de Leshansky *et al.* (2012).

As simulações numéricas de escoamentos bifásicos nestas geometrias muito pequenas são geralmente complexas. Forças capilares grandes, combinadas com o efeito de confinamento, ruptura de gotas e os mecanismos que provocam a separação são apenas alguns exemplos da complexidade neste tipo de simulações (AFKHAMI, LESHANSKY e RENARDY, 2011).

Uma das grandes dificuldades na simulação de gotas ou bolhas em microcanais é o fato de que em baixos números de capilaridade a interface tende a se deformar de forma não física devido à presença de correntes parasitárias. As correntes parasitárias são movimentos não físicos do fluido gerados ao usar modelos, como as técnicas CSF (*Continuum Surface Force*), para aproximar o cálculo da tensão superficial na interface (RENARDY e RENARDY, 2002). Sua magnitude geralmente aumenta com o aumento da força de tensão superficial e pode se tornar tão grande que afeta a predição do campo de velocidade ou em circunstâncias extremas pode causar completa ruptura não física da interface. Estas correntes espúrias são um grande problema no caso de microfluidos devido ao fato de que elas aumentam com o inverso do número de capilaridade, mas podem ser reduzidas em magnitude com maior refinamento da malha e/ou com a diminuição do passo de tempo.

Outra dificuldade encontrada neste tipo de simulação consiste no alto custo computacional pois são necessários grandes refinamentos da malha do domínio computacional, com o intuito de capturar os detalhes do escoamento.

2.2 Modelagem de escoamentos Multifásicos.

As técnicas de modelagem disponíveis para escoamentos multifásicos utilizam os princípios da Mecânica dos Meios Contínuos, que de forma geral podem ser classificadas como apresentado na Figura 2.8 (SARKAR, 2008).

Na modelagem molecular (*Molecular Dynamics, MD*) cada molécula de fluido é acompanhada de maneira individual. As trajetórias de átomos e moléculas são determinadas resolvendo numericamente as equações do movimento de Newton para um sistema de partículas que interagem. As forças entre as partículas e sua energia potencial são definidas de acordo com a mecânica molecular. O modelo mesoscópico considera principalmente métodos baseados em partículas como é o caso de *Dissipative Particle Dynamics (DPD)* e o método de *Lattice Boltzmann (LBM)*. A modelagem macroscópica é dividida em duas classes: os métodos baseados em partículas como é o caso do método *Smoothed Particle Hydrodynamics (SPH)* e os métodos baseados em sistemas de malhas. Os métodos baseados em malhas podem ser com

ou sem captura e acompanhamento da interface. Entre os métodos de maior importância baseados em captura (chamados também métodos de volume), estão o método Volume de Fluido (VOF), *Level Set* (LS), e o acoplamento VOF e LS (CLSVOF). Já entre os métodos de acompanhamento de interface (chamados também métodos de superfície), estão *Front-Tracking* e Fronteira Imersa (FI). Entre os métodos sem captura ou acompanhamento da interface estão o *Dispersed Phase Model* (DPM), *Eulerian Multiphase Model* (EMM), *Algebraic Slip Model* (ASM) e *Eulerian Granular Multiphase Model* (EGMM).

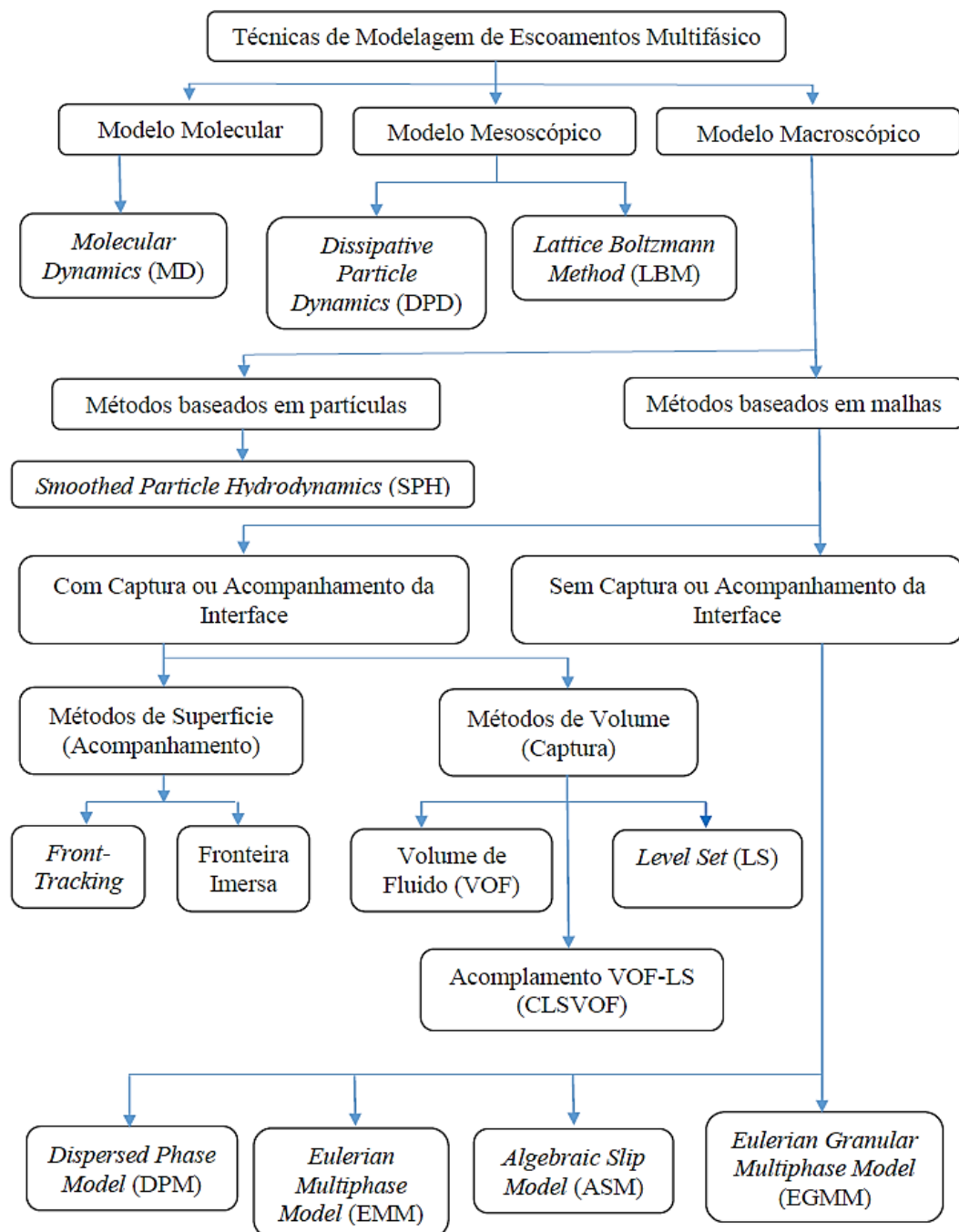


Figura 2.8: Técnicas de modelagem de escoamentos multifásicos. Fonte: Sarkar (2008).

No caso dos métodos baseados em malhas a maior dificuldade está em capturar corretamente a fronteira entre as fases e em calcular as discontinuidades das propriedades físicas através desta interface, tais como massa específica e viscosidade. Uma boa representação da interface implica em um cálculo correto destas propriedades.

Em uma outra classificação geral dos métodos é comum separar os métodos em Euleriano-Euleriano ou Euleriano-Lagrangiano, também chamado de Lagrangiano ou misto. Esta classificação diz respeito à maneira como é considerada a fase dispersa (fase particulada). A fase dispersa pode ser formada por partículas sólidas, bolhas de um gás ou gotas de um líquido, nos escoamentos em que o padrão disperso está presente. A fase contínua pode ser tanto um líquido quanto um gás e desta forma sempre que houver dois ou mais fluidos imiscíveis separados por uma interface, está caracterizado um sistema bi ou multifásico.

Nos métodos de superfície (Acompanhamento), a interface pode ser seguida explicitamente utilizando pontos (partículas) marcadores especiais (Fig. 2.9a). Para os métodos de volume (captura), os fluidos em ambos os lados da interface são, em geral, marcados por partículas de massa desprezível ou por uma função indicadora (Fig. 2.9b). Independente do método empregado, a característica essencial para modelar corretamente a superfície livre deve incluir um esquema para descrever a forma e localização da superfície ao longo do tempo e a aplicação correta de condições de contorno (YEOH e TU, 2010).

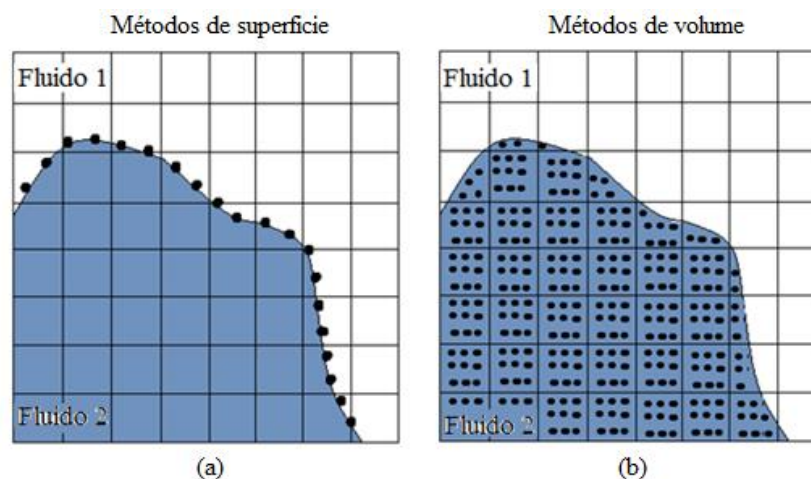


Figura 2.9: Formas de representar superfícies livres e interfaces entre fluidos: (a) Marcadores na interface, (b) Os fluidos são marcados com marcadores sem massa. Fonte: Yeoh e Tu (2010).

Nos métodos de volume (captura) não há necessidade de utilizar elementos computacionais explícitos para marcar a interface (ESTACIO, 2008). Experimentos numéricos têm mostrado que esta metodologia não requer detalhes adicionais no algoritmo para modelar trocas na interface. Uma vantagem deste tipo de método é que a ruptura e coalescência da

interface é tratada automaticamente, mas, em contrapartida, a localização da interface não é bem definida. A desvantagem é que há uma difusão da interface sobre várias células, resultando numa perda de precisão.

No presente trabalho foram utilizados os métodos de captura da interface VOF (implementado no código de OpenFOAM) e o método CLSVOF, o qual foi implementado, posteriormente no código. A seguir são citadas com maior detalhe as técnicas VOF, LS e CLSVOF.

2.3.1 Método Volume de Fluido (VOF)

Hirt e Nichols (1981) propuseram o método denominado Volume de Fluido (VOF) para representar a superfície livre em escoamentos de fluidos. Neste método é utilizada uma função marcadora para reconstrução da superfície livre, que assume valores entre 0 e 1, dependendo da quantidade de fluido em cada célula. O movimento de todas as fases é modelado pela formulação local e instantânea das equações de conservação de massa, quantidade de movimento e energia. A fração volumétrica das fases é calculada em cada célula computacional do domínio por meio da solução da equação da continuidade para cada fase. A fração volumétrica α é uma função cujo valor é unitário em um dos fluidos e zero no outro (Fig. 2.10). Em particular um valor unitário de α corresponde a uma célula cheia com o fluido A, enquanto que um valor zero indica que a célula contém o fluido B. As células com valores de α entre zero e um, contém uma superfície livre.

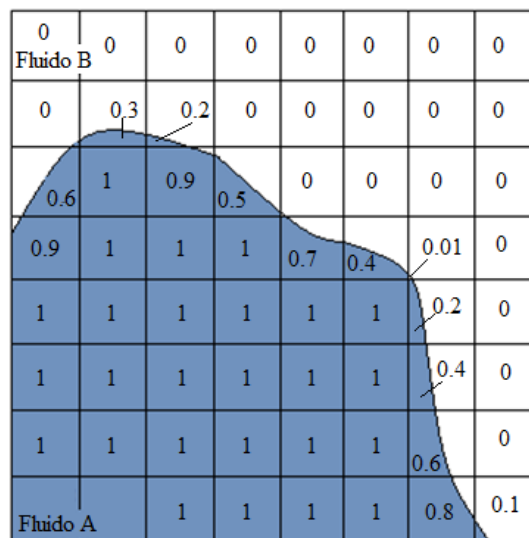


Figura 2.10: Função volumétrica α no método VOF.

A função α mudará de acordo com o movimento da interface e portanto é necessário utilizar uma equação de transporte. A advecção desta função pode ser feita de modo

conservativo ou seja o volume dentro da interface é conservado. Mais detalhes do método são apresentados no Capítulo 3. Como o valor de α por si só não define a exata posição da interface, este método exige uma etapa de reconstrução da geometria, que pode ser feita de diferentes formas.

Os diferentes métodos VOF podem se distinguir pelas características do algoritmo de reconstrução da interface e os esquemas utilizados para a integração no tempo da equação da fração volumétrica. Melhorias sucessivas no método VOF têm o mantido competitivo com outros métodos mais recentes.

Na Figura 2.11 são mostradas duas técnicas, SLIC e PLIC, utilizadas para a reconstrução da interface. No método SLIC (*Simple Line Interface Calculation*), (NOH e WOODWARD, 1976) a interface é reconstruída utilizando uma sequência de segmentos alinhados com a malha (primeira ordem). Estes segmentos são paralelos ou ao eixo x (Fig. 2.11c) ou a eixo y (Fig. 2.11b). No método PLIC (*Piecewise Linear Interface Calculation*), (DEBAR, 1974; YOUNGS, 1982) a interface é aproximada por um segmento de linha reta com o ângulo determinado utilizando o vetor normal à interface naquele volume (segunda ordem), como se apresenta na Figura 2.11d. Nestes métodos o seguimento de linha corta a célula computacional de forma que a fração do volume de fluido se iguale ao valor da fração volumétrica naquela célula. O polígono de fluido resultante é então usado para determinar os fluxos através de qualquer face das células.

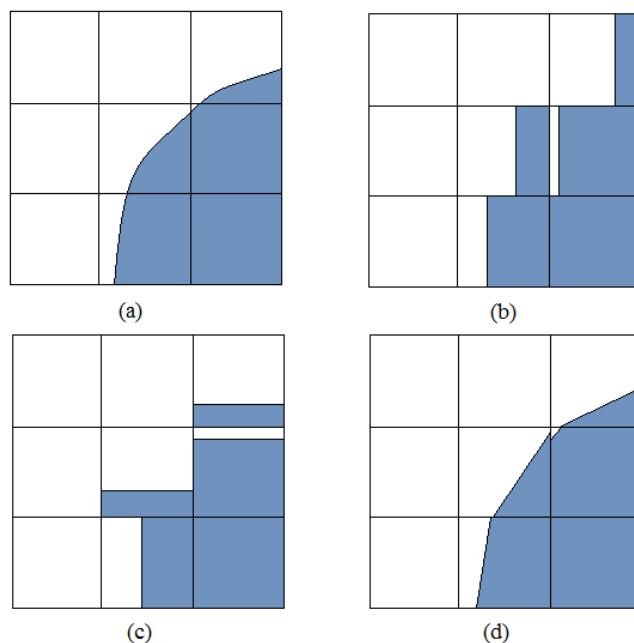


Figura 2.11: Comparação das diferentes técnicas para reconstrução da interface. (a) Interface original, (b) e (c) método SLIC e (d) método PLIC. Fonte: Yeoh e Tu (2010).

As moléculas de fluido na interface de um escoamento bifásico ou próximas a ela produzem forças de atração molecular. Estas forças de tensão superficial tendem a resistir a qualquer aumento na área da interface, sendo preciso inclui-las ao método de modelagem. Brackbill, Kothe e Zemach (1992), propôs o modelo CSF (*Continuum Surface Force*). A idéia básica deste modelo consiste em considerar a relação entre dois fluidos como uma região de transição com uma espessura finita, em vez de uma membrana de espessura zero. O efeito da tensão superficial é assim interpretado como uma força de corpo contínua, distribuída pela região de transição. Lafaurie *et al.* (1994) formulou o modelo CSS (*Continuum Surface Stress*) introduzindo um tensor de pressão capilar. O tensor tem sido empregado como uma correção para o tensor de tensão dinâmica. Mais tarde Jacqmin (1996) propôs o modelo CST (*Continuum Surface Tension*) em que a força de tensão superficial pode ser expressa como a divergência de um tensor de tensão, que é definido exclusivamente pelo gradiente da função característica.

Muitos pesquisadores estudaram o método VOF com o objetivo de melhorar o cálculo da interface no caso tridimensional. Geuyffier *et al.* (1999) apresentaram um método baseado em um tratamento simplificado da equação do movimento com uma malha fixa, regular e uniforme. A técnica PLIC foi implementada pelos autores, com uma advecção de Lagrange da interface. O tratamento da tensão superficial consistiu em suavizar artificialmente a descontinuidade presente na interface com duas técnicas: o CSS (*Continuum Surface Stress*) e o CSF (*Continuum Surface Force*). A combinação destas duas técnicas permitiu realizar cálculos tridimensionais incluindo a reconexão e rompimento da interface.

Rider e Kothe (1998) propuseram um algoritmo para o acompanhamento da interface em duas dimensões baseado em uma solução geométrica de segunda ordem da equação de evolução da fração volumétrica. O método utiliza o volume de material em pontos discretos locais e dados de velocidade para rastrear a interface que pode ter topologia arbitrária e complexa. As soluções são de segunda ordem no espaço através de uma aproximação geométrica da interface com o método PLIC, e no tempo com um esquema de integração multidimensional.

Entre os métodos mencionados até agora para a reconstrução da interface, o método PLIC aparece como uma variante bastante precisa, embora os efeitos devido à tensão superficial continuem a ser um problema. O método CSF (LAFURIE *et al.*, 1994; ZALESKI *et al.*, 1995; SCARDOVELLI e ZALESKI, 1999) figura entre os métodos mais prominentes. Ambos, PLIC e CSF) induzem correntes parasitárias com precisão moderada em relação à determinação da curvatura. As correntes parasitárias são movimentos de fluidos não físicos gerados em regiões

adjacentes a uma interface por variações locais da força de corpo, que geralmente aparecem ao usar implementações como as técnicas mencionadas anteriormente.

Renardy e Renardy (2002) desenvolveram uma representação da força de corpo devido à tensão superficial que elimina efetivamente as correntes parasitárias. O algoritmo apresentado por eles foi chamado PROST (*Parabolic Reconstruction of Surface Tension*). Vários componentes foram implementados no novo algoritmo com o intuito de melhorar a solução das equações de Navier Stokes assim como um esquema de advecção de ordem superior para a interface. Os autores afirmam que o PROST elimina eficazmente as correntes parasitárias frequentes nos métodos VOF.

Wang, Yang e Stern, (2012) apresentaram um método VOF de segunda ordem para o rastreamento da interface e tratamento de interfaces afiladas em malhas estruturadas gerais. A ideia principal deste novo método é o esquema de segunda ordem de construção de uma função distância numa malha estruturada baseada na interface reconstruída. A técnica foi desenvolvida para avaliar o vetor normal à interface usando a função de distância. Com o vetor normal, a interface é reconstruída a partir da função fração do volume através do esquema PLIC. Os autores afirmam que o novo método geral VOF é mais eficiente que o acoplamento Level Set e VOF (CLSVOF) citado neste capítulo.

2.3.2 Método Level Set (LS)

O método *Level Set*, introduzido por Osher e Sethian (1988) e desenvolvido por Sussman, Smereka e Osher (1994) para simulação de escoamentos multifásicos, é uma técnica computacional para o acompanhamento de interfaces que se propagam com velocidade dependente da curvatura. Tal método, está sedimentado na seguinte ideia: “representar uma determinada curva (ou superfície) Γ como a curva de nível zero de uma função Φ de maior dimensão (denominada função Level Set)” (NAPOLITANO, 2004). Neste método a interface é descrita por uma curva de nível zero de uma função contínua definida pela distância algébrica entre um ponto do domínio e a interface. A imagem localizada à esquerda da Fig. 2.12 apresenta unicamente a expansão de uma circunferência, enquanto que a imagem situada à direita ilustra a tal circunferência como sendo a curva de nível zero de uma função de maior dimensão. A normal e a curvatura podem ser facilmente calculadas por meio desta função que possui forma contínua e suave. A função distância precisa ser reinicializada a cada passo de tempo para que a massa seja conservada.

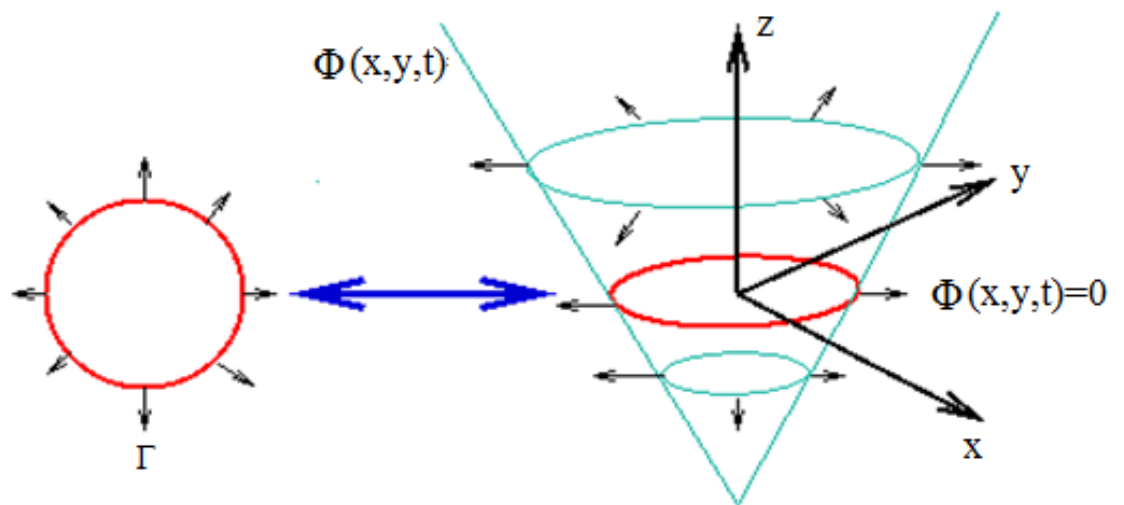


Figura 2.12: Função Level Set. Fonte: Napolitano (2004).

O método LS é especialmente indicado para problemas que envolvem mudanças topológicas, dependência da curvatura ou formação de singularidades. Os algoritmos LS trabalham com fusão topológica e qualquer número de dimensões espaciais e não requerem que o movimento da superfície seja escrito como uma função.

Comparado com o método VOF, o método LS é mais fácil de implementar e adequado para problemas tridimensionais. O maior desafio do método LS em sua proposta inicial é que a conservação de massa é frequentemente violada. Devido à esta dificuldade, escoamentos como ruptura de ondas, formação de sprays e outros não podem ser capturados. Mesmo que a malha possua uma resolução suficiente, existem exemplos como o caso de gotas ou bolhas de ar inicialmente formadas que tendem a diminuir de tamanho durante o passo de tempo da simulação.

Vários autores vêm estudando, aplicando e aprimorando este método, como o caso de Sussman, Smereka e Osher (1994), que apresentaram um tratamento do método *Level Set* para simular escoamentos bifásicos com grandes razões de massa específica e viscosidades, além da inclusão do efeito da tensão superficial. A característica importante da metodologia implementada nesse trabalho é que a função Level Set permanece como uma função distância por todo o tempo da simulação, sem uma reconstrução da interface. Mais tarde Sussman *et al.* (1999) apresentaram um método numérico também utilizando a abordagem *Level Set* para resolver escoamentos bifásicos incompressíveis. Neste caso o método LS é acoplado com uma metodologia de malha adaptativa para resolver escoamentos incompressíveis com a presença de interfaces livres. Embora o método LS não tenha as mesmas propriedades de conservação que o método VOF, o ponto forte do método é a capacidade de calcular com precisão

escoamentos com a presença de interfaces em constantes mudanças de topologia. Além disso, o método LS pode ser generalizado facilmente a três dimensões sem necessidade de acrescentar um código extra a fim de unir a interface dos fluidos separados.

Uma revisão do método LS na qual se discutem variantes e extensões, incluindo o movimento de curvas em três dimensões foi realizada por Osher e Fedkiw (2001). O método de extensão superficial dinâmica, métodos rápidos para problemas de estado estacionário, a difusão do movimento e outras modificações do método LS foram revisados neste trabalho. Eles apresentam uma visão geral para definir o estado da arte da metodologia, assim como a sua aplicação para uma grande variedade de problemas físicos, por exemplo, o caso de escoamentos multifásicos.

Sethian e Smereka (2003) também apresentam uma visão geral do método LS para a solução de escoamentos com a presença de interfaces fluidas. Mostraram uma técnica computacional baseada em formulações implícitas da interface, representada por meio de uma equação diferencial parcial dependente do tempo. Ideias essenciais presentes nestas técnicas, o seu acoplamento com métodos de diferenças finitas para escoamentos incompressíveis e compressíveis são discutidas neste trabalho, além de apresentar várias aplicações incluindo para escoamentos bifásicos.

Uma das dificuldades do método LS é a pouca conservação da massa como já foi dito. No entanto, Nourgaliev *et al.* (2005) mostraram que a conservação da massa do método *Level Set* pode ser significativamente melhorada reduzindo erros de discretização espaciais e que isto pode ser conseguido sem significativas complicações com a implementação de dois elementos importantes: (a) o aumento da precisão na discretização do espaço usando o esquema LWENO (*Linear Weighted Essentially Non-Oscillatory*), e (b) aumentar a resolução local da malha usando o esquema SAMR (*Structured Adaptive Mesh Refinement*).

Luo *et al.* (2006) desenvolveram uma fórmula geral para um método de projeção de segunda ordem combinado com *Level Set* para simular escoamentos multifásicos incompressíveis com mudança de fase. O esquema de terceira ordem ENO (*Essentially Non-oscillatory*) e de segunda ordem semi-implícito (*Crank-Nicholson*), foram usados para atualizar o termo convectivo e de difusão. Os resultados numéricos mostraram que a metodologia é confiável para modelar escoamentos multifásicos com deformações complexas da interface.

Quando a função distância do método *Level Set* é reinicializada na vizinhança de uma linha de contato, existe um “ângulo morto”, que impede uma reconstrução precisa desta função.

O método numérico pode criar correntes parasitárias de velocidade perto desta região. Della e Blanquart (2014) propuseram uma rotina modificada de reinicialização que combina a equação padrão de Hamilton-Jacobi com uma equação de relaxamento para as células da malha ao longo da parede, no ponto cego. O método proposto apresenta menos distorção da interface, induzida numericamente, que outras técnicas de análise. Além disso, esta rotina pode ser facilmente estendida para três dimensões. As vantagens deste método foram apresentadas em simulações de gotas deslizando sobre uma parede em duas e três dimensões.

2.3.3 Acoplamento dos Métodos Level Set - VOF (CLSVOF)

Sussman e Puckett (2000) apresentaram um acoplamento entre os métodos *Level Set* (LS) e Volume de Fluido (VOF) denominado CLSVOF, para o cálculo de escoamentos bifásicos tridimensionais, incompressíveis e axissimétricos. Os resultados foram comparados com dados experimentais de uma bolha de gás ascendendo em um meio líquido e demonstraram que o método é superior a qualquer dos dois métodos individuais. Mais tarde Sussman (2003) estudou com o método CLSVOF o crescimento e colapso de bolhas de vapor utilizando algoritmos de segunda ordem para corrigir a conservação de massa. Neste trabalho foram apresentados exemplos que se aplicam à cavitação de bolhas.

Um método adaptativo do acoplamento *Level-set*/Volume de fluido (ACLSVOF) para simulações de escoamentos interfaciais em malhas triangulares não estruturadas foi desenvolvido por Yang *et al.* (2006). Usando malhas não estruturadas adaptativas, pode-se refinar a malha na proximidade da interface, o que permitiu resolver com eficiência e precisão, complexas mudanças na topologia da interface. O algoritmo desenvolvido aproveita as vantagens dos métodos LS e VOF, o que torna o cálculo do vetor normal à interface e, portanto, o cálculo da tensão superficial mais fácil e precisa e a massa também é melhor conservada. Simulações da deformação de gotas em escoamento extensional mostram que o método pode capturar com precisão a interface e é capaz de analisar um problema físico real.

Chenadec e Pisch (2013) propuseram um novo algoritmo para a captura da interface. O método híbrido Euler-Lagrange para escoamentos multifásicos (HyLEM) é também uma combinação da abordagem *Level Set* e o esquema de transporte VOF. Neste trabalho vários esquemas de integração numérica foram propostos e testados. O método HyLEM implementado preservou exatamente a massa fornecendo erros de truncamento na deformação da malha muito pequenos.

Kwakkel, Breugem e Boersma (2013) empregaram um marcador múltiplo acoplado ao método CLSVOF para escoamentos com presença de várias gotas, aplicando um modelo físico para representar a coalescência e quebra das gotas. No método utilizado cada gota tem suas próprias funções marcadoras definidas localmente. Isto evita o problema da coalescência numérica comum nos métodos *Level Set* e VOF quando duas interfaces de gotas estão mais próximas que o tamanho equivalente de uma célula. A boa concordância com os dados experimentais para cada número de Weber simulado demonstrou a capacidade do método para simular coalescência e rompimento de gotas com realismo físico.

Albadawi *et al.* (2013) propuseram um método simples de acoplamento das metodologias VOF e LS (S-CLSVOF) que melhora o cálculo da tensão superficial. Nesta proposta o cálculo inicial da função Level Set é feito usando a função VOF e, em seguida, uma reinicialização é realizada para obter a função distância. A força de tensão superficial é calculada no método LS considerando a função Delta de Dirac. O código de acoplamento implementado no OpenFOAM denominado S-CLSVOF foi validado com os resultados experimentais da ascensão de uma bolha circular. Os resultados obtidos com esta nova proposta de cálculo da tensão superficial foi também comparado com o código já implementado no OpenFOAM com o *solver* interFoam e a metodologia VOF. Os autores comentam que houve significativa melhora na diminuição das correntes parasitárias.

As primeiras simulações feitas no presente trabalho foram conduzidas com a metodologia Volume de Fluido que já estava implementada no código de OpenFOAM. O objetivo destas simulações foi validar o presente código com a bibliografia consultada. Depois foi implementado no código de OpenFOAM a metodologia S-CLSVOF de Albadawi *et al.* (2013) e foram posteriormente comparadas ambas metodologias, especialmente no estudo dos mecanismos de ruptura/não ruptura de gotas escoando através de microcanais com junções em T.

No próximo capítulo é apresentada a formulação matemática para escoamentos bifásicos, dando ênfase ao uso das metodologias VOF e ao acoplamento dos métodos VOF e LS (S-CLSVOF).

Capítulo 3

Fundamentos matemáticos

3.1 Introdução

Neste capítulo são descritas as equações básicas do movimento, Equações de Navier-Stokes, para escoamentos multifásicos e incompressíveis, as condições de contorno apropriadas para este tipo de escoamento e o procedimento de adimensionalização destas equações. É apresentada também a formulação matemática da metodologia Volume de Fluido (VOF) e seu acoplamento com o método *Level Set* (LS). As equações do movimento nas duas metodologias são resolvidas como se apenas um único fluido estivesse presente. Uma função indicadora que relaciona a fração de uma dada fase/fluido é utilizada para localizar a interface. Nesta formulação de fase única, as propriedades dos materiais e interfaciais são levadas em conta pelas equações de conservação.

3.2 Equações de Navier-Stokes

No presente trabalho, a fase líquida foi tratada como fluido incompressível. Com a suposição de um sistema isotérmico e considerando que a variação da massa específica do gás provocada pela pressão hidrostática do líquido é desprezível, e que todas as simulações foram feitas em regime laminar, a fase gasosa também pode ser tratada como incompressível. As propriedades físicas como massa específica e viscosidade são constantes em cada fase, variando apenas sobre a interface. Desta forma, as equações da conservação da massa e da quantidade de movimento podem ser escritas por

$$\nabla \cdot \mathbf{u} = 0 \quad (3.1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \mathbf{F}_s \quad (3.2)$$

na qual ρ é a massa específica do fluido, \mathbf{u} é o vetor velocidade, t é o tempo, p a pressão, $\boldsymbol{\tau}$ é o tensor de tensões viscosas, \mathbf{g} é o vetor gravidade e \mathbf{F}_s é a força relacionada à tensão superficial por unidade de volume, agindo na interface entre dois fluidos. A pressão p usada no código, em particular no módulo bifásico, do OpenFOAM, é a pressão dinâmica denominada p_{rgh} , sendo $p_{rgh} = p - \rho gh$, e h é a altura do líquido.

As equações da quantidade de movimento são complementadas por uma equação que relaciona as deformações e as tensões no interior do fluido. Neste estudo os fluidos obedecem a lei de Newton da viscosidade, sendo o tensor das tensões viscosas dado por

$$\boldsymbol{\tau} = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \quad (3.3)$$

na qual μ é a viscosidade dinâmica.

3.2.1 Formulação da metodologia VOF

Neste método a fração de volume é utilizada como uma função indicadora (fase α), para marcar os diferentes fluidos. A interface é definida como uma região de transição e é tratada como uma mistura de dois fluidos em cada lado da interface. A fração volumétrica é definida como

$$\alpha = \begin{cases} 1 & (\text{dispersa}) \\ 0 < \alpha < 1 & (\text{transição}) \\ 0 & (\text{contínua}) \end{cases} \quad (3.4)$$

Usando esta definição de fração volumétrica pode-se representar a massa específica e a viscosidade como

$$\rho(\mathbf{x}, t) = \alpha\rho_d + (1 - \alpha)\rho_c \quad (3.5)$$

$$\mu(\mathbf{x}, t) = \alpha\mu_d + (1 - \alpha)\mu_c \quad (3.6)$$

nos quais os índices c e d são utilizados para designar respectivamente as fases contínua e dispersa.

O principal desafio para a advecção da função α é preservar a conservação da massa. O OpenFOAM utiliza uma abordagem com base na formulação do modelo de dois fluidos. Neste modelo, um termo adicional, proveniente da modelagem da velocidade em termos de uma medida ponderada das correspondentes velocidades das fases contínua e dispersa é introduzido na equação de transporte para a fração de volume, proporcionando uma definição mais nítida da interface (BERBEROVIĆ *et al.*, 2009). A metodologia faz uso do modelo Euleriano de dois fluidos para o escoamento de duas fases, na qual as equações da fração de volume são resolvidas separadamente para cada fase individual, como proposto por Cerne, Petelin e Tiselj (2001). As equações para cada uma das frações de fase podem ser expressas como:

$$\frac{\partial(\alpha)}{\partial t} + \nabla \cdot (\mathbf{u}_d \alpha) = 0 \quad (3.7)$$

$$\frac{\partial(1 - \alpha)}{\partial t} + \nabla \cdot [\mathbf{u}_c (1 - \alpha)] = 0 \quad (3.8)$$

Assumindo que as contribuições das velocidades das fases dispersa e contínua para a evolução da superfície livre são proporcionais à fração de fase correspondente, pode-se então definir a velocidade do escoamento no modelo VOF como uma média ponderada, como segue

$$\mathbf{u} = \alpha \mathbf{u}_d + (1 - \alpha) \mathbf{u}_c \quad (3.9)$$

A Equação (3.7) pode ser rearranjada e usada como uma equação de transporte da fração de volume α ,

$$\frac{\partial(\alpha)}{\partial t} + \nabla \cdot (\mathbf{u}\alpha) + \nabla \cdot (\mathbf{u}_r \alpha (1 - \alpha)) = 0 \quad (3.10)$$

na qual $\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$ é o vetor velocidade relativa, designado como “velocidade de compressão”.

A Equação (3.10) tem um termo convectivo adicional, denominado “termo de compressão” (\mathbf{u}_r), tendo em mente o papel de “comprimir” a superfície livre em direção a uma mais nítida, nota-se que o termo compressão representa apenas uma denotação e não é referido ao escoamento compressível.

A velocidade de compressão é levada em consideração apenas na região da interface e é calculada na direção normal à interface para evitar qualquer dispersão (ALBADAWI *et al.*, 2013). Além disso, um fator de compressão c_α é usado para aumentar a compressão como

$$\mathbf{u}_r = \min(c_\alpha |\mathbf{u}|, \max(|\mathbf{u}|)) \frac{\nabla \alpha}{|\nabla \alpha|} \quad (3.11)$$

Neste trabalho foram testados valores de um fator de compressão $c_\alpha = 1, 2$ e 3 .

3.2.2 Tensão Superficial

O termo de força superficial adicionado às equações de Navier-Stokes (Eq. 3.2) cria uma dificuldade adicional já que é um termo singular. Esta dificuldade se manifesta em instabilidades e/ou ruídos numéricos, bem como na representação de efeitos capilares. Nos métodos de acompanhamento da interface a tensão superficial pode ser avaliada diretamente pela representação discreta da interface. Por outro lado, em métodos de captura, como é o caso de VOF e LS, a posição exata da interface é desconhecida e a localização da interface é frequentemente realizada por uma função indicadora.

Vários métodos têm sido propostos. No caso do *software* OpenFOAM é utilizado o método CSF, *continuum surface force*, que representa os efeitos da tensão superficial como uma força volumétrica contínua agindo dentro da região de transição, podendo ser escrita como

$$\mathbf{F}_s = \int_{S(t)} \sigma k(x, t) \mathbf{n} \delta(x) dS \quad (3.12)$$

na qual $k(\mathbf{x}, t)$ é a curvatura da interface, σ o coeficiente da tensão superficial e $\delta(x)$ a função delta de Dirac concentrada na interface e nula nas demais regiões. O vetor unitário normal à interface é dado por:

$$\mathbf{n} = \frac{\nabla\alpha}{|\nabla\alpha|} \quad (3.13)$$

A curvatura é calculada como

$$k(\mathbf{x}, t) = -(\nabla \cdot \mathbf{n}) \quad (3.14)$$

Esta curvatura representa a magnitude do fluxo normal da interface em uma zona específica da célula e indica a direção desse fluxo. Nesta última equação a orientação da interface é calculada por meio do gradiente da função α . Finalmente o termo tensão superficial pode ser aproximado pela seguinte expressão

$$\mathbf{F}_s = \int_{S(t)} \sigma k(x, t) \mathbf{n} \delta(x) dS = \sigma k \nabla \alpha \quad (3.15)$$

Um dos objetivos deste trabalho foi a implementação do acoplamento CLSVOF no código fonte do OpenFOAM. No item seguinte é apresentada a fundamentação matemática desta metodologia e sua comparação com o método VOF.

3.2.3 Acoplamento VOF com LS (CLSVOF)

O acoplamento VOF com LS tem a vantagem da conservação da massa do método VOF e a boa captura da interface do método LS. O algoritmo utilizado neste trabalho, denominado S-CLSVOF foi desenvolvido por Albadawi *et al.* 2013 para a simulação de bolhas ascendendo em meios líquidos. Nessa implementação da função *Level Set*, o campo ϕ é introduzido na posição da interface definida por uma isolinha $\phi = 0$ (Fig. 2.12). Inicialmente, uma única equação de advecção (Eq. 3.10) é resolvida, conforme realizado por Sussman e Puckett (2000), ao invés de duas equações separadas, das frações volumétricas VOF e LS.

O primeiro passo do acoplamento é atribuir um valor inicial à função LS usando a função fração de volume α e assumindo que a posição da interface é definida no contorno da isolinha onde $\alpha = 0,5$

$$\phi_0 = (2\alpha - 1) \cdot \Gamma \quad (3.16)$$

na qual Γ é um número adimensional que depende do tamanho da malha Δx . Neste trabalho foi adotado o valor $\Gamma = 0,75 \Delta x$. O critério principal na escolha deste valor é satisfazer um valor inicial ϕ que está próximo da dimensão da malha. Esta função inicial ϕ_0 tem um valor

positivo na fase contínua e um valor negativo na fase dispersa. Este valor ϕ_0 é recalculado resolvendo a equação de reinicialização

$$\left\{ \begin{array}{l} \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) \\ \frac{\partial \phi}{\partial \tau} = S(\phi_0)(1 - |\nabla \phi|) \end{array} \right. \quad (3.17)$$

na qual τ é o tempo artificial, \mathbf{x} é o vetor posição e $S(\phi_0)$ é uma função de sinal. A solução converge quando o $|\nabla \phi| = 1$. O passo do tempo artificial foi escolhido como $\Delta \tau = 0,1\Delta x$, de modo que não haja mudanças bruscas na função LS durante a reinicialização.

Na equação de reinicialização (Eq. 3.17) são necessárias algumas iterações (ϕ_{corr}). Para os valores escolhidos de Γ e $\Delta \tau$, o número de iterações (ϕ_{corr}) é encontrado de acordo a condição

$$\phi_{corr} = \frac{\epsilon}{\Delta \tau} \quad (3.18)$$

na qual ϵ é a espessura da interface que define o número de células utilizadas para a transição entre os dois fluidos e foi escolhida como $2\epsilon = 3\Delta x$ (ALBADAWI *et al.* 2013).

Como a função LS é contínua, pode-se determinar com precisão o vetor normal à interface, definido como: $\hat{\mathbf{n}} = \nabla \phi / |\nabla \phi|$. Assim, ele fornece uma curvatura de interface mais precisa e suave $k = \nabla \cdot \hat{\mathbf{n}}$.

A força de tensão superficial no acoplamento VOF-LS pode então ser calculada como

$$\mathbf{F}_S = \sigma k(\phi) \delta(\phi) \nabla \phi \quad (3.19)$$

na qual δ é a função Delta de Dirac usada para limitar a influência da tensão superficial para uma região estreita sobre a interface. A função $\delta(\phi)$, que é centralizada na interface, possui um valor zero em ambos os fluidos e é definida como

$$\delta(\phi) = \begin{cases} 0 & \text{se } |\phi| > \epsilon \\ \frac{1}{2\epsilon} \left[1 + \cos\left(\frac{\pi\phi}{\epsilon}\right) \right] & \text{se } |\phi| \leq \epsilon \end{cases} \quad (3.20)$$

As propriedades físicas através de cada célula podem ser definidas como no método VOF e calculadas usando uma função de *Heaviside*, ou degrau, como

$$H(\phi) = \begin{cases} 0 & \text{se } \phi < -\epsilon \\ \frac{1}{2} \left[1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \text{sen}\left(\frac{\pi\phi}{\epsilon}\right) \right] & \text{se } |\phi| \leq \epsilon \\ 1 & \text{se } \phi > \epsilon \end{cases} \quad (3.21)$$

$$\rho(\phi) = \rho_d + (\rho_c - \rho_d)H \quad (3.22)$$

$$\mu(\phi) = \mu_d + (\mu_c - \mu_d)H \quad (3.23)$$

As Equações (3.22) e (3.23) fornecem uma transição mais suave das propriedades através da interface, que o uso das Eqs. (3.5) e (3.6) (ALBADAWI *et al.* 2013) que utilizam os valores da fração volumétrica α . Na Figura 3.1 apresenta-se o fluxograma do acoplamento S-CLSVOF.

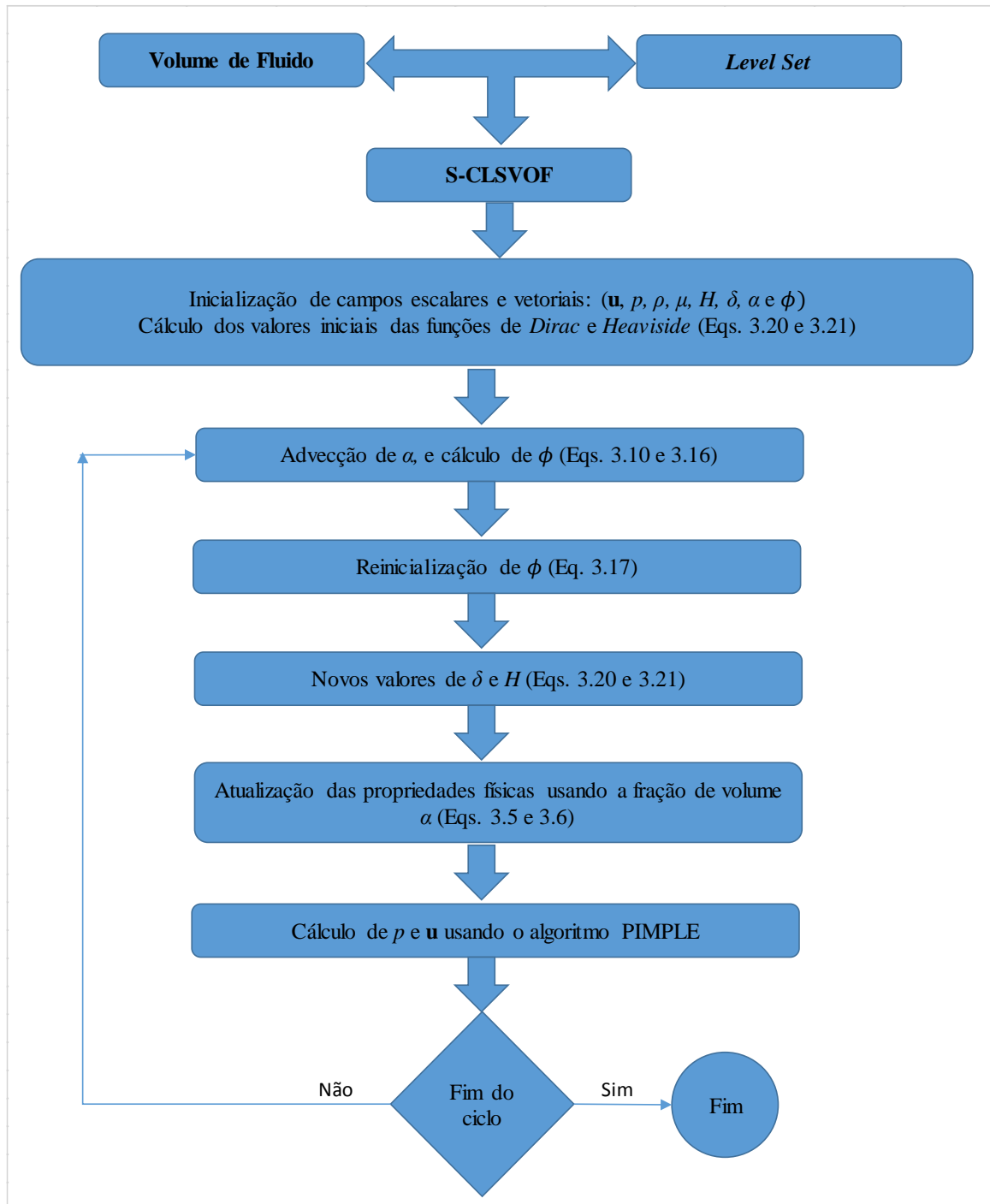


Figura 3.1: Fluxograma do acoplamento S-CLSVOF.

3.3 Condições iniciais e de contorno.

As equações para escoamentos bifásicos discutidas anteriormente necessitam de condições apropriadas para que possam ser resolvidas. A especificação correta destas condições é fundamental para se obter uma solução numérica coerente com o problema em questão. Para os casos estudados são utilizadas as seguintes condições de contorno:

- Nas fronteiras sólidas aplica-se, neste caso, a condição “não deslizamento” (*no-slip*). Esta condição considera que o fluido imediatamente adjacente à superfície sólida deve estar em repouso em relação a essa superfície. Essa condição também é denominada como condição de aderência completa, onde as componentes normal \mathbf{u}_n e tangencial \mathbf{u}_t da velocidade nesta superfície sólida são nulas. Tanto nas simulações de bolhas ascendendo em meios líquidos quando de gotas escoando num microcanal foi aplicada esta condição.
- Outra condição de fronteira que pode ser considerada é a condição de “livre deslizamento” (*slip*), que permite que o fluido deslize livremente sobre a superfície sólida. É também aplicada em problemas nos quais há um plano de simetria. Define-se, neste caso, a componente normal nula e $\partial\mathbf{u}_t/\partial\mathbf{n} = 0$, para a componente tangencial, em que \mathbf{n} é a direção normal à superfície da fronteira. Esta condição só foi usada nas simulações de bolhas ascendendo com o intuito de comparar com a literatura consultada. Nestes casos não foi aplicado nenhum plano de simetria.
- Condição inicial para a velocidade: (a) no caso da bolha ascendendo num meio viscoso, tanto a bolha quanto o líquido têm uma velocidade inicial de nula. A bolha ascende por flutuabilidade. (b) Para a gota movimentando-se num microcanal na maioria das simulações foi estabelecida uma velocidade inicial para a fase dispersa (gota) de 0,04 m/s e para a fase contínua (líquido) foi adotado um valor inicial em uma faixa de 0,0001 a 0,035 m/s com o intuito de obter um valor do número de capilaridade (Ca) entre 0,0007 e 0,09.
- Condição inicial para a pressão: na entrada e na saída a pressão é desconhecida e um valor limite é extrapolado a partir do interior do domínio de fluido. No entanto, se o gradiente da pressão tanto na entrada quanto na saída é pequeno, é suficiente aplicar a condição de Neumann (FAVERO, 2009) ou tipo *zeroGradient*.
- Esta última condição é também usada tanto nas entradas quanto nas saídas do domínio computacional e nas condições iniciais da função LS (ϕ).

Estas condições são chamadas de *condições de acoplamento*, porém as condições para pressão e velocidade na interface virão diretamente das equações da continuidade (Eq. 3.1) e da quantidade de movimento (Eq. 3.2).

3.4 Números adimensionais relevantes

Em problemas de mecânica dos fluidos, as propriedades do escoamento e dos fluidos envolvidos podem ser caracterizadas por grandezas específicas. Essas propriedades dimensionais podem ser agrupadas em parâmetros adimensionais, os quais passam a caracterizar um escoamento (FORTUNA, 2000). A facilidade em estudar problemas nos quais as equações estão adimensionalizadas está intimamente ligada ao fato de se poder trabalhar com modelos nos quais a geometria difere apenas nas escalas (similaridade), possibilitando e garantindo que o comportamento que se quer simular é o mesmo que se observaria na prática. Assim, se em um escoamento as equações de movimento estiverem adimensionalizadas e forem similares, então são irrelevantes as dimensões.

Nos problemas que envolvem a ascensão de uma bolha em um fluido destacam-se três parâmetros adimensionais relevantes, sendo eles:

- **Número de Reynolds (Re):** Nomeado pelo físico e engenheiro irlandês Osborne Reynolds (1842 – 1912). É a razão entre as forças inerciais e as forças viscosas, ou seja:

$$Re = \frac{\rho LU}{\mu} \quad (3.24)$$

onde ρ , L , U e μ são, respectivamente, os valores de referência de massa específica, comprimento característico (por exemplo, diâmetro da bolha), velocidade e viscosidade dinâmica. O número de Reynolds indica se o escoamento é *laminar* (abaixo de um Re crítico), *turbulento* (altos Re), ou se encontra em uma faixa de transição entre laminar e turbulento.

- **Número de Morton (Mo):** Utilizado com o número de Eötvös para caracterizar a forma de bolhas e gotas, é a razão entre o produto das forças gravitacional e viscosa e o produto da força de inércia e a tensão superficial. O número de Morton é definido como

$$Mo = \frac{g\rho\mu_l^4}{\rho_l^2\sigma^3} \quad (3.25)$$

- **Número de Eötvös (Eo):** Nomeado pelo físico húngaro Loránd Eötvös (1848 -1919) é também conhecido como **número de Bond (Bo)**, nomeado pelo físico e engenheiro

Inglês Wilfrid Noel de Bond (1897 – 1937). Com o número de Morton pode ser utilizado para caracterizar a forma de uma bolha/gota e é a razão entre a força gravitacional e a tensão superficial, ou seja

$$Eo = \frac{g\rho D^2}{\sigma} \quad (3.26)$$

no qual g é a aceleração da gravidade, σ é a tensão interfacial e D o diâmetro da bolha.

No caso do movimento de gotas em microcanais o principal parâmetro adimensional é o número de Capilaridade. Além deste, o número de Reynolds e a força interfacial também são parâmetros relevantes do escoamento.

- **Número de capilaridade (Ca):** Representa o efeito relativo entre a força viscosa e a tensão superficial que atua na interface entre um líquido e um gás ou entre dois líquidos imiscíveis. É definido como

$$Ca = \frac{\mu U}{\sigma} \quad (3.27)$$

na qual U é a velocidade característica. No presente trabalho o valor de U é dado como sendo a velocidade da bolha/gota.

Outras variáveis como a velocidade e o tempo foram adimensionalizados da seguinte forma para ascensão de bolhas

$$\mathbf{u}^* = \frac{\mathbf{u}}{\sqrt{gD}}; \quad \tau^* = \frac{t}{\sqrt{D/g}} \quad (3.28)$$

No caso de gotas em microcanais o tempo adimensional é definido como

$$\tau^* = \frac{U t}{w} \quad (3.29)$$

na qual U , t e w são, respectivamente, a velocidade baricêntrica da gota, o tempo e a largura do microcanal.

No capítulo 4 é apresentado o método numérico usado para resolver as equações citadas anteriormente, assim como uma breve descrição do software livre OpenFOAM empregado no presente trabalho.

Capítulo 4

MÉTODO NUMÉRICO

Neste capítulo estão descritas as características principais do *software* OpenFOAM. Neste contexto são citadas as vantagens do uso de OpenFOAM assim como alguns detalhes do módulo para estudos de escoamento bifásicos. Os métodos de discretização espaciais e temporais das equações de Navier-Stokes e o acoplamento pressão-velocidade são também apresentados. Posteriormente é descrita a estrutura de um caso no OpenFOAM para ilustrar as principais funcionalidades dos arquivos que, em geral, são encontrados na maioria dos casos. Devido ao fato de ter sido implementado o acoplamento das metodologias VOF e LS, foi preciso entender o acesso ao código fonte do programa para fazer novas implementações. A reconstrução da interface é também um passo importante tanto no método VOF quanto no método S-CLSVOF. Neste capítulo é descrita a principal diferença entre os *solvers* *interFoam* e *interDyMFoam* previamente disponíveis no código de OpenFOAM e baseados na metodologia VOF. Finalmente são mostradas as principais funcionalidades do pós-processador *ParaView* usado para a análise dos resultados de todas as simulações realizadas neste trabalho.

4.1 Fluidodinâmica Computacional

A Dinâmica dos Fluidos Computacional, DFC ou no inglês CFD, *Computational Fluid Dynamics* é uma poderosa ferramenta para a resolução de problemas envolvendo escoamentos com a utilização de métodos numéricos. Têm sido aplicada em diferentes áreas como aerodinâmica e hidrodinâmica de embarcações, combustão, bioengenharia, oceanografia, além de aplicações na engenharia química e outras. Dentre as vantagens da utilização de técnicas de CFD, as principais são: a redução do custo e do tempo necessário para a obtenção de resultados, a possibilidade do estudo de sistemas em que experimentos são muito difíceis ou impossíveis de serem realizados, a viabilização do estudo de sistemas sob condições perigosas além de seus limites normais de operação e a geração de resultados mais detalhados do que os gerados na forma experimental.

A metodologia de solução de problemas utilizando técnicas numéricas envolve três etapas: o pré-processamento, o processamento e o pós-processamento.

No pré-processamento ocorre a definição e construção da geometria na região de interesse, a geração da malha e a escolha das condições de contorno e iniciais que melhor representam o problema a ser estudado. Além disso, os métodos de discretização são definidos

nesta etapa. A geração da malha consiste na divisão do domínio em subdomínios (volumes de controle ou células). Também são definidas as propriedades dos fluidos e se necessário as equações constitutivas a serem usadas em conjunto com as leis de conservação tais como, modelos de mudança de fase, modelos de turbulência e outros. (VERSTEEG e MALALASEKERA, 1995). Neste trabalho, por se tratar da dinâmica de bolhas e gotas em líquidos no regime laminar e sem transferência de calor, não foi analisada a mudança de fase.

No processamento, as equações matemáticas que representam o problema estudado são resolvidas com os métodos escolhidos e no caso do OpenFOAM os resultados são armazenados em pastas que são geradas à medida que o tempo da simulação avança.

Após a resolução do problema (processamento), os resultados são visualizados e avaliados por meio de uma ferramenta de pós-processamento. Atualmente existem várias ferramentas disponíveis para este fim, tanto gratuitas quanto comerciais. Dentre elas, pode-se citar o *Ensign* (*Computational Engineering International*), o *Tecplot* e o *Paraview*. Estas ferramentas permitem a visualização da malha e da geometria, assim como os campos das variáveis como pressão, temperatura, velocidade e vorticidade, a geração de gráficos vetoriais, o rastreamento de partículas dentre outras variáveis. Algumas delas permitem ainda a criação de animações temporais que torna possível uma melhor compreensão das mudanças do escoamento ao longo do tempo.

Os códigos numéricos são ferramentas que fornecem grandes vantagens e uma maior facilidade na análise de vários tipos de escoamentos. A importância dessas ferramentas têm crescido muito nos últimos anos devido à confiabilidade e a rapidez com que os resultados são obtidos, além da economia frente a testes experimentais. Este fato deve-se aos avanços no desenvolvimento da modelagem dos fenômenos físicos, das técnicas numéricas para solucionar o problema e do aumento da capacidade computacional. Como consequência, o constante aprimoramento das técnicas e metodologias presentes nos códigos é considerado um ponto crucial do CFD. Desta forma, é interessante que o usuário possa interagir com o código fonte, possibilitando implementar novas formas de solucionar um problema específico. Por este motivo, o código OpenFOAM foi escolhido para ser utilizado no presente trabalho pois, além de ser um *software* com licença gratuita, é também um código aberto, o que permite ao usuário fazer novas implementações se necessário.

4.2 Principais características do OpenFOAM

O OpenFOAM foi criado em 1993, sob o nome de FOAM (*Field Operation and Manipulation*), por Herry Weller e Hrvoje Jasak, então estudantes do *Imperial College*, com o objetivo de operar e manipular campos tensoriais visando sua aplicação na fluidodinâmica. Nos primeiros anos de desenvolvimento o FOAM foi feito com um objetivo comercial pela empresa inglesa Nabla, (LOPES, 2008), com a ideia de fornecer ao usuário final uma ferramenta para simulações de processos fisicamente complexos e oferecendo serviços de consultoria sobre o uso do software e implementação de novos *solvers* com modelagem complexa. Contudo, os desenvolvedores chegaram à conclusão de que a abordagem comercial não era a ideal. Assim, em 10 de dezembro de 2004 o código FOAM tornou-se de domínio público e o programa passou a ser chamado de OpenFOAM, referente ao código aberto, conquistando números cada vez maiores de usuários, principalmente, no meio acadêmico. A comunidade científica têm tratado o OpenFOAM com muito interesse em vários projetos de pesquisa e também tem tido muita aceitação no meio industrial. De fato, grandes empresas multinacionais, tais como Audi, Airbus, Bayer, Shell, Honda, entre outras, já visualizaram seu potencial e utilizam o OpenFOAM no desenvolvimento de projetos de alta tecnologia.

Para entender o funcionamento das bibliotecas do OpenFOAM é necessário ter um conhecimento prévio de C++. O programa foi desenvolvido para o tratamento de problemas que envolvem campos tensoriais, tendo por fim, sua aplicação em diferentes áreas como na dinâmica dos fluidos. O C++ é uma extensão da linguagem C desenvolvida por Bjarne Stroustrup no início da década de 1980, na Bell Laboratories, (DIETEL e DIETEL, 2006). O C++ fornece vários recursos que aprimora a linguagem C, mas, sobretudo, fornece capacidades para a programação orientada a objetos. Objetos são essencialmente componentes de *software* reutilizáveis que modelam itens do mundo real. A maior vantagem na abordagem aplicada à orientação a objetos é remover algumas das falhas encontradas na abordagem sequencial ou contínua. Na abordagem orientada a objetos, os dados são tratados como elementos críticos do programa, chamados objetos, e não é permitido alterá-los livremente. As propriedades inerentes à orientação a objetos em C++ propiciam uma perda de cerca de 10% na velocidade de processamento em relação à linguagem C, o qual pode variar dependendo do conhecimento do programador em relação à efetividade computacional (LOPES, 2008).

O OpenFOAM está dividido em três módulos: os *solvers*, os utilitários e as bibliotecas. As bibliotecas podem ser usadas para fins gerais, como tratamento de campo, pós-processamento e cálculo. Os *solvers* consistem nos códigos utilizados para a resolução de

problemas de campo, os utilitários se encarregam do pré e do pós-processamento e as bibliotecas funcionam como suporte aos *solvers* e utilitários, como por exemplo, fornecendo modelos físicos. As bibliotecas são acompanhadas por um conjunto amplo de *solvers* que definem diferentes fenômenos físicos. Uma breve descrição dessas bibliotecas foi publicada por Weller *et al.* (1998) e mais informações podem ser encontradas na Guia do Usuário do OpenFOAM, no capítulo 3 (OpenCFD LTD, 2009) e no Guia do programador capítulos 1-2 (OpenCFD LTD, 2009) cuja leitura é altamente recomendada para os usuários do OpenFOAM.

O OpenFOAM permite a simulação em paralelo, a utilização de malhas estruturadas e não-estruturadas, assim como a importação das mesmas de *softwares* livres e comerciais. Foi escrito de forma a correlacionar a linguagem matemática à linguagem da programação computacional. Desta forma, as equações diferenciais são representadas por uma estrutura de código que se assemelha à linguagem matemática. As equações da mecânica do contínuo são geralmente apresentadas como equações diferenciais parciais em 3 dimensões no espaço e com variação no tempo. Estas equações contêm escalares, vetores, tensores e seus respectivos campos. A solução deste conjunto de equações envolve cálculo tensorial, procedimentos de discretização e implementação de algoritmos de solução de sistemas de equações lineares.

O OpenFOAM utiliza o método dos Volumes Finitos para a discretização das equações diferenciais. Possui duas bibliotecas básicas: a **FVM.H** e a **FVC.H**. A primeira corresponde à classe *finiteVolumeMethod* e é encarregada de realizar operações implícitas de discretização. Os resultados obtidos pelo uso dessas bibliotecas são armazenados em um objeto do tipo **fvMatrix<Type>**. Desta forma, os termos discretizados por esta biblioteca pertencem à matriz que será resolvida durante o processamento. Na segunda biblioteca, *finiteVolumeCalculus* (FVC.H), os termos da equação diferencial são tratados explicitamente e não são armazenados em uma matriz. As operações com esta classe retornam em um campo geométrico (*geometricField<Type>*). Esta classe é útil no tratamento de termos fonte e termos que podem diminuir ou aumentar a diagonalidade da matriz (MENEZES, 2013).

Como exemplo, a representação da equação do movimento no OpenFOAM é dada como:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\phi\mathbf{u}) - \nabla(\mu\nabla\mathbf{u}) = -\nabla p \quad (4.1)$$

e representada pelo código:

solve

```
(
  fvm::ddt(rho,U)
  + fvm::div(phi,U)
  - fvm::laplacian(mu,U)
  ==
  - fvc::grad(p)
):
```

onde os *templates* *fvm* e *fvc* referem-se respectivamente a operações de discretização implícita e explícita usando Volumes Finitos. O sistema linear resultante é resolvido com a função de classe *solve*.

Os usuários do OpenFOAM não necessitam de um grande conhecimento de programação orientada a objetos e C++ para alterar códigos simples, mas devem conhecer os princípios por trás da orientação a objetos e ter um conhecimento básico da sintaxe de C++ (LOPES, 2008).

4.3 Discretização por Volumes Finitos

O termo “discretização” se refere à aproximação de um problema em um domínio contínuo por elementos discretos. O método dos Volumes Finitos, assim como os métodos de Diferenças Finitas e Elementos Finitos, é aplicado na discretização do espaço físico e temporal. O primeiro tipo de discretização considera a divisão do domínio do espaço em um conjunto de volumes de controle conectados entre si (contíguos) ou com o contorno do domínio. A discretização do espaço físico constrói a chamada malha computacional. Se o problema em estudo é transiente é possível fazer uma discretização temporal, subdividindo o tempo em um número finito de intervalos de tempo.

Na Figura 4.1(a) é representado um domínio discretizado. Um volume de controle é composto por um conjunto de faces, que por sua vez é composta por um conjunto de arestas. As faces podem conectar dois volumes de controle ou um volume de controle ao contorno do domínio. Apresenta-se na Figura 4.1(b) as variáveis relevantes na interconexão de dois volumes de controle vizinhos, P e N . O vetor unitário \mathbf{n} normal à face f é definido como $\mathbf{n} = \mathbf{S}/|\mathbf{S}|$ onde \mathbf{S} é o vetor elemento de área que aponta para fora da área, enquanto que a distância d entre o centro dos volumes P e N é definida como $d = x_N - x_P$. Uma malha é considerada ortogonal quando d é paralelo ao vetor \mathbf{S} em cada face da malha. Note que o vetor \mathbf{S} é normal à face f e

sua magnitude é igual à área da face. Este vetor aponta do volume de interesse P para o vizinho N .

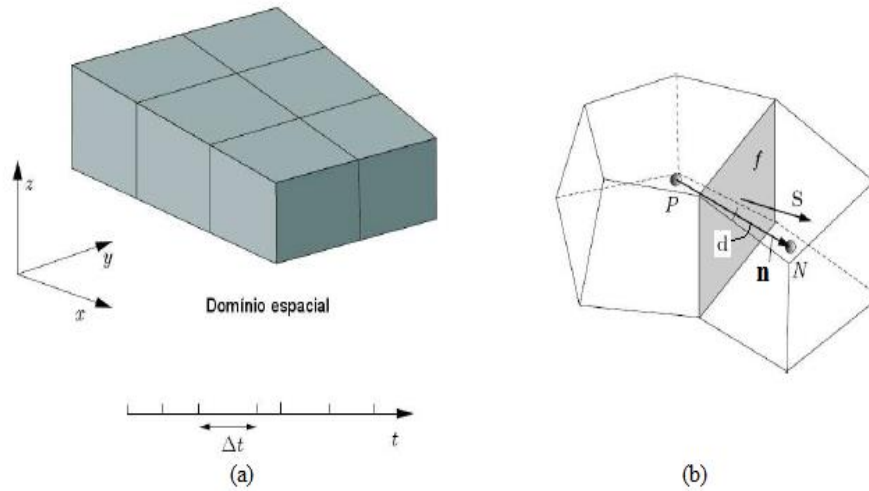


Figura 4.1: (a) Discretização nos domínios de espaço e tempo, (b) Parâmetros da discretização por Volumes Finitos. Fonte: Rusche (2002).

Geralmente, as variáveis dependentes são alocadas no centro dos volumes (arranjo co-localizado). A maioria dos problemas numéricos é resolvida com técnicas de arranjo co-localizado, inclusive no OpenFOAM devido à sua simplicidade de controle dos índices da malha na implementação computacional.

O objetivo da discretização é transformar uma ou mais equações em um sistema de equações algébricas, (RUSCHE, 2002). A solução deste sistema aproxima-se da solução das equações originais em alguns locais pré-determinados no espaço e no tempo. Partindo de uma equação de transporte para uma variável φ tensorial genérica:

$$\underbrace{\frac{\partial \rho \varphi}{\partial t}}_{\text{derivada no tempo}} + \underbrace{\nabla \cdot (\rho \mathbf{u} \varphi)}_{\text{termo convectivo}} = \underbrace{\nabla \cdot (\Psi \nabla \varphi)}_{\text{termo difusivo}} + \underbrace{S_\varphi(\varphi)}_{\text{termo fonte}} \quad (4.2)$$

onde ρ é a massa específica, \mathbf{u} é o campo de velocidade e Ψ é o coeficiente de difusão e $S_\varphi(\varphi)$ representa o termo fonte. Os termos da equação de transporte representam a taxa de mudança por unidade de volume (derivada no tempo), o fluxo por convecção por unidade de volume (termo convectivo), a taxa de transporte devido à difusão (termo difusivo) e da taxa de produção/destruição por unidade de volume (termo fonte). Sua discretização por volumes finitos é obtida integrando sobre um volume de controle V_P e um intervalo de tempo, Δt , como:

$$\int_t^{t+\Delta t} \left[\int_{V_P} \frac{\partial(\rho \varphi)}{\partial t} dV + \int_{V_P} \nabla \cdot (\rho \mathbf{u} \varphi) dV \right] dt =$$

$$\int_t^{t+\Delta t} \left[\int_{V_P} \nabla \cdot (\Gamma \nabla \boldsymbol{\varphi}) dV + \int_{V_P} S_\varphi(\boldsymbol{\varphi}) dV \right] dt \quad (4.3)$$

O teorema de Gauss é usado para converter as integrais de volume em integrais sobre a superfície S , no contorno do volume de controle, da seguinte forma:

$$\int_{V_P} (\nabla \cdot \boldsymbol{\varphi}) dV = \int_S \boldsymbol{\varphi} \cdot d\mathbf{S} \quad (4.4)$$

Desta forma a Eq. (4.2) pode ser reescrita como:

$$\begin{aligned} & \int_t^{t+\Delta t} \left[\left(\frac{\partial(\rho\boldsymbol{\varphi})}{\partial t} \right)_P V_P + \sum_f F_{\boldsymbol{\varphi}f(F,M,\gamma)} \right] dt \\ &= \int_t^{t+\Delta t} \left[\sum_f \Gamma_f(\mathbf{S} \cdot \nabla_f \boldsymbol{\varphi}) + (S_{I\varphi P} + S_E) V_P \right] dt \end{aligned} \quad (4.5)$$

O termo fonte $S_\varphi(\boldsymbol{\varphi})$ mostrado na Eq. (4.1), pode ser uma função genérica de $\boldsymbol{\varphi}$ e pode ser linearizado antes da discretização, como:

$$S_\varphi(\boldsymbol{\varphi}) = S_I \boldsymbol{\varphi} + S_E \quad (4.6)$$

onde S_I e S_E podem ser dependentes de $\boldsymbol{\varphi}$ que integrado no volume de controle resulta em:

$$\int_V S_\varphi(\boldsymbol{\varphi}) dV = S_I \boldsymbol{\varphi}_P V_P + S_E V_P \quad (4.7)$$

Ao assumir que os volumes de controle não variam com o tempo, a Eq. (4.3) pode ser reescrita como

$$\begin{aligned} & \int_t^{t+\Delta t} \left[\left(\frac{\partial(\rho\boldsymbol{\varphi})}{\partial t} \right) V_P + \sum_f F_{\boldsymbol{\varphi}f(F,M,\gamma)} \right] dt \\ &= \int_t^{t+\Delta t} \left[\sum_f \Psi_f(\mathbf{S} \cdot \nabla_f \boldsymbol{\varphi}) + (S_{I\varphi P} + S_E) V_P \right] dt \end{aligned} \quad (4.8)$$

A exatidão do método de discretização depende da variação assumida para a função $\boldsymbol{\varphi}$, no espaço e no tempo, no ponto P . A fim de obter um método preciso de segunda ordem, esta variação deve ser linear no espaço e no tempo. A discretização da derivada no tempo mostrada na Eq. (4.7) e a integral no tempo podem ser calculadas diretamente como:

$$\left(\frac{\partial(\rho\boldsymbol{\varphi})}{\partial t}\right)_P = \frac{\rho_P^n \boldsymbol{\varphi}_P^n - \rho_P^0 \boldsymbol{\varphi}_P^0}{\Delta t} \quad (4.9)$$

$$\int_t^{t+\Delta t} \boldsymbol{\varphi}(t) dt = \frac{1}{2} (\boldsymbol{\varphi}^0 + \boldsymbol{\varphi}^n) \Delta t \quad (4.10)$$

onde $\boldsymbol{\varphi}^n = \boldsymbol{\varphi}(t + \Delta t)$ representa o novo valor de $\boldsymbol{\varphi}$ para passo de tempo a ser resolvido enquanto $\boldsymbol{\varphi}^0 = \boldsymbol{\varphi}(t)$ é o valor no passo anterior. Utilizando as Equações (4.8), (4.9) e (4.10), obtem-se a forma discretizada, da Eq. (4.2), por:

$$\begin{aligned} \rho_P \frac{\boldsymbol{\varphi}_P^n - \boldsymbol{\varphi}_P^0}{\Delta t} V_P + \sum_f [(1 - W) F_f^n \boldsymbol{\varphi}_f^n + w F_f^0 \boldsymbol{\varphi}_f^0] \\ - \sum_f [(1 - W) \Gamma_f^n \mathbf{S} \cdot (\nabla_f \boldsymbol{\varphi})^n + w \Gamma_f^0 \mathbf{S} \cdot (\nabla_f \boldsymbol{\varphi})^0] \\ = (1 - w) S_E^n V_P + w S_E^0 V_P + (1 - w) S_I^n \boldsymbol{\varphi}_P^n V_P + w S_I^0 \boldsymbol{\varphi}_P^0 V_P \end{aligned} \quad (4.11)$$

Para diferentes valores de W , vários métodos de integração no tempo podem ser usados. Por exemplo, o método de Euler explícito de primeira ordem é obtido para $W = 1$ e o de segunda ordem de Crank-Nicholson para $W = 1/2$. Os valores nas faces são calculados a partir dos valores no centro dos volumes em cada lado da face e usando métodos adequados para a interpolação dos termos convectivos e difusivos. Neste trabalho foi usado o método de Euler implícito para simulação de bolhas ascendendo em um meio líquido e o método de Crank-Nicholson para a dinâmica de gotas escoando em microcanais.

Como $\boldsymbol{\varphi}_f$ e $\nabla_f \boldsymbol{\varphi}$ também dependem do valor de $\boldsymbol{\varphi}$ nos volumes vizinhos, a Eq. (4.11) pode ser representada como uma equação algébrica válida para o volume P .

$$a_P \boldsymbol{\varphi}_P^n + \sum_N a_N \boldsymbol{\varphi}_N^n = b_P \quad (4.12)$$

A Equação anterior pode ser escrita como um sistema algébrico de equações, considerando todos os volumes de controle da malha, como:

$$\mathcal{A} \boldsymbol{\varphi} = \mathbf{b} \quad (4.13)$$

onde \mathcal{A} é uma matriz esparsa com coeficientes diagonais a_P e não-diagonais a_N , enquanto que $\boldsymbol{\varphi}$ e \mathbf{b} são, respectivamente, os vetores das variáveis e os termos fontes referentes ao centro dos volumes. A matriz \mathcal{A} pode ser decomposta em duas matrizes contendo apenas os coeficientes diagonais \mathcal{D} e os não-diagonais \mathcal{N} , tal que:

$$\mathcal{A} = \mathcal{D} + \mathcal{N} \quad (4.14)$$

A estrutura da matriz \mathcal{A} é um parâmetro importante na escolha do método mais apropriado para resolver o sistema linear e pode variar dependendo da dimensão do problema, da forma de discretização das equações e da ordenação dos volumes de controle na malha. Em aplicações CFD, cerca de 20% do tempo de computação necessário para resolver um dado problema está associado à solução do sistema linear e 80% à obtenção dos parâmetros das equações discretizadas (LOPES, 2008).

O método de discretização padrão adotado pelo OpenFOAM é a integração de Gauss para volumes finitos. A integração de Gauss é baseada na soma dos fluxos das variáveis nas faces do volume, que devem ser interpolados a partir do centro dos volumes. O usuário pode escolher livremente o tipo de método de interpolação a ser usado. Dentre estes, pode-se citar as abordagens de interpolação linear (diferenças centrais), *upwind*, QUICK, MUSCL, TVD (*Total Variation Diminishing*) e NV (*Gamma Normalised Variable*). Na discretização no tempo, o OpenFOAM dispõe dos métodos de Euler explícito, implícito e Crank-Nicholson implementados em seu código. Também dispõe dos métodos Gauss-Seidel, AMG (*Algebraic Multi Grid*) e variantes do gradiente conjugado como ICCG (*Incomplete Cholesky Conjugated Gradients*), DCG (*Deflated Conjugate Gradient*) e BICG (*Bi-Conjugate Gradient*) como os métodos de solução de sistemas lineares.

Os métodos de solução do sistema de equações algébricas implementados no OpenFOAM são iterativos e portanto baseiam-se em reduzir o resíduo das equações até um valor pré-estabelecido. O método de solução é interrompido quando o resíduo torna-se menor do que a tolerância especificada (*tol*) e a razão entre os resíduos da iteração atual e inicial for menor do que a tolerância relativa (*relTol*).

4.4 Acoplamento pressão–velocidade

A solução das Equações (3.1) e (3.2) é feita de forma segregada, o que leva à necessidade de tratar o acoplamento pressão-velocidade. Os métodos de Projeção, ou métodos de Passo Fracionário, têm origem nas observações de Chorin (1968) sobre o papel da pressão em escoamentos incompressíveis. Chorin (1968) constatou que a pressão não desempenha papel termodinâmico, mas força a condição de incompressibilidade, o que leva à uma discretização baseada na separação de operadores. Nessa estratégia, velocidade e pressão são determinadas em dois passos. No primeiro, um campo de velocidades auxiliar \mathbf{u}' é calculado com a equação de balanço da quantidade de movimento, desprezando-se a condição de

incompressibilidade. No segundo, o campo de velocidades auxiliar \mathbf{u}' é projetado no espaço dos campos vetoriais com divergente nulo para calcular a pressão ou a correção de pressão. A pressão corrigida possibilita a atualização do campo de velocidades.

O problema associado com a solução do conjunto de equações e acoplamento pressão-velocidade pode ser resolvido com uma estratégia de solução iterativa com os diferentes algoritmos existentes tais como o SIMPLE (*Semi-Implicit Method for Pressure-Linked Equations*) (PATANKAR e SPALDING, 1972). O algoritmo é baseado na hipótese de que a velocidade obtida pela solução da equação de conservação de quantidade de movimento, só irá satisfazer a equação de conservação da massa, se o campo de pressão estiver correto. Conseqüentemente, é preciso corrigir tanto a velocidade como a pressão. No caso de problemas transientes é recomendado usar o algoritmo PISO (*Pressure-Implicit with Splitting of Operators*) desenvolvido por Issa (1986). O PISO envolve um passo preditor e dois passos de correção para satisfazer a conservação de massa. Um outro algoritmo, o PIMPLE é derivado dos algoritmos PISO e SIMPLE e possui a particularidade de efetuar ciclos múltiplos de iterações, no mesmo intervalo de tempo, usando o valor da última iteração como estimativa inicial para a próxima iteração. Essa particularidade é denominada *Outler Correction Loops*.

Os algoritmos PISO, SIMPLE e PIMPLE estão implementados no OpenFOAM para resolver o acoplamento pressão-velocidade presente nas equações de escoamentos multifásicos. No presente trabalho, os escoamentos bifásicos simulados são transientes e, portanto o algoritmo PIMPLE é utilizado. Neste caso em cada passo de tempo, o *solver* usa o algoritmo SIMPLE para controlar a convergência combinado com o algoritmo PISO para ajustar a correção de pressão.

4.5 Solvers disponíveis no OpenFOAM

O OpenFOAM oferece um conjunto de *solvers* que evoluíram no decorrer do tempo e que estão disponíveis para os usuários e tem sido utilizado principalmente para CFD. Mas tem também encontrado uso em outras áreas, como análise de esforços, eletromagnetismo e nas finanças, já que é fundamentalmente um instrumento para a resolução de equações diferenciais parciais, em vez de um pacote CFD no sentido tradicional. No caso multifásico, o OpenFOAM oferece uma lista de *solvers* disponíveis para os usuários. Na Tabela 4.1 apresentam-se alguns dos *solvers* disponíveis na versão 2.4 do OpenFOAM, para aplicações em escoamentos multifásicos, com uma breve descrição de cada um.

Os dois *solvers* *interFoam* e *interDyMFoam* foram usados neste trabalho e são apresentados na Tabela 4.1. Ambos já estavam implementados no código, são adequados para análise de escoamentos multifásicos e são baseados na metodologia VOF, para acompanhar a evolução temporal da interface entre os fluidos ou fases. O solver *interDyMFoam* tem as mesmas características matemáticas do *interFoam* mas possui a capacidade de refinamento dinâmico da malha e será explicado com mais detalhe no Item 4.7.

Tabela 4.1: Solvers multifásicos disponíveis no OpenFOAM.

Solvers Multifásicos	Descrição
<i>cavitatingFoam</i>	Para cavitação transiente com base no modelo de equilíbrio homogêneo a partir da qual a compressibilidade do líquido/vapor da mistura é obtido.
<i>compressibleInterFoam</i>	Para 2 fluidos compressíveis, isotérmicos, imiscíveis usando VOF.
<i>compressibleInterDyMFoam</i>	Mesma descrição anterior com refinamento dinâmico.
<i>compressibleMultiphaseInterFoam</i>	Para n fluidos compressíveis que capturam as interfaces e inclui tensão superficial e os efeitos de contato angular para cada fase.
<i>interFoam</i> (Usado neste trabalho com VOF e S-CLSVOF)	Para 2 fluidos incompressíveis, isotérmicos, imiscíveis usando VOF.
<i>InterDyMFoam</i> (Usado neste trabalho com VOF)	Mesma descrição anterior com refinamento dinâmico.
<i>interMixingFoam</i>	Para 3 fluidos incompressíveis, dois dos quais são miscíveis, utilizando VOF.
<i>interPhaseChangeFoam</i>	Para 2 fluidos incompressíveis, isotérmicos e imiscíveis com mudança de fase utilizando VOF.
<i>interPhaseChangeDyMFoam</i>	Mesma descrição anterior com refinamento dinâmico.
<i>LTSinterFoam</i>	Para 2 fluidos incompressíveis, isotérmicos e imiscíveis no estado estacionário utilizando VOF.
<i>multiphaseEulerFoam</i>	Para um sistema de muitas fases de fluidos compressíveis, incluindo transferência de calor.
<i>multiphaseInterFoam</i>	Para n fluidos incompressíveis que capturam as interfaces e inclui tensão superficial e os efeitos de ângulo de contato para cada fase.
<i>multiphaseInterDyMFoam</i>	Mesma descrição anterior com refinamento dinâmico.
<i>twoLiquidMixingFoam</i>	Para a mistura de 2 fluidos incompressíveis.
<i>twoPhaseEulerFoam</i>	Para um sistema de 2 fases com um fluido incompressível em fase dispersa.

4.6 Estrutura de um caso no OpenFOAM

O usuário do OpenFOAM possui total liberdade para criar ou modificar um *solver* ou código específico. Outro benefício do OpenFOAM é que por se tratar de um programa escrito na linguagem de programação C++, oferece uma grande flexibilidade para edição. Como foi mencionado anteriormente, a estrutura básica do programa divide-se em três categorias: *solvers*, utilitários e bibliotecas.

Qualquer caso simulado requer definir uma série de pastas que contém informações referentes aos *solvers* e utilitários. Um esquema dessas pastas é mostrado na Fig. 4.2, ilustrando um caso a ser estudado, denominado *case*.

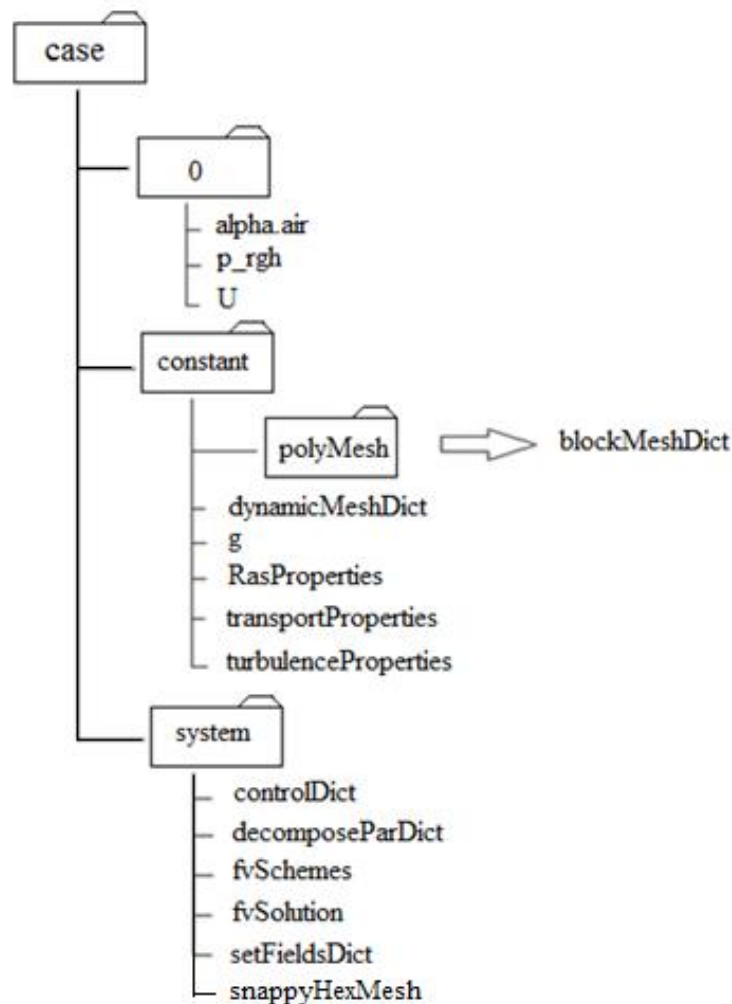


Figura 4.2: Esquema de pastas que formam um caso no OpenFOAM.

- Pasta **0**: Contém os arquivos individuais de dados para os campos das variáveis como pressão, velocidade e fração volumétrica. Os dados podem ser tanto condições iniciais como condições de contorno que o usuário especifica para definir o problema.

- Pasta **constant**: Contém as informações da malha do caso em uma subpasta chamado *polyMesh* e arquivos que especificam as propriedades físicas dos materiais/fluidos envolvidos na simulação, parâmetros do refinamento dinâmico e dos modelos de turbulência se for o caso.
- Pasta **system**: Os arquivos contidos neste diretório estão associados com o procedimento de solução do caso. Aqui são definidos os parâmetros de controle da simulação, os esquemas de discretização, os métodos de solução dos sistemas de equações e as informações relacionadas com as fases presentes.

No Apêndice 1 é apresentado com maior detalhe o conteúdo dos arquivos que compõem estas pastas. É importante ressaltar que o OpenFOAM pode usar qualquer sistema de unidades, neste trabalho usou-se o Sistema Internacional (SI).

A liberdade do acesso ao código fonte no OpenFOAM fornece ao usuário a oportunidade de analisar e estudar os algoritmos e modelos para ajustá-lo a um problema específico. Contudo, o usuário do OpenFOAM que quiser alterar algum código deve possuir conhecimentos de C++ e ler os manuais disponíveis na página oficial além de saber trabalhar com a plataforma linux, embora, as últimas versões (V3.0, V4.0 e V5.0) podem ser instaladas também no Windows. No Capítulo 4 do trabalho de Favero (2009) é apresentada uma explicação detalhada do *software* OpenFOAM tanto para usuários iniciantes quanto para desenvolvedores, ou seja, que desejam modificar o código. No Apêndice 2 algumas das pastas mais importantes do código fonte são detalhadas assim como a forma de acesso do caso que se deseja fazer modificações.

4.7 Descrição dos solvers *interFoam* e *interDyMFoam*

O solver *interFoam* é usado para escoamentos com a presença de dois fluidos imiscíveis, incompressíveis e é baseado no método de captura VOF (Volume de Fluido). Nesta metodologia é usado um passo de tempo adaptativo dependendo do número de Courant $Co = |\mathbf{u}| \delta t / \delta x$. Para escolher o novo passo de tempo, um número máximo de Courant Co^0 é calculado a partir das condições de fluxo, usando \mathbf{u} e δt do intervalo de tempo anterior. O novo intervalo do tempo δt^n é obtido usando a seguinte expressão (BANAIBEDALRUHMAN, 2015)

$$\delta t^n = \min \left\{ \frac{Co_{max}}{Co^0} \delta t^0; \left(1 + 0,1 \frac{Co_{max}}{Co^0} \right) \delta t^0; 1,2 \delta t^0; \delta t_{max} \right\} \quad (4.15)$$

na qual δt^0 é o passo de tempo do instante anterior, Co_{max} é o número de Courant máximo e δt_{max} é o passo de tempo máximo, ambos são especificados no arquivo da pasta *system/controlDict* (Fig. 4.2).

É fundamental conhecer o valor da fração volumétrica α , pois esse parâmetro afeta as propriedades físicas tais como massa específica, viscosidade, bem como a curvatura da interface. Assim, a equação da fração volumétrica α é resolvida em subciclos em cada intervalo do tempo. O novo intervalo do tempo desse subciclo é calculado a partir do passo de tempo para o fluxo a partir da equação

$$\delta t_{sc} = \frac{\delta t}{nAlphaSubCycles} \quad (4.16)$$

na qual *nAlphaSubCycles* é o número de sub-ciclos definidos no arquivo da pasta *system/fvSolution* (Fig. 4.2).

O *solver interDyMFOam* é osemelhante ao *interFoam*, mas com a capacidade de refinamento dinâmico de malha. Os parâmetros para o refinamento dinâmico (Anexo A), são definidos no arquivo *dynamicMeshDict* da pasta *constant* (Fig. 4.2). A vantagem deste *solver* é uma maior precisão na forma da interface entre os fluidos. Na Figura 4.3 mostra-se a malha gerada pelos dois *solvers* e a diferença na forma da interface, para um mesmo instante de tempo, de um caso de uma bolha ascendendo num meio líquido.

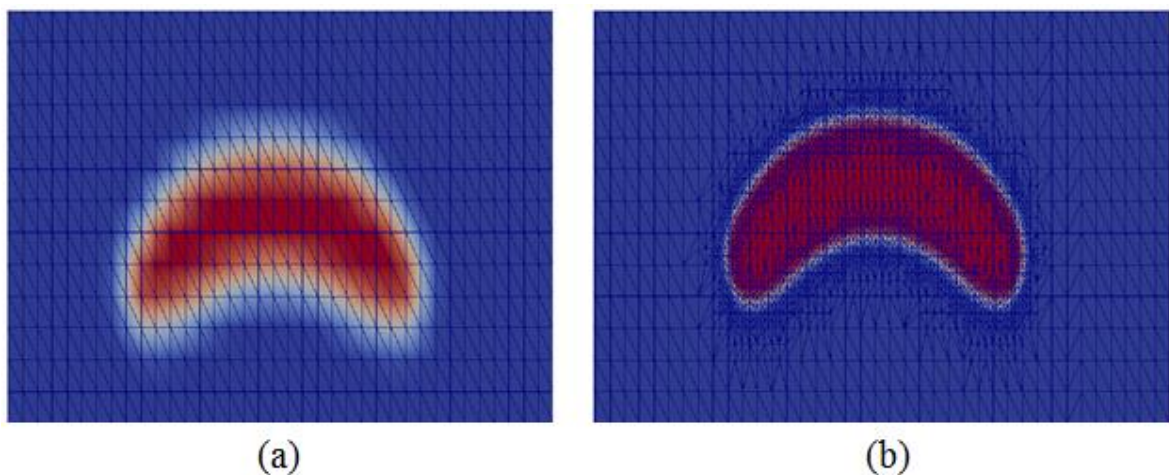


Figura 4.3: Comparação da forma de uma bolha ascendendo num meio líquido com os *solvers* (a) *interFoam* e (b) *interDyMFOam*.

No Apêndice 3 é apresentado com maior detalhe a forma com que o *solver interDyMFOam* faz o refinamento em células hexaédricas (3D) e os arquivos do código fonte relacionados com esta função.

4.8 Implementação do acoplamento VOF e LS no OpenFOAM

Como já foi dito no Capítulo 1, um dos objetivos deste trabalho foi a implementação do acoplamento S-CLSVOF no OpenFOAM.

Neste trabalho o *solver* foi denominado S-CLSVOF, de forma análoga à do algoritmo proposto por Albadawi *et al.* (2013). O algoritmo foi implementado no *solver interFoam* e foi adaptado aos problemas estudados. As funcionalidades dos arquivos do código fonte do *solver interFoam* com a metodologia VOF e os arquivos novos incluídos com a metodologia S-CLSVOF são apresentados no Apêndice 4.

4.9 Pós-processador ParaView

O *ParaView* é um programa multi-plataforma, paralelo e *open-source*, projetado para análise de dados e visualização científica, baseado no VTK que é um formato de arquivos utilizado em diferentes programas. O *ParaView* suporta uma grande variedade de tipos de dados como malhas estruturadas, não estruturadas e imagens 2D e 3D. Possui diferentes formas de visualização tais como isosuperfícies, cortes, linhas de corrente, vetores, animações, etc. É possível converter os resultados obtidos pelo OpenFOAM para formatos lidos por outros software como FLUENT, Tecplot, Fieldview, Ensign e GMV, utilizando ferramentas fornecidas junto com o OpenFOAM. Existe ainda uma ferramenta de conversão dos resultados do OpenFOAM para o formato VTK possibilitando a leitura dos dados em qualquer visualizador que use VTK.

Neste trabalho foram feitas simulações de bolhas ascendendo em meios líquidos e gotas escoando em microcanais e o cálculo da velocidade baricêntrica tanto da bolha quanto da gota foi usando o *ParaView*. Isto pode ser feito pela seleção da guia *filters* e a seleção *calculator*, onde é introduzida a equação para o cálculo da velocidade baricêntrica, definida como

$$U_b = \frac{\int_V (U_y \cdot e_z) dV}{\int_V (\alpha_g) dV} \quad (4.17)$$

na qual U_b é a velocidade baricêntrica, U_y é a velocidade da fase gasosa na direção vertical e α_g é a fração volumétrica da fase gasosa. Neste cálculo é aplicada estatística descritiva disponível em *filters*. A velocidade máxima pode ser lida em qualquer etapa na janela do lado direito correspondente à matriz de velocidade. Este cálculo é ilustrado na Figura 4.4 por meio da captura da tela do *ParaView*.

No Capítulo 5 são apresentados os resultados das simulações feitas neste trabalho relacionadas com ascensão de bolhas em meios líquidos e gotas escoando em microcanais com junção em T .

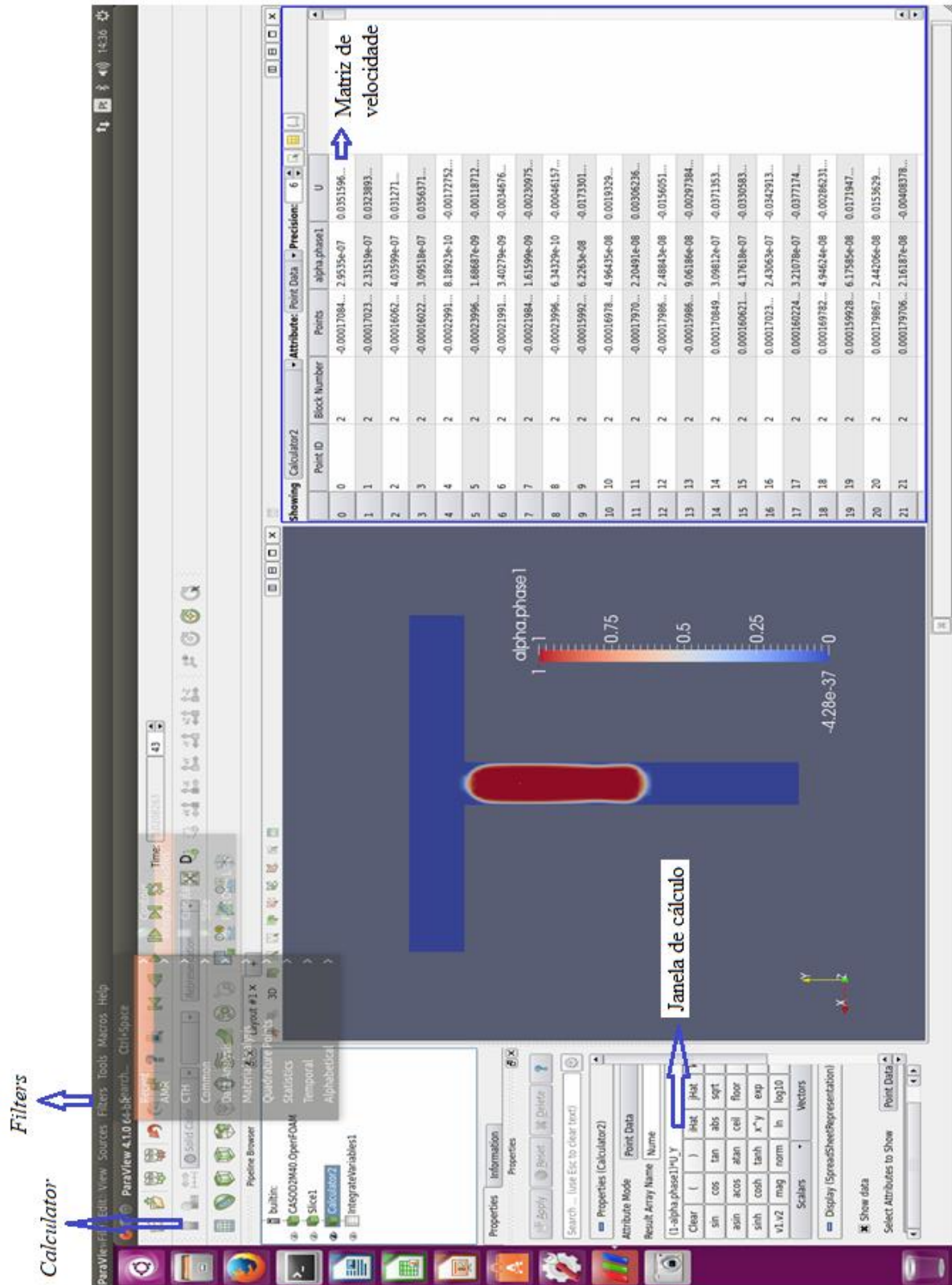


Figura 4.4: Tela de cálculo da velocidade baricêntrica utilizando ParaView.

Capítulo 5

Resultados e Discussões

São apresentados neste capítulo os resultados das simulações numéricas realizadas neste trabalho. As simulações foram realizadas em um computador com processador intel Core i7-3450 com 3,10 GHz, com 12 GB de memória RAM DDR3, disponível no Laboratório de Transferência de Calor (LabTC), da UNIFEI. Também foi usado no caso de simulação de gotas em microcanais um Cluster do Laboratório de Astrofísica Computacional da UNIFEI composto por seis máquinas com 40 processadores por máquina e 96842 MB de memória RAM, por máquina (Anexo B).

Antes de iniciar as simulações, vários casos testes, disponíveis no tutorial do OpenFOAM foram realizados. Com isto, foi possível adquirir domínio na utilização do programa, assim como na etapa de pós-processamento realizada no *software ParaView*.

Foram simuladas bolhas ascendendo em meios líquidos com o objetivo de estudar a influência das propriedades dos fluidos na forma final da bolha e calcular sua velocidade terminal. Posteriormente foi simulada a dinâmica de uma gota escoando através de um microcanal com junção em T, com o objetivo de estudar os mecanismos de ruptura/não ruptura da gota através da junção.

Para os dois problemas simulados foram usados os *solvers interFoam* e S-CLSVOF com a metodologia VOF com o objetivo de estabelecer comparações entre eles. Todos os resultados foram comparados com resultados encontrados na literatura.

A seguir são apresentados os resultados das simulações da ascensão de uma bolha em um meio líquido em repouso.

5.1 Ascensão de uma bolha

No caso de bolhas ascendendo em meios líquidos foram feitas simulações para determinar tanto a velocidade da bolha quanto sua forma final. Uma representação da condição inicial da bolha no domínio computacional pode ser vista na Figura 5.1. No caso do *solver interDyMFoam* com a metodologia VOF foi preciso estudar o efeito dos parâmetros do arquivo *dynamicMeshDict*. Em seguida foram estudados os efeitos das dimensões (x,y,z) do domínio computacional e das condições de contorno: livre deslizamento e não deslizamento. A velocidade terminal em cada caso foi determinada pelo cálculo da velocidade baricêntrica. Foram também estudados os efeitos dos números de Eötvös, Mortom e Reynolds, validando cada caso com os resultados encontrados na bibliografia consultada (BHAGA e WEBER, 1981;

HUA e LOU, 2007; GEORGE, REZENDE e MALISKA, 2008). Posteriormente foram simulados diferentes casos para estudar o efeito da razão de massas específicas (ρ_l/ρ_g) e viscosidades (μ_l/μ_g) sobre a velocidade terminal.

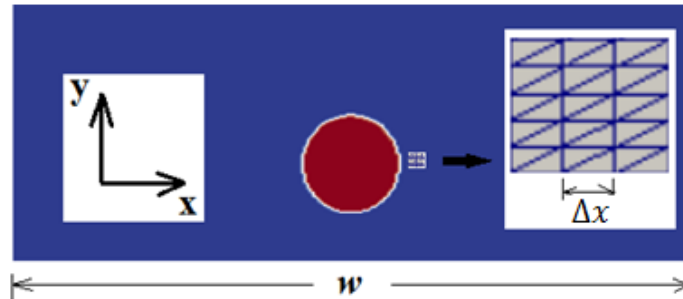


Figura 5.1: Vista da bolha no instante inicial dentro do domínio computacional em corte no plano xy e uma ampliação da malha gerada.

Todos os resultados foram visualizados pelo *software ParaView*, no qual foi calculada a velocidade terminal da bolha e foi analisada sua forma final.

5.1.1 Efeito dos parâmetros de refinamento.

Nestas primeiras simulações foi estabelecida a condição de livre deslizamento na parede. O diâmetro da bolha usado foi de $D = 0,0261$ m sendo o mesmo valor utilizado no trabalho de George, Rezende e Maliska, (2008). As propriedades físicas utilizadas na simulação foram obtidas de Bhaga e Weber (1981) e estão apresentadas na Tabela 5.1. Para estes testes o tamanho do domínio computacional escolhido foi de $w = 0,2$ m nas três dimensões. O tamanho inicial da malha é $\Delta x/w = 0,0083$ (Fig. 5.1).

Tabela 5.1: Propriedades físicas dos fluidos para os testes de refinamento.

$\rho_l \left(\frac{kg}{m^3} \right)$	$\rho_g \left(\frac{kg}{m^3} \right)$	$\mu_l \left(\frac{kg}{m \cdot s} \right)$	$\mu_g \left(\frac{kg}{m \cdot s} \right)$	$\sigma \left(\frac{N}{m} \right)$
1314	1,185	1,2536	$1,766 \times 10^{-5}$	0,0769

O refinamento dinâmico com o *solver interDyMFoam* foi realizado com o intuito de representar melhor a interface da bolha. No arquivo *dynamicMeshDict* (Anexo B) foi preciso definir alguns parâmetros. No início é importante definir qual é o campo a ser refinado, que neste trabalho é a fração volumétrica α chamada no código de *alpha.phase1*, a qual corresponde a bolha. Em seguida foi preciso definir o número máximo de células no domínio e limitar o refinamento sobre o contorno da bolha, utilizando o parâmetro *maxCells*. Em geral este número deve ser superior ao valor total de células em todo o domínio computacional, no início da simulação, onde a malha base é uniforme.

Neste caso foi preciso definir o termo “relação de refinamento” (RR), como o cociente entre o número máximo de células no refinamento dinâmico ($maxCells$, vide Anexo A) e o número total de células em todo o domínio computacional. Nestas primeiras simulações foram definidas 120 células em cada direção, ou seja, 1728000 células totais. Foram simulados 3 casos, com $RR = 2,9$; 5,2 e 8,7. Por exemplo, no caso de $RR = 2,9$ o $maxCell = 2,9 \times 1728000$, ou seja, no contorno da interface pode ser construído até um máximo de 5011200 de células.

Na Figura 5.2 apresenta-se uma relação entre a velocidade baricêntrica adimensional da bolha ($U_b^* = U_b/\sqrt{gD}$) e o tempo adimensional ($\tau^* = t/\sqrt{D/g}$) para os três casos simulados e na Fig. 5.3 apresenta-se a influência sobre a forma da bolha. Como é observado nessas figuras, o RR afeta pouco a velocidade e a forma final da bolha. Portanto as demais simulações foram feitas com $RR = 2,9$, dado que a medida que o RR aumenta, o tempo da simulação também aumenta.

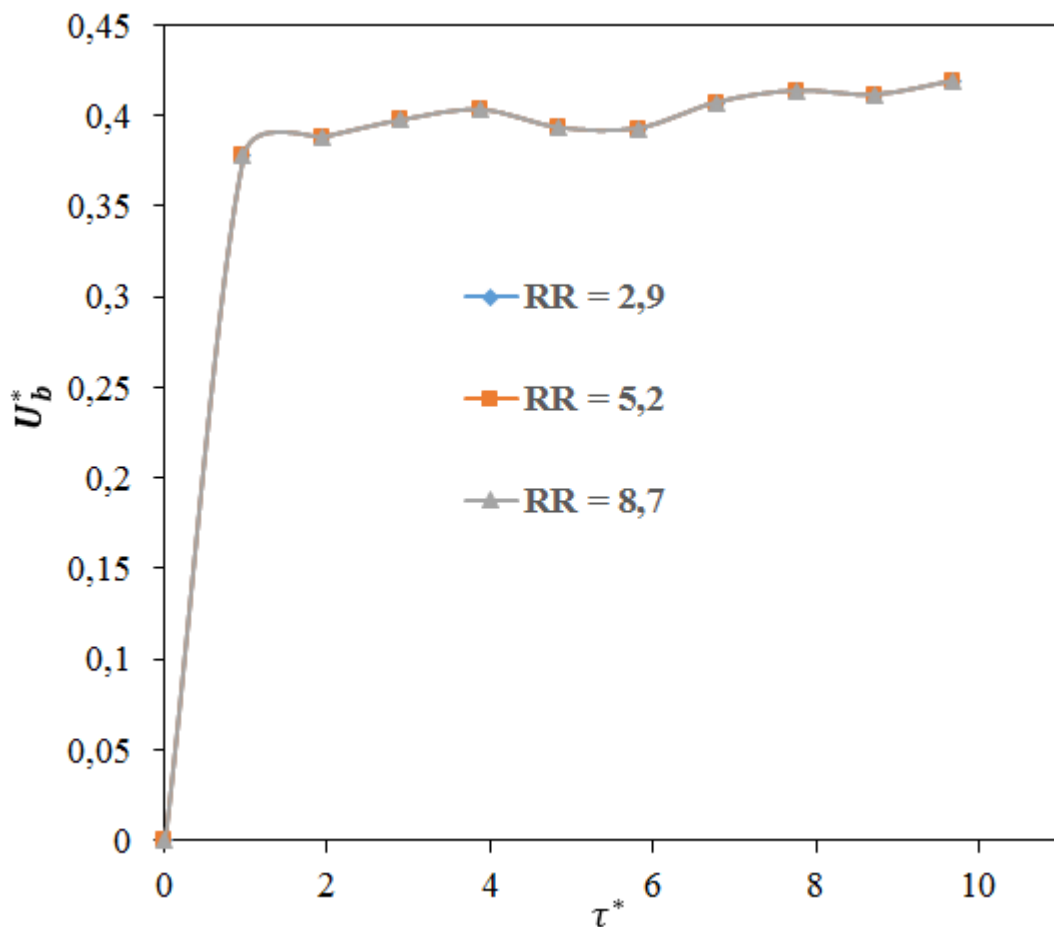


Figura 5.2: Influência da relação do refinamento sobre a velocidade da bolha.

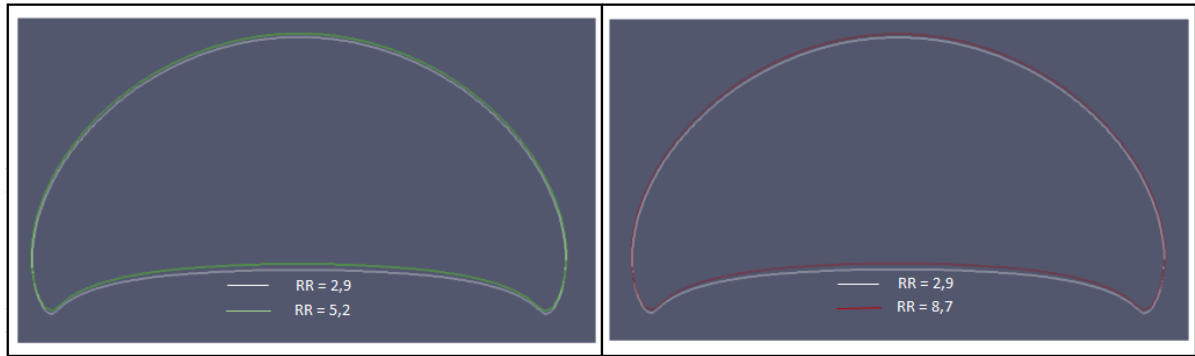


Figura 5.3: Influência da relação do refinamento sobre a forma da bolha.

Na Figura 5.4 ilustra-se, em corte, para o instante de tempo de 0,5 s, o refinamento dinâmico gerado sobre a bolha com $RR = 2,9$. Este refinamento é feito nas regiões de maiores gradientes da função fração volumétrica, α . O passo de tempo inicial utilizado para todos os casos estudados foi de 10^{-4} segundos, porém este valor é recalculado durante a simulação de acordo com o critério de estabilidade ou número de Courant definido no arquivo *controlDict* (Anexo C). O número de Courant (Co) é uma variável local do domínio de fluido que representa o fluxo advectivo em cada volume finito.

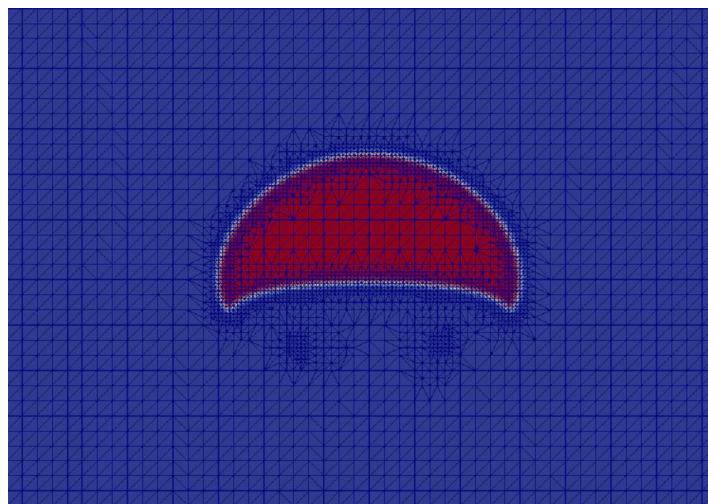


Figura 5.4: Refinamento dinâmico sobre a interface da bolha.

Outro parâmetro importante no arquivo *dynamicMeshDict* é o número *nBufferLayers*, denominado neste trabalho “nBL”. Esse parâmetro especifica quantas camadas a malha deve manter entre dois níveis de refinamento e é usado para suavizar as transições entre os tamanhos de célula e eliminar possíveis ondas de pressão artificiais devido à mudanças bruscas no tamanho da célula. Foram testados valores de nBL entre 1 e 7.

Na Figura 5.5 são mostrados os resultados da velocidade baricêntrica adimensional da bolha em função do tempo, para diferentes valores de nBL.

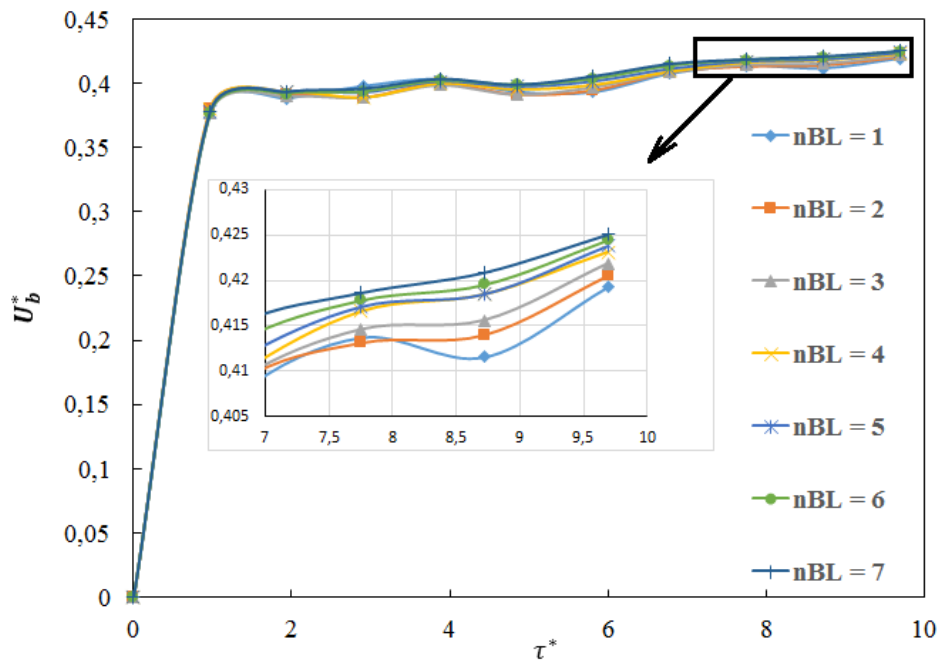


Figura 5.5: Influência do número nBufferLayers na velocidade da bolha.

Na Figura 5.6 apresenta-se a influência de nBL sobre a forma final da bolha. Observa-se a pouca influência deste parâmetro na velocidade terminal da bolha e na sua forma final. Portanto, um número de nBL=1 foi escolhido para todas as simulações feitas com o *solver interDyMFoam*, no caso de bolhas ascendendo em meios líquidos.

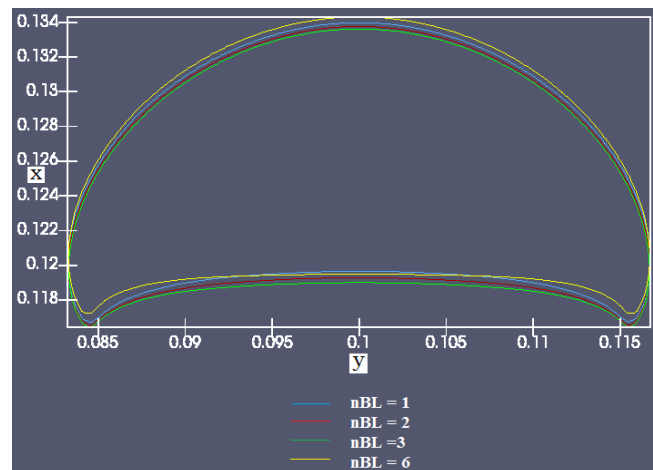


Figura 5.6: Influência do número nBufferLayers na forma final da bolha

5.1.2 Efeito das dimensões do domínio computacional, do tamanho da malha e das condições de contorno nas paredes.

Para analisar a influência do tamanho do domínio computacional, da malha e das condições de contorno, foi usado o *solver interDyMFoam*. Neste ponto foram utilizados quatro domínios com diferentes dimensões laterais, definidos em função do diâmetro da bolha (D),

como mostra a Tab. 5.2. O diâmetro da bolha é o mesmo usado anteriormente ($D = 0,0261 \text{ m}$). A dimensão vertical (eixo y) foi mantida constante em $7D$. Os primeiros 3 domínios foram definidos de acordo ao trabalho de George, Rezende e Maliska, (2008) com o objetivo de validar a metodologia usada. A malha computacional gerada inicialmente, no instante zero, é uniforme e a quantidade de células nas três direções para os quatro casos foram definidas para um tamanho de célula de $\Delta/w = 0,00625$ (ver Figura 5.1) e estão mostradas na tabela 5.2, porém, outros tamanhos de células foram simulados. Cada caso foi simulado com a condição de livre deslizamento e não deslizamento na parede. Na Figura 5.7 é mostrada a metade do domínio tridimensional.

Tabela 5.2: Tamanho dos domínios computacionais.

Caso	Domínio (x, y, z)	Número de células (x, y, z)
1	3D x 7D x 3D	(48,120,48)
2	4D x 7D x 4D	(60,120,60)
3	7D x 7D x 7D	(120,120,120)
4	10D x 7D x 10D	(157,120,157)

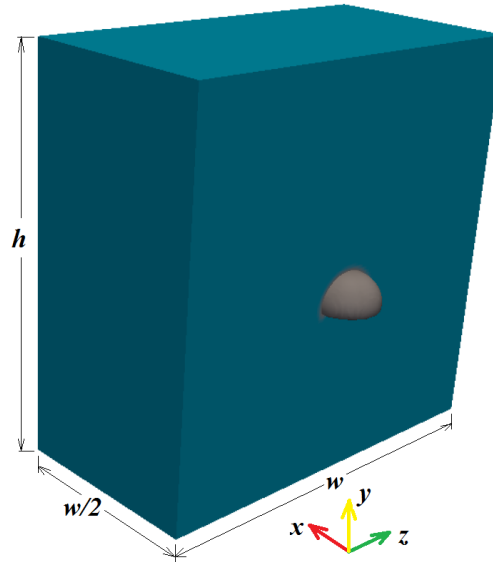


Figura 5.7: Domínio computacional em três dimensões para o Caso 3.

Como dito anteriormente o refinamento dinâmico é iniciado em $t=\Delta t$ e continua sendo realizado até o final da simulação à medida que a bolha ascende no meio líquido. A velocidade terminal em cada caso foi determinada segundo a Eq. (4.17). Na Figura 5.8 mostra-se a evolução temporal da velocidade adimensional da bolha, para os quatro domínios estudados.

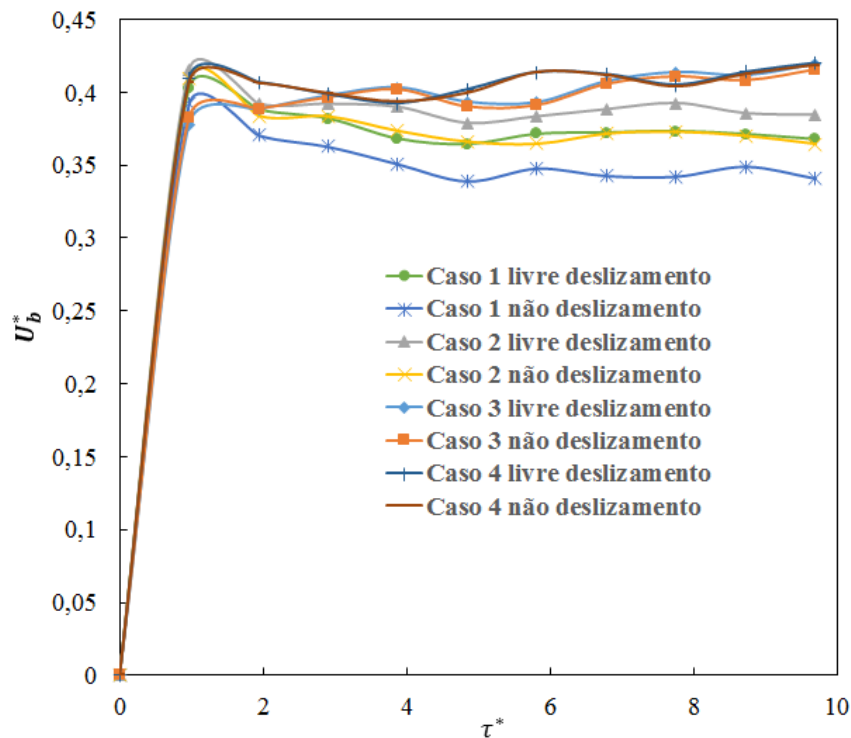


Figura 5.8: Evolução temporal da velocidade baricêntrica para os casos simulados com condição de não deslizamento e livre deslizamento na parede com o *solver* interDyMFoam.

Na Figura 5.8 observa-se que a velocidade baricêntrica adimensional da bolha é maior nos casos 3 e 4 e menor nos domínios menores, evidenciando um arrasto menor, como esperado. A condição de contorno na parede modifica o comportamento do arrasto, produzindo um valor maior para a condição de não deslizamento, gerando desta forma uma velocidade terminal menor. Na Figura 5.8 evidencia-se que nos casos 3 e 4 o efeito da parede não têm relevância, já que existe pouca diferença na velocidade adimensional com as condições de livre e não deslizamento. Os valores da velocidade terminal, (U_b), estão muito próximos aos reportados por George, Rezende e Maliska, (2008), para as mesmas condições de simulação, tal como mostra a Tab. 5.3.

Tabela 5.3: Valores obtidos da velocidade terminal.

Caso	Velocidade terminal (m/s) (George, Rezende e Maliska, 2008)	Velocidade terminal U_b (m/s) (Presente estudo)
1	0,167	0,184
2	0,188	0,197
3	0,243	0,212
4	-	0,206

Os resultados anteriores mostram que um domínio computacional com uma razão da largura (dimensão lateral) com respeito ao diâmetro da bolha de aproximadamente sete (7) tem

pouca influência nos resultados e foi o escolhido para as demais simulações tal como mostra a Fig. 5.9. A altura h depende do número de Reynolds calculado com a velocidade terminal e este valor variou de $7D$ a $12D$.

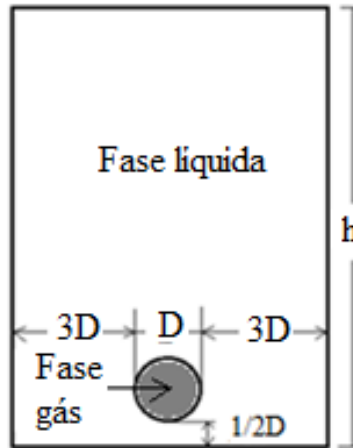


Figura 5.9: Dimensões do domínio computacional adotado neste trabalho.

Posteriormente foi estudado o efeito do refinamento da malha na forma e na velocidade da bolha. Neste caso foram escolhidos tamanhos de malha com $\Delta x/w = 0,025; 0,0125; 0,0083; 0,00625$ e $0,005$ (vide Fig. 5.1). Nas Figuras 5.10 e 5.11 observa-se que a partir de um tamanho de malha $\Delta/w = 0,00625$, existe pouca influência sobre a sua velocidade e a forma final da bolha.

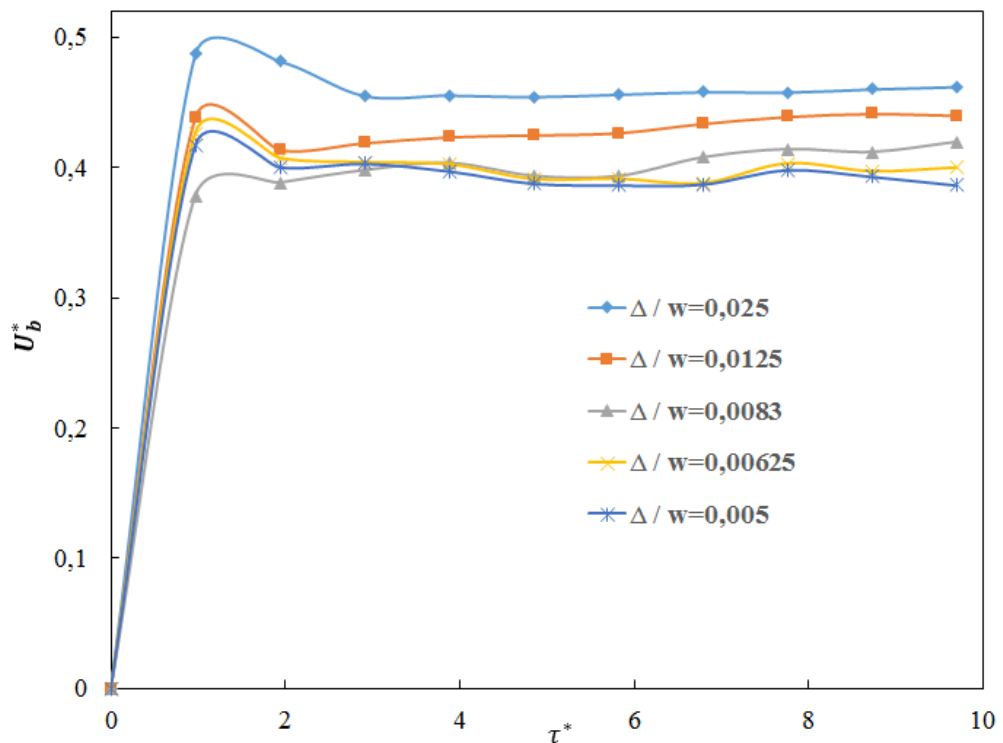


Figura 5.10: Influência do tamanho da malha na velocidade terminal para o caso 3.

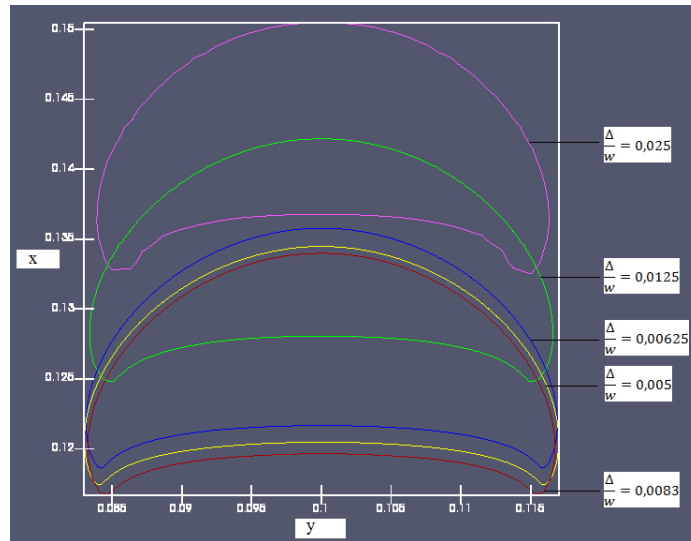


Figura 5.11: Influência do tamanho da malha na forma final da bolha.

Com objetivo de comparar os resultados anteriores obtidos com o *solver interDyMFoam* com os resultados obtidos com os *solvers interFoam* e S-CLSVOF, os mesmos quatro casos anteriores foram estudados. Estas simulações foram realizadas com a condição de livre deslizamento e tamanho de malha de $\Delta/w = 0,00625$. A Figura 5.12 mostra a evolução temporal da velocidade terminal adimensional (U_b^*). Observa-se que na região inicial, onde a bolha inicia a ascensão (τ^* de 0 a 1) as três metodologias têm comportamentos similares. Na região de acomodação (τ^* de 1 a 6) as metodologias apresentam valores de velocidade adimensional muito próximas e na última região ($\tau^* > 6$) as três metodologias têm comportamentos distintos.

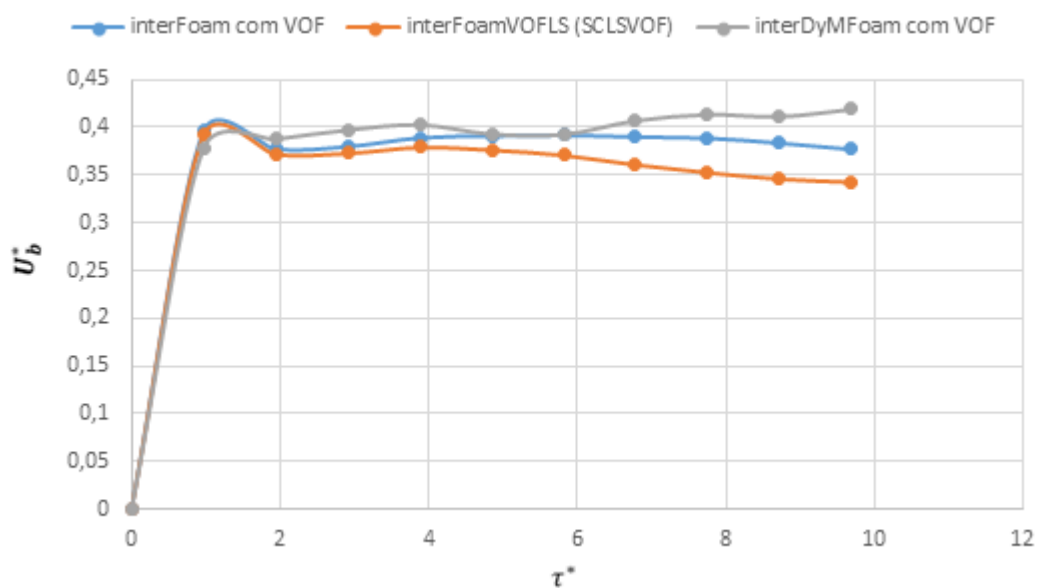


Figura 5.12: Evolução temporal da velocidade terminal adimensional da bolha com as três metodologias.

Na Figura 5.13 são apresentados os campos de fração volumétrica (α) obtidos com as três metodologias e o resultado obtido por George, Rezende e Maliska (2008). São mostrados três instantes de tempo, sendo o último correspondente à forma final da bolha.


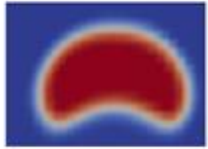














Metodologia	interFoam com VOF	interFoamVOFLS (SCLSVOF)	interDyMFoam com VOF	George, Rezende e Maliska (2008)
t=0,1s				
t=0,2s				
t=0,3s				
t=0,4s				

Figura 5.13: Comparação das formas das bolhas obtidas com as três metodologias estudadas com o resultado de George, Rezende e Maliska (2008).

Observa-se na Fig. 5.13 uma grande similaridade da forma da bolha para todos os resultados, porém, o refinamento dinâmico com o *solver interDyMFoam* define melhor a forma da interface, além de apresentar melhores resultados de velocidade terminal mais próximas da literatura. Portanto, nas simulações subsequentes de bolhas, será usado apenas o *solver interDyMFoam*.

No próximo item são apresentados os resultados das simulações feitas com o objetivo de estudar a influência da variação dos números adimensionais Eötvös, Morton e Reynolds sobre a forma final da bolha e sobre o valor da velocidade terminal.

5.1.3 Efeito da variação dos números adimensionais

Para esta etapa foi estudada a dinâmica de bolhas de ar ascendendo em soluções aquosas de açúcar, tal como foi reportado no trabalho experimental de Bhaga e Weber (1981), com o propósito de comparar os resultados da literatura com os obtidos pela metodologia VOF, utilizando o *solver interDyMFoam*. Os resultados também foram comparados com a simulação

feita por Hua e Lou (2007), onde implementaram um algoritmo numérico baseado na metodologia *Front-Tracking*, também para estudar a forma e a velocidade terminal de bolhas. As propriedades físicas dos fluidos foram escolhidas de maneira similar aos trabalhos de Bhaga e Weber (1981) e são mostradas na Tab. 5.4.

Tabela 5.4: Propriedades dos fluidos para as segunda e terceira etapa da simulação.

$\rho_l \left(\frac{kg}{m^3} \right)$	$\rho_g \left(\frac{kg}{m^3} \right)$	$\mu_l \left(\frac{kg}{m \cdot s} \right)$	$\mu_g \left(\frac{kg}{m \cdot s} \right)$	$\sigma \left(\frac{N}{m} \right)$
1314 - 1390	1,185	0,08 – 2,8	$1,766 \times 10^{-5}$	0,0769 – 0,08

O tamanho do domínio computacional foi mantido de acordo com o mostrado na Fig. 5.9 e o diâmetro da bolha foi calculado de maneira a obter os números de Eötvös e Morton desejados. Nesta etapa inicial foram estudados seis casos aqui denominados A, B, C, D, E e F.

Na Figura 5.14 são mostrados os resultados qualitativos da forma da bolha e os valores obtidos para os números de Reynolds calculados com a velocidade terminal da bolha. Nesta mesma figura são mostrados os resultados experimentais de Bhaga e Weber (1981) e os resultados numéricos de Hua e Lou (2007). O desvio absoluto, ε , os Re calculados no presente trabalho e os dados experimentais de Bhaga e Weber (1981) são apresentados também na Figura 5.14.

A forma da bolha concorda com a literatura na maioria dos casos analisados e somente os casos D e F apresentaram diferença com o reportado por Bhaga e Weber (1981). Esta diferença também mostrada nos resultados de Hua e Lou (2007) pode estar relacionada com a forma inicial da bolha no experimento, que é diferente dos testes numéricos. Os autores não explicam com clareza se a bolha era perfeitamente esférica no tempo inicial.

No caso A, a forma da bolha é denominada de calota elipsoidal. Nos casos B e C é chamada calota esférica e nos casos E e F é denominada “*Skirted*” pelo fato de possuir “saías” ou abas laterais. Os resultados obtidos no presente trabalho concordam com os apresentados no diagrama de Clift (Figura 2.2). Para os casos simulados com exceção do caso D o número de Reynolds está bem próximo do valor experimental reportado por Bhaga e Weber (1981). Nos casos B, C e D a diferença de valores do número de Reynolds é maior comparando a presente simulação com o trabalho experimental. Esta diferença é maior ainda no caso D provavelmente devido à condição instável da bolha.

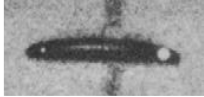


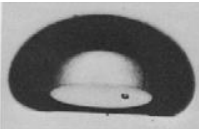


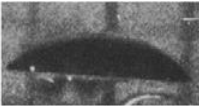


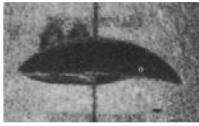







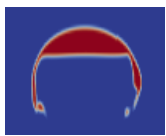
Caso	Resultados experimentais (Bhaga e Weber, 1981)		Resultados numéricos (Hua e Lou, 2007)		Presente estudo	
	Condição	Forma final da bolha	Condição	Forma final da bolha	Condição	Forma final da bolha
A	$Eo = 32,2$ $Mo = 8,2 \times 10^{-4}$ $Re = 55,3$		$Eo = 32,2$ $Re = 52,96$ $\epsilon = 4,23\%$		$Eo = 32,2$ $Mo = 8,2 \times 10^{-4}$ $Re = 54,54$ $\epsilon = 1,37\%$	
B	$Eo = 243$ $Mo = 266$ $Re = 7,77$		$Eo = 243$ $Re = 8,397$ $\epsilon = 7,5\%$		$Eo = 243$ $Mo = 266$ $Re = 6,77$ $\epsilon = 12,87\%$	
C	$Eo = 115$ $Mo = 4,63 \times 10^{-3}$ $Re = 94,0$		$Eo = 115$ $Re = 88,701$ $\epsilon = 5,64\%$		$Eo = 115$ $Mo = 4,63 \times 10^{-3}$ $Re = 87,53$ $\epsilon = 6,88\%$	
D	$Eo = 237$ $Mo = 8,2 \times 10^{-4}$ $Re = 259$		$Eo = 237$ $Re = \text{Instável}$		$Eo = 237$ $Mo = 8,2 \times 10^{-4}$ $Re = 372,86$ $\epsilon = 30,54\%$	
E	$Eo = 339$ $Mo = 43,1$ $Re = 18,3$		$Eo = 339$ $Re = 17,912$ $\epsilon = 2,12\%$		$Eo = 339$ $Mo = 43,1$ $Re = 18,29$ $\epsilon = 0,05\%$	
F	$Eo = 641$ $Mo = 43,1$ $Re = 30,3$		$Eo = 641$ $Re = 29,931$ $\epsilon = 1,22\%$		$Eo = 641$ $Mo = 43,1$ $Re = 30,36$ $\epsilon = 0,2\%$	

Figura 5.14: Comparação da forma final e números adimensionais reportados na literatura e na presente simulação.

No geral, os resultados obtidos com a metodologia VOF (*solver interDyMFoam*) para esta faixa de adimensionais, concordam tanto com os resultados experimentais quanto com os numéricos da literatura consultada. Isso mostra que apesar das desvantagens da metodologia VOF, no que diz respeito à definição correta da interface, o refinamento dinâmico contribui para a correta captura das mudanças topológicas.

5.1.4 Efeito das razões de massa específica e viscosidade

Nesta etapa foram realizadas simulações para analisar o efeito das razões de massa específica e viscosidade sobre a velocidade terminal e a forma da bolha. Para este estudo foi usado o caso E da Fig. 5.14, com $Eo = 339$ e $Mo = 43,1$. Foram mantidas as mesmas propriedades do líquido da Tab. 5.4 e as propriedades do gás foram calculadas para cada razão de massa específica e viscosidade analisadas.

As razões de massa específica simuladas foram de $\rho_l/\rho_g = 2, 10, 100$ e 1000 com razão de viscosidade constante ($\mu_l/\mu_g = 100$). Na Figura 5.15 são apresentados os resultados do número de Reynolds da bolha em função do tempo adimensional para as diferentes razões de massa específica.

Observa-se que a razão de massa específica tem pouco efeito sobre o valor do Reynolds terminal, mostrado na região ampliada da Fig. 5.15. O número de Reynolds obtido para as razões $\rho_l/\rho_g = 2$ e 1000 diferem aproximadamente de 1%. A forma final da bolha nos quatro casos também foi muito próxima. A deformação da bolha depende principalmente do valor da tensão superficial, que permaneceu inalterada para todos os casos. No entanto, a pequena influência da razão de massa específica tem efeito sobre a posição final da bolha. Maiores razões de massa específica produzem menores valores de Reynolds terminal. De outro modo, o efeito da razão de massa específica é um pouco mais significativo sobre a velocidade terminal do que sobre a forma da bolha, como também comentado no trabalho de George, Rezende e Maliska (2008).

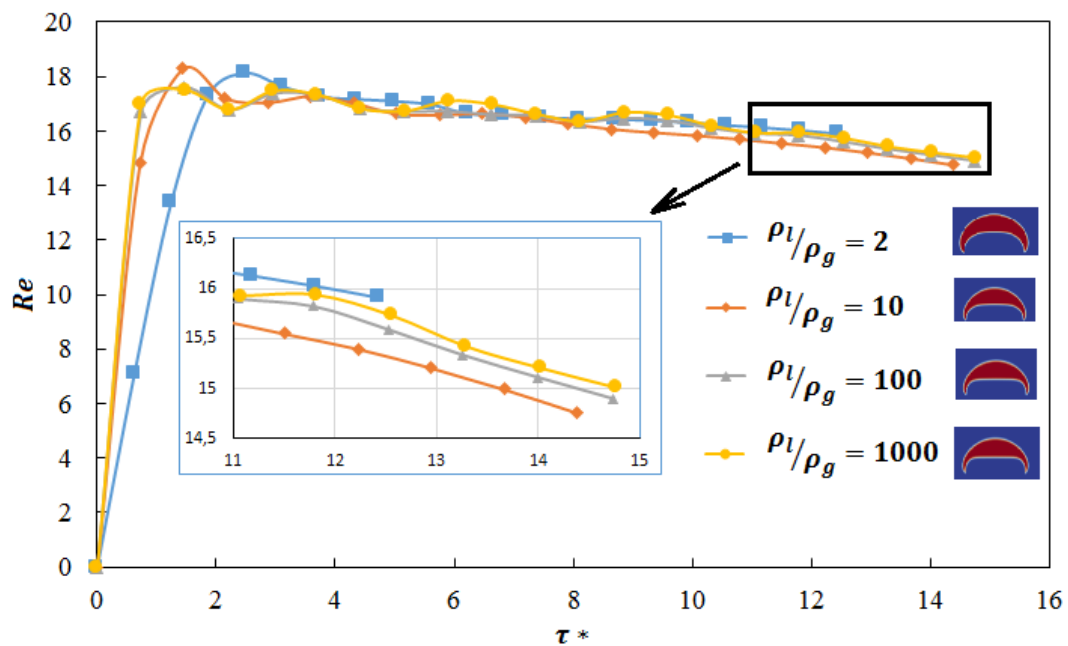


Figura 5.15: Número de Reynolds da bolha em função do tempo adimensional para $Eo=339$, $Mo=43,1$ e $\mu_l/\mu_g = 100$.

Na Figura 5.16 observa-se o efeito da razão de viscosidade sobre o número de Reynolds da bolha, para os quatro casos simulados ($\mu_l/\mu_g = 5, 10, 100$ e 500), com uma razão de massa específica constante ($\rho_l/\rho_g = 1000$).

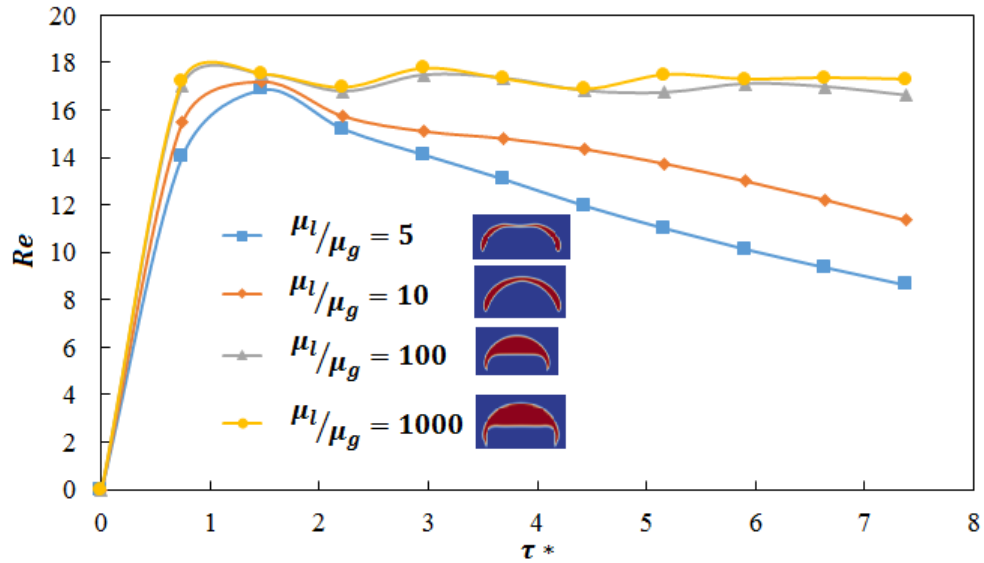


Figura 5.16: Número de Reynolds da bolha em função do tempo adimensional para $Eo=339$, $Mo=43,1$ e $\rho_l/\rho_g = 1000$.

À medida que a razão de viscosidade diminui, a bolha torna-se mais lenta e a tensão de cisalhamento na bolha é mais significativa. Para uma menor razão de viscosidade, maior é a recirculação formada acima da bolha provocando uma maior deformação em seu centro, o que atrasa a sua ascensão. Isto pode ser visto na Figura 5.17, por meio dos campos dos vetores velocidade.

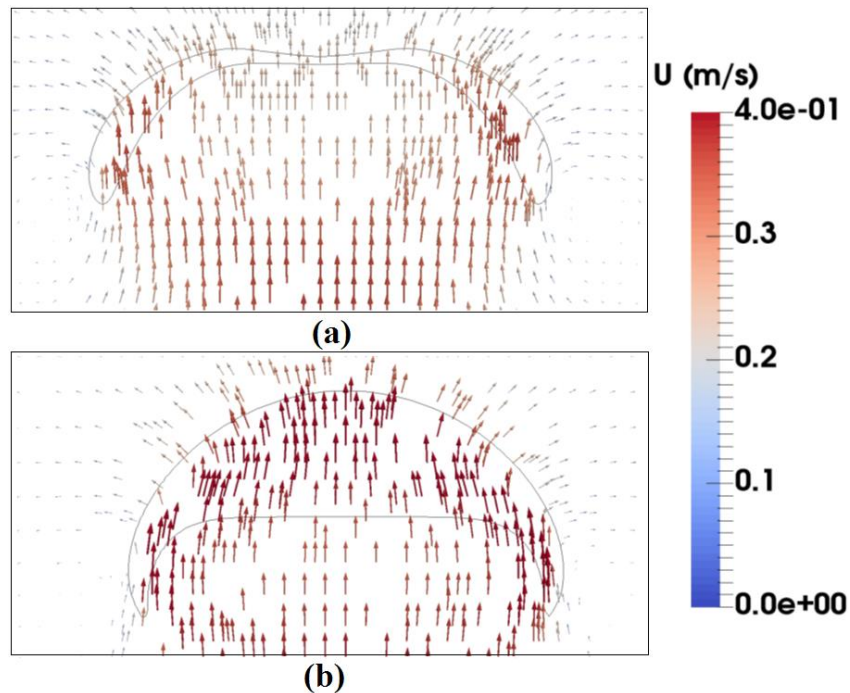


Figura 5.17: Campo de velocidades sobre a bolha para $Eo=339$, $Mo=43,1$, $\rho_l/\rho_g = 100$ com (a) $\mu_l/\mu_g = 5$ e (b) $\mu_l/\mu_g = 100$.

Para todos os casos simulados da ascensão da bolha a metodologia VOF, com refinamento dinâmico (*solver interDyMFoam*) mostrou-se adequada.

A seguir são apresentados os resultados das simulações de uma gota em um microcanal com junção em T . Como foi explicado no capítulo 2, estas aplicações estão presentes em processos envolvendo reação química, mistura, emulsão, extração e síntese de medicamentos e materiais, assim como na micro-electrônica, engenharia genética, bioengenharia, miniaturização de dispositivos e em outros campos como na indústria aeroespacial e nuclear. As simulações são feitas neste campo de pesquisa para que se possa obter um maior entendimento da dinâmica da gota através do microcanal. Objetiva-se assim poder controlar o processo de transporte e ruptura da gota necessários na produção contínua de gotículas ou bolhas altamente monodispersas.

5.2 Transporte de uma gota em um microcanal

Os primeiros estudos com manipulação de gotas em microcanais, foram realizados em resposta a duas idéias. Em primeiro foi utilizada esta técnica para produzir gotas de forma controlada e reprodutível, para possível aplicação na ciência dos materiais (UMBANHOWAR, PRASAD e WEITZ, 2000, e UTADA *et al.* 2005). A segunda idéia foi desenvolver o chamado laboratório miniaturizado em um único *chip* microprocessador. Neste *chip*, microcanais em pequenos volumes podem ser usados para redes de controle de muitas reações químicas (SONG, TICE e ISMAGILOV, 2003).

5.2.1 Definição da geometria e malha.

Nesta etapa foram feitas simulações de uma gota em um microcanal de seção retangular com bifurcação em T (*T-junction*), com o objetivo de estudar o mecanismo de ruptura da gota através da junção. Esta geometria em T foi escolhida pelo fato de ser amplamente utilizada nos processos práticos. Além disso, na literatura é possível encontrar um maior número de resultados com esta configuração.

Inicialmente a gota têm uma forma retangular e as propriedades físicas utilizadas na simulação foram definidas de acordo com a literatura consultada, com o intuito de validar os resultados obtidos. Na Figura 5.18 são apresentadas as dimensões e a geometria do microcanal e da gota. As dimensões do microcanal ($w \times w$) foram as mesmas em todas as simulações ($400 \times 400 \mu\text{m}$) e o comprimento da gota no momento que ela atinge a bifurcação foi estabelecido entre $1,5 < l_o/w < 4,5$.

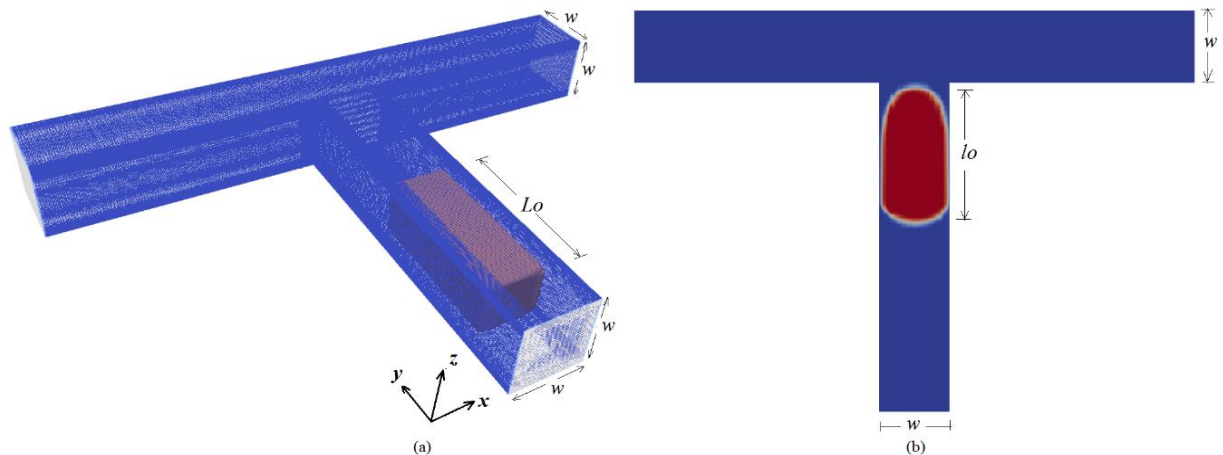


Figura 5.18: Dimensões principais do domínio computacional e da gota: (a) vista tridimensional no instante inicial e (b) vista 2D no momento que a gota chega na junção.

O comprimento da gota l_0 é medido no momento em que ela chega na junção em T , o qual é determinado na etapa de pós-processamento. Inicialmente a gota tem um comprimento um pouco maior (L_0) definido na etapa de pré-processamento de acordo com o valor desejado. Todas as simulações foram feitas com uma gota inicial de seção transversal $300 \times 300 \mu\text{m}$ e a influência de outros valores sobre a condição de ruptura da gota não foi estudada neste trabalho.

O domínio computacional inicial foi gerado com células hexaédricas usando o utilitário *blockMeshDict*, gerando inicialmente 1.728.000 células computacionais. Foi necessário refinar a malha próximo das paredes do microcanal, com o intuito de modelar a fina camada de líquido existente entre a gota e a parede e, desta forma, simular a espessura desta película de lubrificação. Para isto a ferramenta *snappyHexMesh* foi utilizada, a qual permite fazer diferentes níveis de refinamento em qualquer zona do domínio computacional, tal como mostra a Figura 5.19. No Anexo D apresenta-se o arquivo *snappyHexMeshDict* onde são definidos os parâmetros para o refinamento do domínio computacional próximo das paredes e na região da junção. Observa-se que existe maior refinamento na junção e na direção x , com a intenção de capturar com maiores detalhes a dinâmica de ruptura da gota que ocorre nesta região. Com esta ferramenta foram adicionadas à malha base aproximadamente 2.000.000 de células computacionais.

Em todas as simulações é imposta uma velocidade inicial para a fase dispersa (gota) de $0,04 \text{ m/s}$ e para a fase contínua a velocidade na entrada está na faixa de $0,005$ a $0,035 \text{ m/s}$. Utilizou-se como base experimental os resultados apresentados por Fu *et al.* (2011). Os mecanismos de ruptura através da junção em T foram comparados com o trabalho teórico de Leshansky *et al.* (2012). Estes autores realizaram análises experimentais e teóricas, para

analisar a dinâmica e condição de ruptura de gotas e bolhas em microcanais também com junção em T .

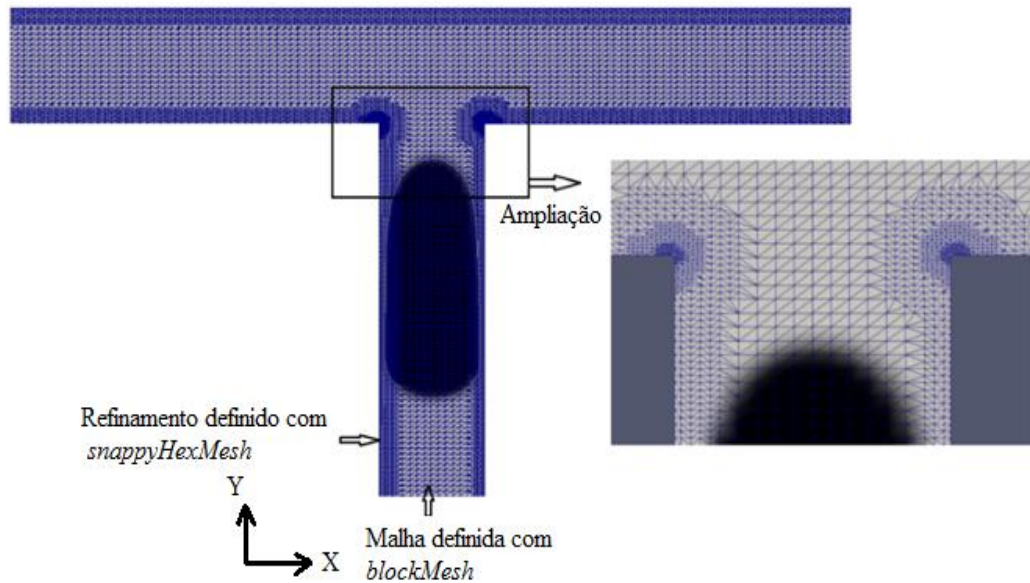


Figura 5.19: Domínio computacional gerado com o arquivo *blockMeshDict* e a ferramenta *snappyHexMesh*.

O número de capilaridade afeta a condição de ruptura e dependendo do valor da velocidade da fase dispersa ao chegar na bifurcação foram obtidos resultados com números de capilaridade na faixa de $0,01 \leq Ca \leq 0,09$. Estes valores permitem comparar com a literatura consultada. O número de capilaridade definido na Eq. (3.27), foi calculado com base na velocidade da fase dispersa no momento que a gota chega na junção em T (Fig. 5.18b), a qual foi determinada com auxílio do programa de pós-processamento *ParaView*. As propriedades dos fluidos foram definidas de acordo como o trabalho de Hoang *et al.* (2013) e Jullien *et al.* (2009), e são apresentadas na Tab. 5.5.

Tabela 5.5: Propriedades físicas dos fluidos.

Fluido	ρ (kg/m^3)	μ ($\frac{kg}{m \cdot s}$)	σ (N/m)
Fase contínua	770	8×10^{-3}	0,005
Fase dispersa (gota)	1000	1×10^{-3}	-

O tempo de processamento depende da velocidade inicial da fase contínua, ou seja, do número de capilaridade obtido, como mostrado na Tab. 5.6.

Tabela 5.6: Tempo total gasto nas simulações feitas no *Cluster* para os diferentes números de Capilaridade.

Ca	0,09	0,08	0,05	0,03	0,02	0,01
t (h)	9	10	18	28	36	64

5.2.2 Estudos dos parâmetros da simulação.

Foram simulados três (3) casos com $Ca = 0,09$ e $l_0/w = 2,04$ e valores do fator de compressão (Eq. 3.11) $C_\alpha = 1, 2$ e 3 . Encontrou-se maior distorção da interface para $C_\alpha = 3$ como apresentado na região do retângulo da Figura 5.20a. Os vetores velocidade sobre esta região da interface para os três valores de C_α estudados são apresentados nas Figuras ampliadas 5.20b, 5.20c e 5.20d. Observa-se mudanças abruptas no campo de velocidade que afetam a forma da interface para $C_\alpha = 2$ e $C_\alpha = 3$.

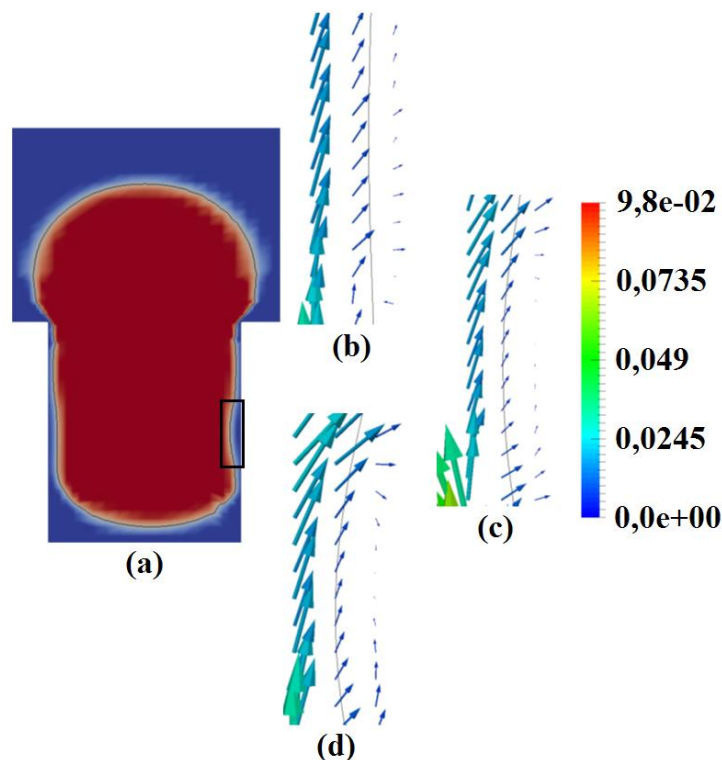


Figura 5.20: (a) Forma da interface no instante de tempo $t = 0,02$ s com um valor do fator de compressão de $C_\alpha = 3$. Ampliação da região do retângulo para (b) $C_\alpha = 1$, (c) $C_\alpha = 2$ e (d) $C_\alpha = 3$.

Desta forma, pode-se concluir que utilizando valores elevados deste parâmetro podem aparecer distorções não físicas na interface. Na Equação 3.11, onde C_α é utilizado, a velocidade de compressão é levada em consideração apenas na região da interface e é calculada na direção normal, para evitar qualquer dispersão. O Fator de compressão C_α não afetou o mecanismo de ruptura nos testes realizados. Portanto o valor unitário foi utilizado nos demais estudos.

Uma das grandes dificuldades na simulação de gotas em microcanais é o fato de que com baixos números de capilaridade a interface tende a se deformar de forma não física devido a presença de correntes parasitárias. As correntes parasitárias são movimentos não físicos do

fluido, geradas ao usar modelos, tais como as técnicas CSF (*Continuum Surface Force*), para aproximar o cálculo da tensão superficial na interface (RENARDY e RENARDY, 2002). Sua magnitude geralmente aumenta com o aumento da força de tensão superficial e pode se tornar tão elevada que afeta a predição do campo de velocidade ou em circunstâncias extremas pode causar completa ruptura não física da interface. Estas correntes espúrias são um grande problema no caso de microfluidos devido ao fato de que elas aumentam com o inverso do número de capilaridade mas podem ser reduzidas em magnitude com maior refinamento da malha e com a diminuição do passo de tempo. Neste trabalho as correntes parasitárias foram controladas tanto pelo refinamento da malha quanto pela definição de um número de Courant máximo em cada simulação. Na Figura 5.21 são apresentados os vetores velocidade para um caso de uma mesma gota com $Ca = 0,04$ e $l_o/w = 1,87$, durante o processo de ruptura em dois intervalos de tempo, $t = 0,095$ s e $t = 0,09625$ s e números de Courant de 0,2 e 0,5.

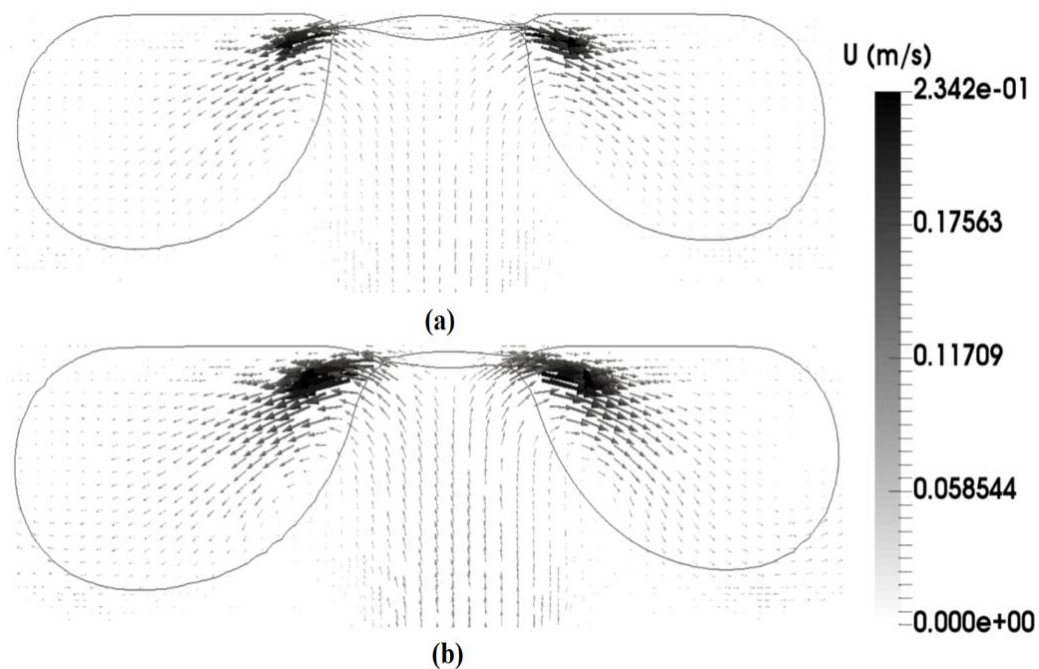


Figura 5.21: Vetores velocidades sobre a gota para um caso com $Ca = 0,04$, $l_o/w = 1,87$ para (a) $Co = 0,2$ e (b) $Co = 0,5$.

O instante em que a ruptura ocorre é diferente e no caso de $Co = 0,5$, a ruptura ocorre de forma assimétrica. No caso de um número Courant de 0,2 a ruptura ocorre a 0,095 s, enquanto que para $Co = 0,5$ (Fig. 5.23), a ruptura ocorreu em $t=0,09625$ s. É possível observar nessa figura que no momento da ruptura existe maior intensidade dos vetores velocidade no caso (b). Esse aumento pode ter sido produzido pela presença de correntes parasitárias em torno

da região de ruptura e desta forma, o valor de 0,2 para o número de Courant foi utilizado para todas as outras simulações.

A metodologia S-CLSVOF implementada no OpenFOAM foi comparada com a metodologia VOF, já implementada no código. Na Figura 5.22 é apresentada a comparação entre as duas metodologias durante a ruptura da gota para um caso com $Ca = 0,04$ e $lo/w = 1,87$.

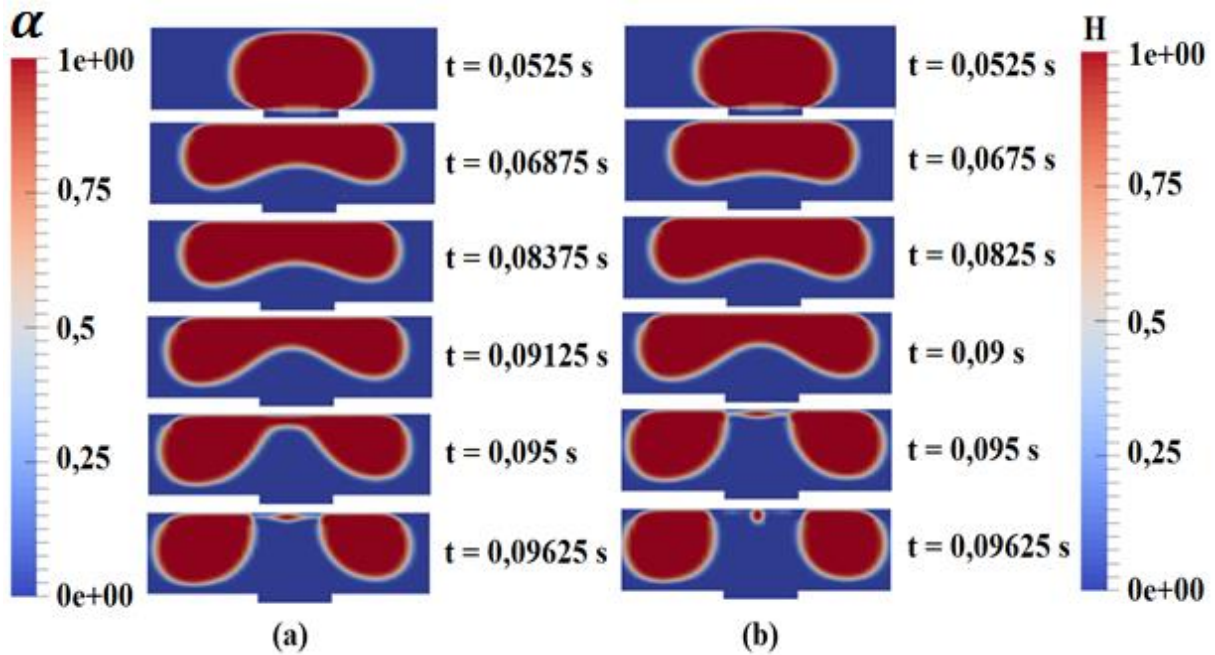


Figura 5.22: Evolução temporal da gota para um caso com $Ca = 0,04$, $lo/w = 1,87$ e $Co = 0,2$ com as metodologias (a) VOF e (b) S-CLSVOF.

Observa-se na Fig. 5.22 a tendência de ocorrer ruptura não simétrica com a metodologia VOF. Com o método S-CLSVOF essa tendência é reduzida. Com o método VOF, observa-se que a gota formada no lado direito é menor do que a do lado esquerdo. Esta assimetria, cuja causa ainda não é muito bem identificada, pode ter sido originada por correntes espúrias que foram reduzidas com o método S-CLSVOF. Ambas simulações foram realizadas com o mesmo tamanho de malha definido na seção 5.2.1. Para identificar melhor a causa da assimetria no escoamento deve-se realizar testes com bolhas estacionárias e medir as correntes parasitárias. Estes testes não foram realizados, mas como a metodologia acoplada S-CLSVOF apresentou melhores resultados, pode-se concluir que estas correntes espúrias foram reduzidas já que as mesmas condições de simulação foram usadas tanto com VOF quanto com o acoplamento.

As Figuras 5.23 e 5.24 mostram os campos de velocidade na junção do microcanal obtidos com ambas as metodologias no momento em que ocorre a ruptura. Observa-se uma diminuição da intensidade do campo de velocidade no interior da gota no momento da ruptura

com o método S-CLSVOF. O acoplamento melhora o cálculo e a definição da espessura da interface e como consequência obtém-se uma melhor estimativa da curvatura $k(\varphi)$, contribuindo para um corpo de velocidade mais simétrico.

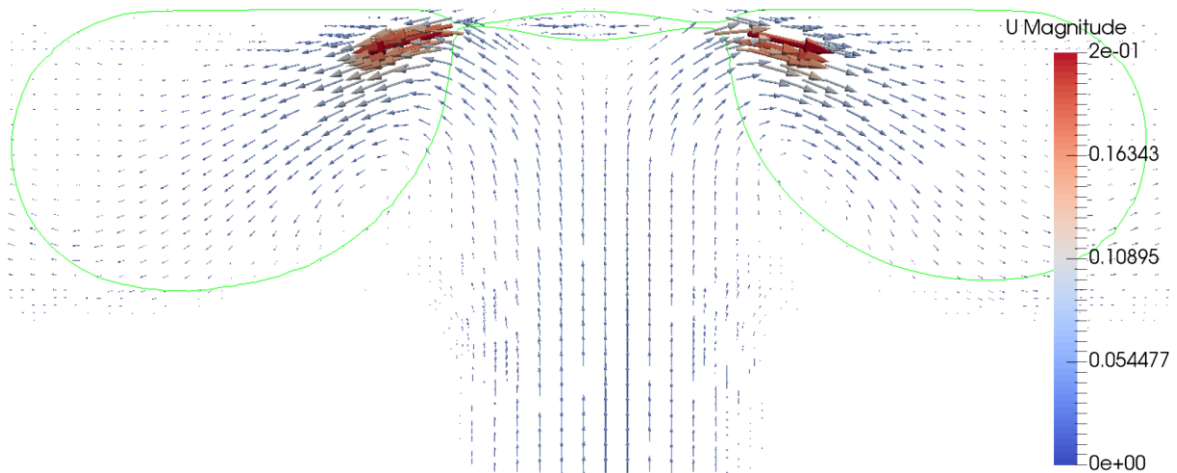


Figura 5.23: Campo dos vetores velocidade no momento da ruptura da gota com a metodologia S-CLSVOF.

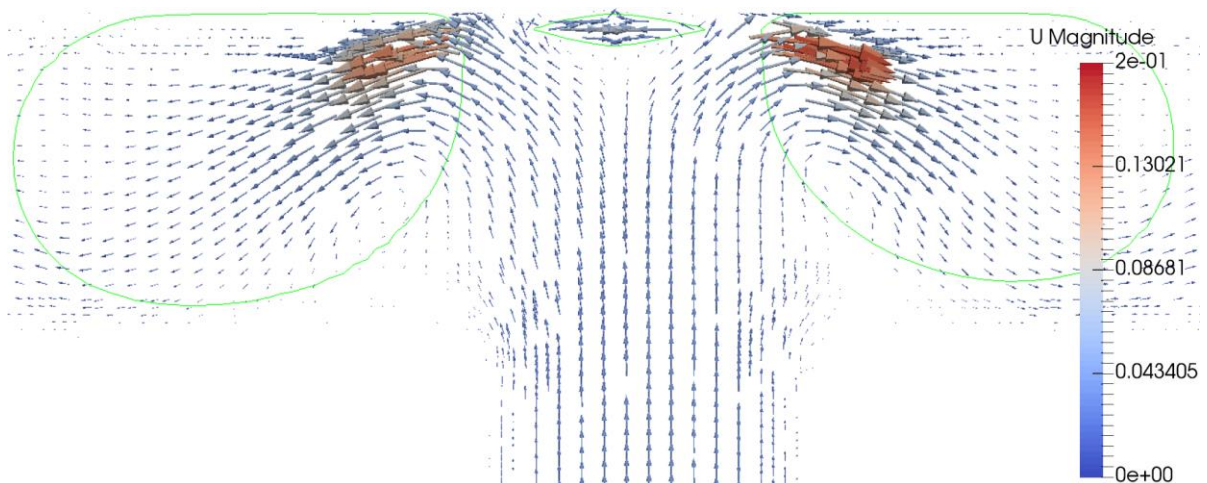


Figura 5.24: Campo dos vetores velocidade no momento da ruptura da gota com a metodologia VOF.

Neste trabalho, as correntes parasitárias foram diminuídas também através do controle do número de Courant buscando-se obter a forma da interface mais precisa quanto possível, principalmente no momento da ruptura. As correntes parasitárias são limitadas tanto pelo termo de inércia quanto pelo termo viscoso das equações de Navier-Stokes, o que torna o controle ou a eliminação uma tarefa difícil.

Na seção a seguir, as simulações foram realizadas com o método acoplado S-CLSVOF para estudar os mecanismos de ruptura de uma gota através da junção em T.

5.2.3 Mecanismos de ruptura/não ruptura de gotas com junções em T.

A condição de ruptura de gotas e bolhas escoando através de junções em T é fortemente influenciada pelo número de capilaridade (Ca) e pela relação l_0/w no momento em que ela chega na junção. Muitos estudos numéricos e experimentais reportam diagramas de ruptura da interface de bolhas/gotas expressado em termos destes dois parâmetros. Nesta etapa foi testada a metodologia S-CLSVOF implementada para prever a condição de ruptura ou não da gota através da junção.

Na Figura 5.25 são apresentados os resultados do presente trabalho em conjunto com as linhas de transição obtidas por Leshansky *et al.* 2012, (referenciado no capítulo 2).

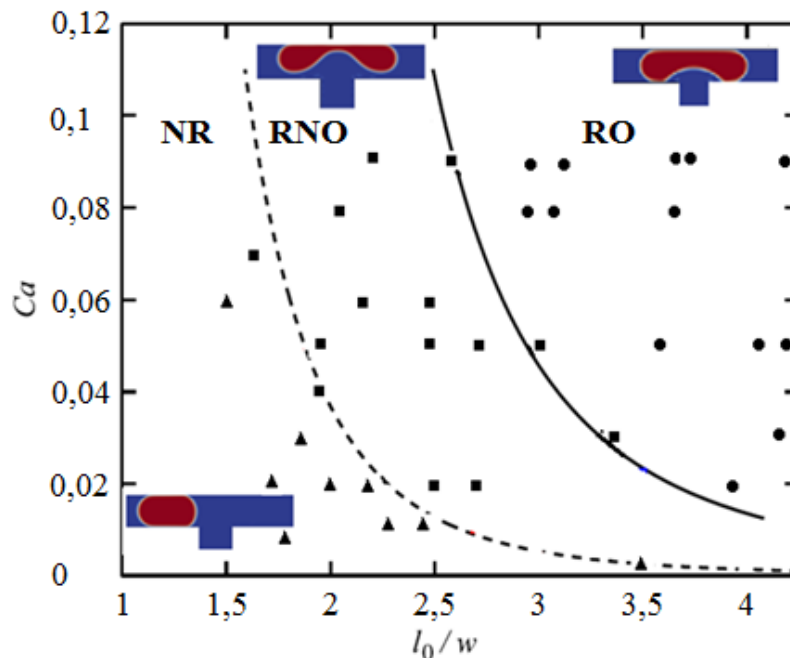


Figura 5.25: Regimes de ruptura de gotas em microcanais com junção em T. A curva tracejada é a transição R/NR e a curva contínua é a transição RO/RNO (Leshansky *et al.* 2012). Os símbolos correspondem aos resultados numéricos do presente trabalho: ▲ NR, ■ RNO e ● RO.

Neste gráfico são apresentadas duas curvas que separam a transição entre os regimes de ruptura através de junções em T . A curva tracejada é a transição entre os mecanismos de ruptura/não-ruptura (R/NR). A ruptura da gota pode se apresentar de duas maneiras, obstruída ou bloqueada (RO) e não obstruída (RNO). A curva contínua é a transição entre estes dois regimes RO e RNO. No caso da ruptura com obstrução existe uma distância muito pequena entre a parede e a gota durante a ruptura, normalmente invisível a olho nu, e em geral acontece com gotas que tem comprimento maior que no caso do regime não obstruído ($l_0/w > 3$). No caso de RNO esta distância entre a parede e a gota é visível. Hoang *et al.* (2013) apresentaram no seu trabalho uma figura similar (Anexo E) com apenas dois regimes de ruptura, NR e RNO.

Os resultados das simulações apresentam boa concordância com as regiões delimitadas por Leshaansky *et al.* 2012.

Na Figura 5.26 apresenta-se a distribuição do campo de velocidade ao redor da gota na junção em T , no regime de NR para $Ca = 0,01$ e $l_o/w = 2,3$ em comparação com um caso similar do trabalho de Fu *et al.* (2011).

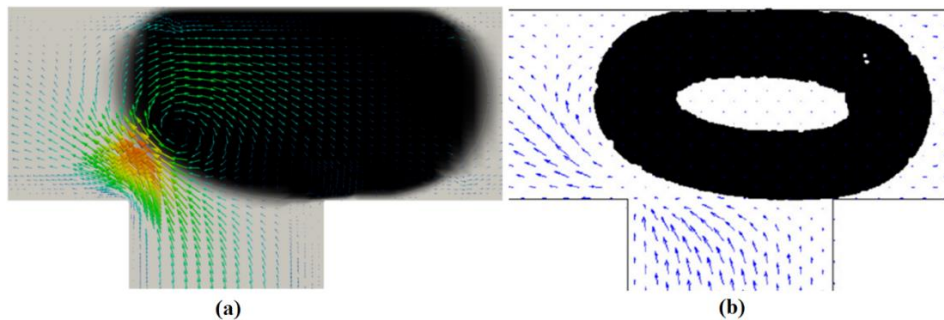


Figura 5.26: Vetores velocidade instantâneos para $Ca = 0,01$ e $l_o/w = 2,3$: (a) presente estudo e (b) Fu *et al.* (2011).

A gota entra na junção em T mantendo o líquido parcialmente obstruído do lado direito. Neste instante o líquido só pode passar pelo pequeno espaço entre a gota e a parede, então, abre-se um túnel de um lado da junção e eventualmente a gota se move na direção oposta. Parte do líquido que passa por esta abertura retorna e empurra a gota para a direita, no instante seguinte nota-se que no ponto onde a gota é pressionada cria-se uma recirculação do líquido.

Vários estudos computacionais e experimentais relatam o mecanismo de rompimento de gotas ou bolhas em função da menor espessura adimensional da gota (δ/w) (Jullien *et al.* 2009, Fu, *et al.* 2011e Leshaansky *et al.* 2012). Na Figura 5.27 é ilustrada esta espessura na região da junção, também denominada no inglês de *neck thickness* ou espessura do “pescoço”.

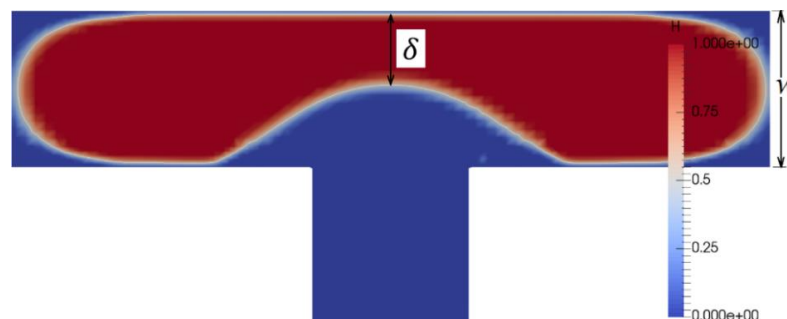


Figura 5.27: Espessura da gota (δ) durante a ruptura no regime RNO.

Na Figura 5.28 é apresentada uma comparação de uma simulação feita neste trabalho para o caso de $Ca = 0,02$ e $l_o/w = 2,5$, com um caso similar do trabalho experimental de Fu *et al.* (2011).

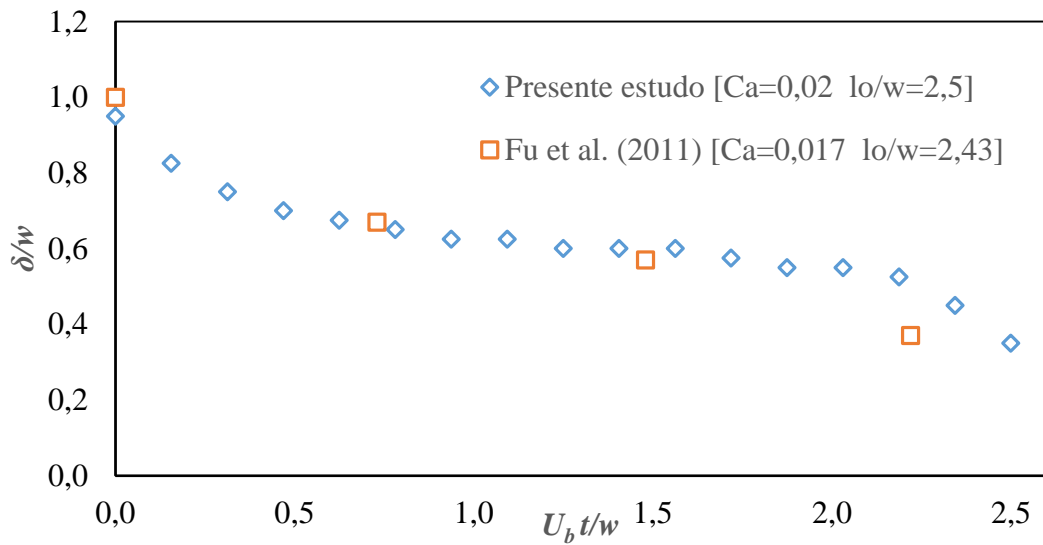


Figura 5.28: Evolução temporal da espessura da gota durante sua ruptura em comparação com um caso similar do trabalho de Fu *et al.* (2011).

O processo de diminuição do “pescoço” da gota é acompanhado pelo seu alongamento de forma linear com o tempo. Como comentado por Fu, *et al.* (2011) e observado na Fig. 5.28 a diminuição de δ/w é realizada pelo fato da pressão do líquido sobre a gota também diminui com o tempo.

A diminuição da espessura adimensional da gota (δ/w) com o tempo para um número de capilaridade $Ca = 0,09$ é mostrada nas Figuras. 5.29 e 5.30, para diferentes comprimentos de gotas, respectivamente, para os regimes RO e RNO.

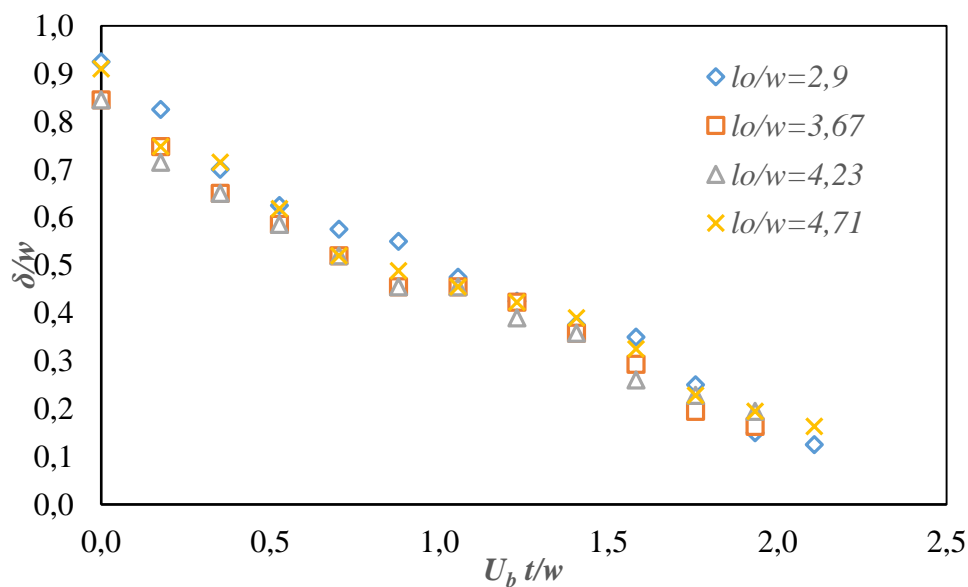


Figura 5.29: Evolução temporal da espessura da gota δ/w , para $Ca = 0,09$ no regime RO para diferentes relações de l_0/w .

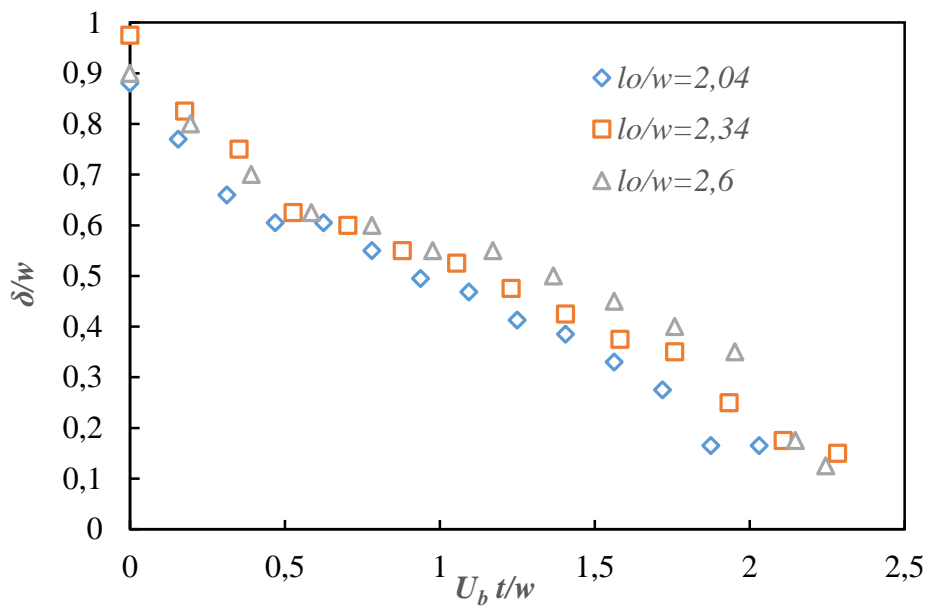


Figura 5.30: Evolução temporal da espessura da gota δ/w , para $Ca = 0,09$ no regime RNO para diferentes relações de l_o/w .

No caso RO (Fig. 5.29) o comportamento da espessura adimensional da gota é semelhante para os diferentes valores de l_o/w . O comportamento do escoamento também é aproximadamente o mesmo para os tamanhos da gota simulados, para o regime de RNO (Fig. 5.30). A espessura crítica da gota (espessura no momento da ruptura), em ambos os regimes de separação variou entre 0,15 e 0,2.

Todos os testes realizados para a faixa de números de capilaridade $0,01 \leq Ca \leq 0,09$ e faixas de relações do comprimento da gota e do microcanal de $2,2 \leq l_o/w \leq 4,71$ foram agrupados com o intuito de gerar um modelo matemático para representar o comportamento da gota durante o processo de ruptura. Com ajuda do programa LAB FIT conseguiu-se obter uma correlação matemática que relaciona a espessura adimensional da gota (δ/w) com o tempo adimensional ($\tau^* = U_b t/w$), com o número de capilaridade e com a geometria de gota. O LAB FIT é um programa desenvolvido para o tratamento de dados experimentais, ajuste de curvas, gráficos 2D e 3D, estatísticas básicas, propagação de erros e outras ferramentas matemáticas (SILVA *et al.* 2004).

Esta correlação possui um modelo similar ao proposto por Leshaansky *et al.* 2012 utilizando seus resultados numéricos. Depois de testar vários modelos, encontrou-se, o que melhor ajustou os dois regimes de ruptura (RO e RNO), dado por

$$\frac{\delta}{w} = X1 - X2 * C_a^{X3} * X4 * \left(\frac{l_o}{w}\right)^{X5} * X6 * \tau^{*G} \quad (5.1)$$

onde $X1$, $X2$, $X3$, $X4$, $X5$ e $X6$ são as constantes desta equação. A Equação 5.2 representa o modelo matemático para os regime RO e RNO obtidos neste trabalho.

$$\frac{\delta}{w} = 0,8062 - 0,6447C_a^{0,3222} \left(\frac{l_o}{w}\right)^{0,1489} \tau^{*0,804} \quad (5.2)$$

A correlação apresentada por Leshansky *et al.* 2012 é dada pela expressão

$$\frac{\delta}{w} = 1 - 0,582 \left(\frac{\alpha'}{3C_a}\right)^{-\frac{1}{7}} \tau^{*\frac{3}{7}} \quad (5.3)$$

onde α' é uma constante relacionada com a geometria da gota e τ^* o tempo adimensional (Ut/w).

Foram traçadas curvas da evolução temporal da espessura adimensional da gota usando as Eqs. (5.2) e (5.3) com $C_a = 0,09$ and $l_o/w = 2,9$ (Figura 5.31).

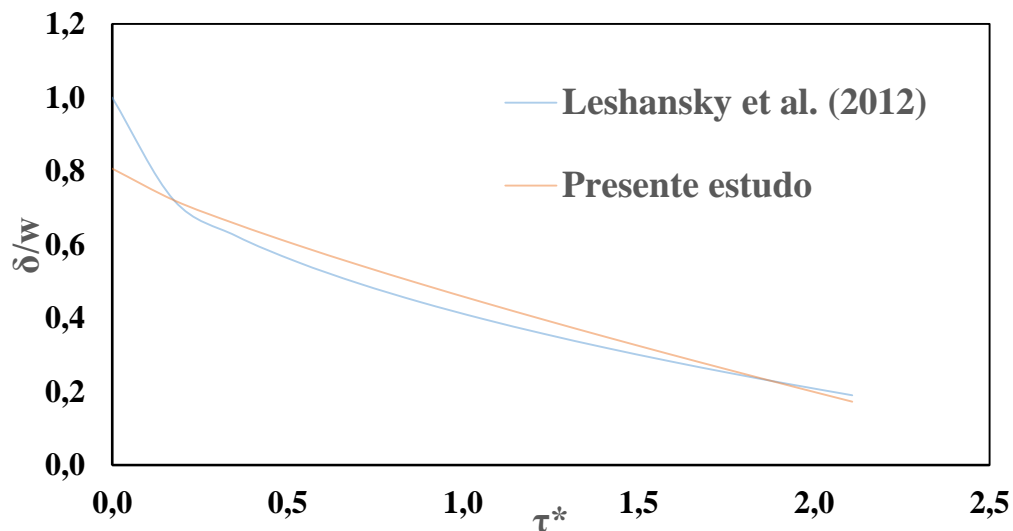


Figura 5.31: Modelo matemático obtido neste trabalho e modelo proposto por Leshansky *et al.* (2012) com $C_a = 0,09$ e $l_o/w = 2,9$ e $\alpha' = 0,25$.

Observa-se pela Figura 5.31 que a correlação obtida com os resultados numéricos do presente trabalho, Eq. (5.2), pode ser usada para prever a evolução da espessura da gota com

precisão satisfatória, no intervalo de números de capilaridade estudados. No início do intervalo de tempo a diferença com a correlação de Leshansky *et al.* 2012 é maior e em torno de 19%.

Outros testes com diferentes números de capilaridade devem ser feitos para verificar a validade da correlação proposta fora da faixa testada. Porém, quanto menor o valor de Ca maior o tempo de simulação tanto devido à necessidade de uma malha mais refinada quanto devido ao tempo gasto para a gota atingir a bifurcação.

Capítulo 6

CONCLUSÕES E TRABALHOS FUTUROS

6.1 Conclusões

Um estudo numérico da dinâmica de bolhas ascendendo em um fluido estacionário e da dinâmica de uma gota em microcanais com junção em T foi apresentado neste trabalho utilizando a metodologia Volume de Fluido (VOF) e o acoplamento Volume de Fluido-*Level Set* (S-CLSVOF), por meio do programa OpenFOAM.

As simulações da dinâmica de bolhas ascendendo foram realizadas com o objetivo de estudar a influência dos parâmetros adimensionais Reynolds, Eötvös e Morton na velocidade terminal e na forma final da bolha. Também foram analisados os efeitos do tamanho do domínio computacional e das condições de contorno de livre deslizamento e de não deslizamento. Nestas simulações foi usada a metodologia VOF com os *solvers interFoam* e *interDyMFoam* e o acoplamento S-CLSVOF com o *solver interFoam*. Os resultados mostraram que um domínio computacional com dimensões laterais de sete vezes o diâmetro da bolha não afeta a velocidade terminal. Enquanto a razão das massas específicas mostrou ter pouca influência sobre a dinâmica da bolha e sobre sua forma final, a razão de viscosidade tem influência significativa sobre o Reynolds terminal e sobre a sua forma final. Em todos os casos estudados no presente trabalho os resultados mostram boa concordância com os resultados da literatura.

A dinâmica de uma gota movendo-se em um microcanal com junção em T foi também estudada numericamente também com as duas metodologias de captura de interface VOF e S-CLSVOF. Neste caso objetivou-se estudar os mecanismos de ruptura da gota através da junção e sua dependência com o número de capilaridade e com variáveis geométricas, tais como a relação entre o comprimento da gota e a largura do microcanal (l_o/w). Nestas simulações foi preciso fazer refinamento do domínio computacional próximo a parede do microcanal e na região da junção utilizando a ferramenta *snappyHexMesh*, com o objetivo de modelar a fina camada de líquido que existe entre a gota e a parede.

Comprovou-se que para os números de capilaridade estudados neste trabalho ($0,01 \leq Ca \leq 0,09$) o surgimento de correntes parasitárias tende a mudar o mecanismo de ruptura da gota assim como deforma-la de forma não física. Após alguns testes foi escolhido um número de Courant $C_0 = 0,2$ para diminuir a magnitude dos erros numéricos.

Todos os resultados estão em boa concordância com a literatura consultada. O OpenFOAM mostrou-se adequado para simular escoamento com gotas/bolhas nos regimes estudados. O algoritmo S-CLSVOF que foi proposto por Albadawi *et al.* 2013 para modelar bolhas ascendendo em meios líquidos, mostrou-se adequado também para modelar escoamentos em microcanais com junções em T , uma vez que permitiu prever com precisão o regime de ruptura através de uma melhor definição da interface.

6.2 Trabalhos Futuros.

Tendo em vista os resultados obtidos, algumas sugestões para continuação do presente trabalho são propostas. Em relação aos avanços numéricos associados ao uso da técnica S-CLSVOF, sugere-se que novos estudos enfoquem:

- Implementação de malha dinâmica na metodologia S-CLSVOF, sendo o próximo passo na solução de problemas tanto na ascensão de bolhas em meios líquidos quanto na dinâmica de gotas em microfluidos.
- No caso do uso da metodologia S-CLSVOF para simulação em microfluidos, é preciso melhorar o cálculo da força de tensão superficial, pois as correntes parasitárias são mais significativas principalmente com baixos números de capilaridade como os estudados neste trabalho.

Visando a simulação de escoamentos bifásicos em microcanais, um estudo mais aprofundado de alguns fenômenos deve ser realizado, por exemplo:

- Modelar os mecanismos de ruptura no caso de bolhas/gotas através de microcanais com junções em T com diferentes geometrias e seções transversais do canal.
- Estudar a influência das razões de massa específica e viscosidade entre as fases dispersas e contínuas também sobre os regimes de ruptura.
- Aprofundar o estudo da influência da pressão da fase contínua sobre a diminuição da espessura da gota durante a ruptura.

Referências Bibliográficas

- AFKHAMI, S.; LESHANSKY, A. M. E RENARDY, Y. (2011), “Numerical investigation of elongated drops in a microfluidic T-junction”, *Physics of Fluids*, Vol. 23, 022002, pp.1-14.
- ALBADAWI, A.; DONOGHUE, D. B.; ROBINSON, A. J.; MURRAY, D. B.; DELAURÉ, Y. M. C. (2013), “Influence of surface tension implementation in Volume of Fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment”, *International Journal of Multiphase Flow*, Vol 53, pp. 11-28.
- BANAIBEDALRUHMAN, A. (2015). *Dynamic Meshing around fluid-fluid interfaces with applications to droplet tracking in contraction geometries*, Dissertations, Master’s – Michigan Technological University, Michigan, USA, 191 p.
- BERBEROVIĆ, E., VAN HINSBERG, N. P., JAKIRLIĆ, S., ROISMAN, I. V. e TROPEA, C. (2009), Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, *Physical Review*, E79, 036306.
- BERGLES, A. E.; COLLIER, J. G.; DELHAYE, J. M.; HEWITT, G. F. e MAYINGER, F. (1981). *Two-phase Flow and Heat Transfer in the Power and Process Industries*. Washington, Hemisphere Publ. Corp.
- BHAGA, D.; WEBER, M. E. (1981). “Bubbles in viscous liquids: shapes, wakes and velocities”, *J. Fluid Mech*, Vol.105, pp.61-85.
- BRACKBILL, J. U.; KOTHE, D. B.; ZEMACH, C. (1992). “A continuum method for modeling surface tension”. *Journal of Computational Physics*, Vol.100, pp.335-354.
- BRENNEN, C. E. (2005). *Fundamentals of Multiphase Flows*. Cambridge, Cambridge University Press, 407 p.
- CERNE, G., PETELIN, S. e TISELJ, I. (2001). Coupling of the interface tracking and the two-fluid models for the simulation of incompressible two-phase flow, *Journal of computational physics*, 171, p. 776-804.
- CHENADEC, V. L.; PITSCH, H. (2013), “A 3D Unsplit forward/backward Volume-of-Fluid approach coupling to the Level Set methods”, *Journal of Computational Physics*, Vol.233, pp.10-33.
- CHORIM, A. (1968), “Numerical solution of the Navier-Stokes equations”, *Match. Comp.*, Vol.22, pp.745-762.
- CLIFT, R.; GRACE, J. R. (1978), *Bubbles, Drops and Particles*. New York, Academic Press, 380 p.
- DEBAR, R. (1974), “Fundamentals of the KRAKEN code”, *Lawrence Livermore Laboratory*, UCID-17366, pp 1-17.
- DELLA, R. G.; BLANQUART, G. (2014), “Level set reinitialization at a contact line”, *Journal of Computational Physics*, Vol.265, pp.34-49.
- DIAZ, M. E.; IRANZO, A.; CUADRA, D.; BARBERO, R.; MONTES, F. J. e GALÁN, M. A. (2008), “Numerical simulation of the gas-liquid flow in a laboratory scale bubble column Influence of bubble size distribution and non-drag forces”, *Chemical Engineering Journal*, Vol 1390, pp 363-379.

- DIETEL, H. M.; DIETEL, P. J. (2006).** *C++ Como Programar*, 5ª Edição. São Paulo, SP. Brasil Pearson. Prentice Hall.
- ESMAEELI, A.; TRYGGVASON, G. (1998)**, “Direct numerical simulations of bubbly flows. Part1: Low Reynolds number arrays”, *J. Fluid Mech.* Vol.377, pp. 313-345.
- ESTACIO, K. C. (2008)**, *Simulação de escoamentos de fluidos em superfícies definidas por pontos não organizados*. Tese de Doutorado – Universidade de São Paulo, USP, São Carlos, SP, Brasil. 163 p.
- FACCINI, J. L. H. (2008)**, *Estudo de escoamentos bifásicos estratificados usando técnicas ultra-sônicas*. Tese de Doutorado - Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear, RJ, Brasil, 177 p.
- FAVERO, J. L. (2009)**, *Simulação de escoamentos viscoelásticos: desenvolvimento de uma metodologia de análise utilizando o software OpenFOAM e equações constitutivas diferenciais*. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. Porto Alegre, RS. 127 p.
- FORTUNA, A. O. (2000)**, *Técnicas Computacionais para Dinâmica dos Fluidos*, São Paulo, Brasil, Editora da Universidade de São Paulo, 435 p.
- FU, T. Ma. Y.; FUNFSCHILLING, D. e LI, H. Z. (2011)**, “Dynamics of bubble breakup in a microfluidic T-junction divergence”, *Chemical Engineering Science*, Vol. 66, pp. 4184-4195.
- FUKANO, T. e KARIYASAKI, A. (1993)**, “Characteristics of gas-liquid two-phase flow in capillary tube”, *Nuclear Engineering and Design*, Vol. 141, pp. 59-68.
- GEORGE, I.; REZENDE, R.; MALISKA, C. (2008)**. “Estudo numérico do escoamento ascendente de uma bolha de gás em um meio líquido”, *1º Encontro Brasileiro sobre Ebulição, Condensação e Escoamento Multifásico Líquido-Gás*, Florianópolis, Brasil.
- GUEYFFIER, D.; LI, J.; NADIM, A.; SCARDOVELLI, R. E.; ZALESKI, S. (1999)**, “Volume of fluid Interface Tracking with smoothed surface stress methods for tree-dimensional flows”, *Journal of Computational Physics*, Vol.152, pp. 423-456.
- GUO, M. T.; ROTEM, A.; HEYMAN, J. A.; WEITZ, D. A. (2012)**, “Droplet microfluidics for high-throughput biological assays”, *Lab on a Chip*, Vol 12, pp. 2146-2155.
- HEWITT, G. F. (1982)**, *Handbook of Multiphase Systems*, Hemisphere Publishing Corporation, New York, 479 p.
- HETSRONI, G.; MOSYAK, A.; SEGAL, Z.; POGREBNYAK, E. (2003)**. “Two-phase flow patterns in parallel micro-channels”, *International Journal of Multiphase Flow*, Vol.29, pp. 341-360.
- HIRT, C. W.; NICHOLS, B. D. (1981)**, “Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries”, *Journal of Computational Physics*. Vol.39, pp.201-225.
- HOANG, D. A. (2012)**. *Breakup of confined droplets in microfluidos*, Dissertations, Master’s – van Gyeongsang National University, Zuid-Korea, 109p.
- HOANG, D. A.; van STEIJN, V.; PORTELA, L. M.; KREUTZER, M. T. e KLEIJN, C. R. (2013)**, “Benchmark numerical simulations of segmented two-phase

flows in microchannels using the Volume of Fluid method”, *Computers and Fluids*, Vol. 86, pp.28-36.

HUA, J.; LUO, J. (2007), “Numerical simulation of bubble rising in viscous liquid”, *Journal of computational physics*, Vol. 222, pp. 769-795.

ISSA, R. I. (1986), “Solution of the implicitly discretized fluid flow equations by operator-splitting”, *Journal of Computational Physics*. Vol.62, pp.40-65.

JACQMIN, D. (1996). “An energy approach to the continuum surface tension method”, *American Institute of Aeronautics and Astronautics, Inc*, Paper 96-0858.

JULLIEN, M. C., CHING, M. J. T. M., COHEN, C., MENETRIER, L., TABELING, P. (2009), “Droplet breakup in microfluidic T-junction at small capillary numbers”, *Physics of Fluids*, Vol. 21, 072001 pp. 1-6.

KWAKKEL, M.; BREUGEM, W. P.; BOERSMA, B. J. (2013), “Extension of a CLSVOF method for droplet-laden flows with a coalescence/breakup model” *Journal of Computational Physics*, Vol.253, pp. 166-188.

LAFURIE, B.; NARDONE, C.; ZALESKI, S.; ZANETTI, G. (1994), “Modelling merging and fragmentation in multiphase flows with SURFER”, *Journal of Computational Physics*, Vol.113, pp. 134-147.

LESHANSKY, A. M.; AFKHAMI, S.; JULLIEN, M. C. e TABELING, P. (2012), “Obstructed breakup of slender drops in a microfluidic T junction”, *Physical Review Letters*, PRL 108, 264502, pp 1-5.

LINK, D. R.; ANNA, S. L.; WEITZ, D. A. e STONE, H. A. (2004), “Geometrically mediated breakup of drops in microfluidic device”, *Phys. Rev. Lett.* Vol. 92, N° 5, 054303, pp 1-4

LOPES, R. S. L. (2008), *Desenvolvimento de metodologias para simulação de escoamentos polidispersos usando código livre*. Tese de Doutorado. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 244 p.

LUO, X. Y.; NI, M. J.; YING, A.; ABDU, M. (2006), “Application of the level set method for multi-phase flow computation in fusion engineering” *Fusion Engineering and Design*, Vol.81, pp.1521-1526.

MENEZES, De M. T. (2013), *Implementação de modelos de transporte de massa multicomponente no OpenFOAM*, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 68 p.

NAPOLITANO, L. M. (2004), *Implementação numérica do método Level Set para propagação de curvas e superfícies*. Dissertação de Mestrado. Universidade de São Paulo, USP, 74 p.

NOH, W. F.; WOODWARD, P. (1976), “SLIC (Simple Line Interface Calculations)”, *Lecture Notes in Physics*, Vol.59, pp. 330-340.

NOURGALIEV, R. R.; WIRI, S.; DINH, N.T.; THEOFANOUS, T. G. (2005), “On improving mass conservation of Level Set by reducing spatial discretization errors” *International Journal of Multiphase Flow*, Vol.31, pp.1329-1336.

OpenCFD LTD, OpenFOAM, The Open Source CFD Toolbox, User Guide, 2009.

OpenCFD LTD, OpenFOAM, The Open Source CFD Toolbox, Programmer’s Guide, 2009.

- OSHER, S.; FEDKIW, R. (2001)**, “Level Set methods: An overview and some recent results”, *Journal of Computational Physics*, Vol.169, pp. 463-502.
- OSHER, S. SETHIAN, J. A. (1988)**, “Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations”, *Journal of Computational Physics*. Vol.79, pp.12-49.
- PARK, H. S. e PUNCH, J. (2008)**. “Friction factor and heat transfer in multiple microchannels with uniform flow distribution”, *International Journal of Heat and Mass Transfer*, Vol. 51, pp. 4535-4543.
- PATANKAR, S. V.; SPALDING, D. B. (1972)**, “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows” *Int. J. Heat Mass Transfer*, Vol.15, pp. 1787-1806.
- RAYMOND F.; ROSANT, J. M. (2000)**, “A numerical and experimental study of the terminal velocity and shape of bubbles in viscous liquids”, *Chemical Engineering Science*, Vol. 55, pp. 943-955.
- RENARDY, Y.; RENARDY, M. (2002)**, “PROST: A parabolic reconstruction of surface tension for the volume-of-fluid method”, *Journal of Computational Physics*, Vol.183, pp. 400-421.
- RIDER, W.; KOTHE, D. B. (1998)**, “Reconstructing Volume Tracking”, *Journal of Computational Physics*, Vol.141, pp.112-152.
- RUSCHE, H. (2002)**, *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. Tese de Doutorado. University of London and Imperial College, London, 343 p.
- SARKAR, S. (2008)**, *3-D Multiphase flow modeling of spray cooling using parallel computing*, Tese de Doutorado. University of Arkansas, Fayetteville, 164 p.
- SCARDOVELLI, R.; ZALESKI S. (1999)**, “Direct numerical simulation of free-surface and interfacial flow”, *Annu. Rev. Fluid Mech.* Vol.31 pp. 567–603.
- SETHIAN, J. A.; SMEREKA, P. (2003)**, “Level Set methods for fluid interfaces”, *Annu. Rev. Fluid Mech*, Vol.35, pp. 341-372.
- SILVA, W. P., SILVA, C. M. D. P. S., CAVALCANTI, C. G. B., SILVA, D. D. P. S., SOARES, I. B., OLIVEIRA, J. A. S. AND SILVA, C. D. P. S. (2004)**, LAB Fit Ajuste de Curvas: Um software em português para tratamento de dados experimentais, *Revista Brasileira de Ensino de Física*, v. 26. n. 4, p. 419-427
- SONG, H.; TICE, J. D. e ISMAGILOV, R. F. (2003)**, “A microfluidic system for controlling reaction networks in time”, *Angew. Chem., Int. Ed.*, 42(7), pp. 768-772.
- SUSSMAN, M. (2003)**, “A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles”, *Journal of Computational Physics*, Vol.187, pp. 110-136.
- SUSSMAN, M.; ALMGREN, A. S.; BELL, J. B.; COLELLA, P.; HOWELL, L. H.; WELCOME, M. L. (1999)**, “An adaptive Level Set approach for incompressible two-phase flows”, *Journal of Computational Physics*, Vol.148, pp. 81-124.
- SUSSMAN, M. PUCKETT, E. G. (2000)**, “A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows”, *Journal of Computational Physics*, Vol.162, pp. 301-337.

- SUSSMAN, M.; SMEREKA, P.; OSHER, S. (1994)**, “A level Set approach for computing solutions to incompressible two-phase flow”, *Journal of Computational Physics*, Vol.114, pp. 146-159.
- TOMIYAMA, A.; CELATA, G. P.; HOSOKAWA, S.; YOSHIDA, S. (2002)**, “Terminal velocity of single bubbles in surface tension force dominant regime”, *International Journal of Multiphase Flow*, Vol. 28, pp. 1497-1519.
- TRIPLETT, K. A.; GHIAASIAAN, S. M.; ABDEL-KHALIK, S. I. e SADOWSKI, D. L. (1999a)**, “Gas-liquid two-phase flow in microchannels. Part I: two-phase flow patterns”, *International Journal of Multiphase Flow*, Vol.25, pp. 377-394.
- TRIPLETT, K. A.; GHIAASIAAN, S. M.; ABDEL-KHALIK, S. I.; LEMOUEL, A.; MCCORD, B. N. (1999b)**, “Gas-liquid two-phase flow in microchannels. Part II: void fraction and pressure drop”, *International Journal of Multiphase Flow*, Vol.25, pp. 395-410.
- TRYGGVASON, G.; SCARDOVELLI, R. e ZALESKI, S. (2011)**, *Direct numerical simulations of gas-liquid multiphase flows*. Cambridge, Cambridge university press, 322 p.
- UMBANHOWAR, P. B.; PRASAD, V. e WEITZ, D. A. (2000)**, “Monodisperse emulsion generation via drip break off in a coflowing stream”. *Langmuir*, 16, pp 347-351.
- UTADA, A. S.; LORENCEAU, E.; LINK, D. R.; KAPLAN, P. D.; STONE, H. A. e WEITZ, D. A. (2005)**. “Monodisperse double emulsions generated from a microcapillary device”. *Science*, Vol 308, pp 537-541.
- VERSTEEG, H. K.; MALALASEKERA, W. (1995)**. *An introduction to Computational Fluid Dynamics*. New York, USA, John Wiley & Sons Inc. 255 p.
- WANG, Z.; YANG, J.; STERN, F. (2012)**. “A new volume-of-fluid method with a constructed distance function on general structured grids”, *Journal of Computational Physics*, Vol.231, pp. 3703-3722.
- WELLER, H. G.; TABOR, G.; JASAK, H.; FUREBY, C. (1998)**, “A tensorial approach to computational continuum mechanics using object-oriented techniques”, *Journal Computers in Physics*, Vol.12, Issue 6. Pp. 620-631.
- WHITESIDES, G. M. (2006)**. “The origins and future of microfluidic”, *Nature*, Vol 447, n.27, pp. 368-373.
- YANG, X.; JAMES, A. J.; LOWENGRUB, J. ZHENG, X.; CRISTINI, V. (2006)**, “An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids”, *Journal of Computational Physics*, Vol.217, pp. 364-394.
- YEOH, G. H.; TU, J. (2010)**, *Computational Techniques for Multi-Phase Flows*. Burlington, USA, Butterworth-Heinemann, 643 p.
- YOUNGS, D. L. (1982)**, “Time-dependent multi-material flow with large fluid distortion”, *Numerical methods for fluid dynamics*, London: Academic Press, p.273-285.
- ZALESKI, S.; LI, J.; SUCCI, S.; SCARDOVELLI, R.; ZANETTI, G. (1995)**, “Direct numerical simulation of flows with interfaces” *Proceedings of the 2nd international conference on multiphase flow*, Kyoto, Apr. 3–7, Vol. 2, pp. PT2-1–PT2-12.

Apêndices.

Apêndice 1

1.1 Arquivos que compõem cada pasta de um caso em OpenFOAM

Pasta 0: Contém os arquivos individuais de dados para campos particulares. Os dados podem ser tanto condições iniciais e condições de contorno que o usuário especifica para definir o problema como resultados escritos por OpenFOAM.

Pasta constant: Contém as informações da malha (subpasta *polyMesh*) e arquivos que especificam as propriedades físicas dos materiais/fluidos envolvidos na simulação. Por exemplo, no caso de se utilizar refinamento dinâmico tem-se o arquivo *dynamicMeshDict* no qual define-se os parâmetros relacionados com o refinamento dinâmico, o arquivo *transportProperties* contém as propriedades dos fluidos envolvidos no problema e os arquivos *RASProperties* e *turbulenceProperties* no qual estão os dados do modelo de turbulência utilizado no problema em estudo. Como já foi dito nos capítulos anteriores neste trabalho foram feitas simulações de bolhas ascendendo em meios líquidos e gotas escoando em microcanais, portanto os escoamentos são tratados no regime laminar. A subpasta *polyMesh* contém o arquivo *blockMeshDict* com informações topológicas e geométricas da malha. Estes dados podem ser atualizados durante a simulação, tendo o usuário liberdade para apagar informações sobre os volumes de controle (volume, área das faces, posição do centro do volume/face, e outros.) e recalculá-las quando for necessário. Pode-se alterar, por exemplo, o refinamento da malha (mudança topológica) ou melhorar a representação da geometria (malhas poliédricas móveis).

Pasta system: Os arquivos contidos neste diretório estão associados com o procedimento de solução do caso. Pelo menos 3 arquivos devem estar contidos neste diretório: *controlDict*, onde define-se os parâmetros de controle da simulação, como o tempo de início e término da simulação, passo de tempo, controle de escrita de dados, e outros, *fvSolution*, que seleciona os métodos para resolver o sistema de equações lineares e suas tolerâncias, assim como outros parâmetros de controle do algoritmo de solução, *fvSchemes*, seleciona as aproximações de discretização por volumes finitos usados na solução do problema. Pode também conter outros arquivos como: *setFieldsDict* onde são definidas as informações relacionadas com as fases presentes na simulação (para o caso de escoamentos bifásicos), *decomposeParDict* onde se fixam os parâmetros relacionados com a decomposição do domínio para o processamento em

paralelo, isto é, a geometria e campos associados são divididos para processadores separados para a solução.

Apêndice 2

2.1 Acesso ao código fonte

No OpenFOAM os solvers estão separados do código fonte e são chamados de *application*. Nesta pasta há subdiretórios para cada solver. Na Figura A1 apresenta-se o caminho de acesso ao código fonte no caso da versão 2.4 de OpenFOAM.

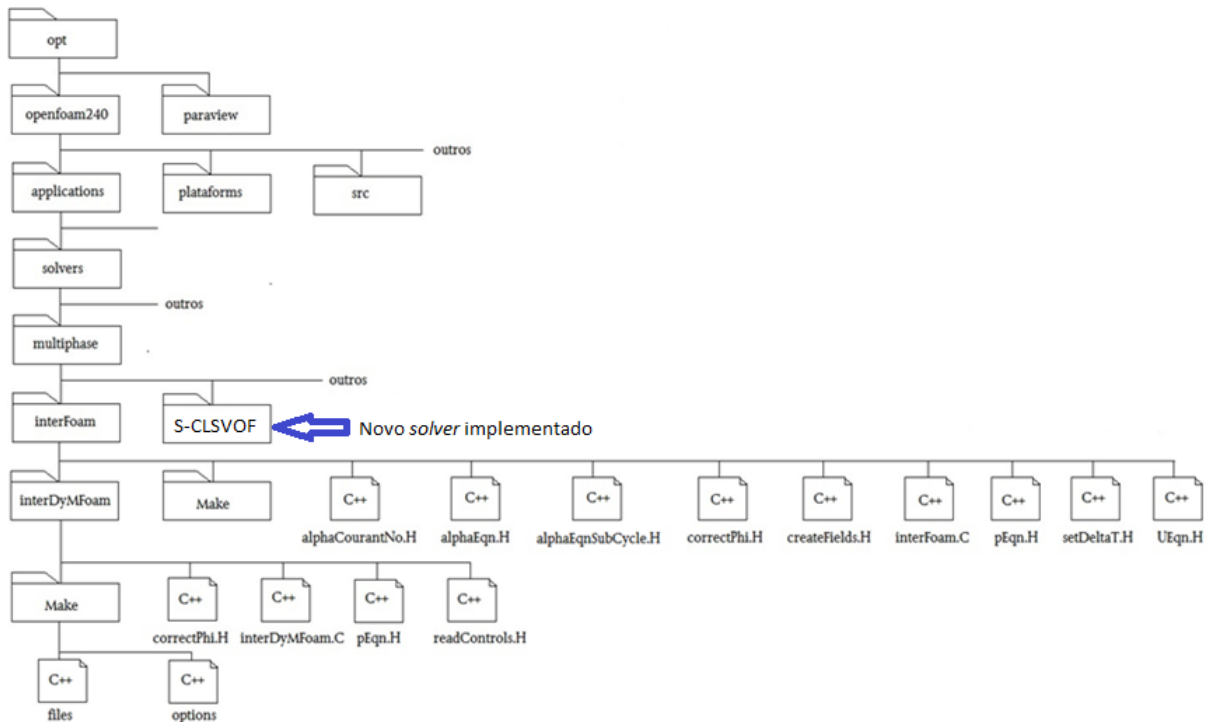


Figura A1: Algumas pastas e arquivos importantes do código fonte no OpenFOAM.

Na pasta **opt** encontra-se o código fonte do OpenFOAM e ParaView. Na pasta **OpenFoam240** estão as pastas *application*, *platforms*, *src* entre outras. Se quisermos modificar, por exemplo, o *solver* *interFoam* do módulo *multiphase* seguimos o caminho: *application/solvers/multiphase* obtendo nesta última pasta os solvers mostrados na Tabela 4.2.

Dado que o *solver* a modificar é o *interFoam*, então, é recomendável criar uma cópia dele com outro nome por exemplo *S-CLSVOF*. Estes *solvers* não podem ser modificados sem antes acessar a uma janela de linux e solicitar o privilégio de “super usuário”, por meio do comando *sudo nautilus*. Permitida esta condição pode-se salvar as modificações feitas nos arquivos. Antes de compilar o novo *solver* é preciso fazer algumas alterações no arquivo *files* da pasta *Make* para que tudo possa ser compilado no novo *solver* criado sem substituir o *solver* inicial. Em primeiro lugar, muda-se o nome do arquivo para seu novo nome e apaga-se o arquivo de

dependências (.dep). Este arquivo de dependências é criado durante o processo de compilação. O arquivo *file* no novo solver *my_interFoam* fica da seguinte forma:

```
S-CLSVOF.C
EXE = $(FOAM_USER_APPBIN)/S-CLSVOF
```

Não são necessárias alterações para o arquivo *options*. Depois destas modificações procede-se ao processo de compilação digitando no terminal o seguinte comando:

```
wmake
```

Se tudo funcionar corretamente, um novo *solver* de binário deve aparecer no diretório `FOAM_USER_APPBIN`. Também pode-se salvar este *solver* binário no seguinte caminho de acesso, `opt/openfoam240/plataforms/linux64GccDPOpt/bin`, desta forma os casos com o novo *solver* podem ser executados desde o mesmo tutorial de OpenFOAM. Os arquivos do código fonte do *solver interFoam* pode-se olhar no Anexo A. Antes de fazer modificações neste código é preciso entender as funcionalidades de seus arquivos, como são:

- **alphaCourantNo.H**: calcula a média do número de Courant tendo presente o valor máximo definido no arquivo *controlDict* (Fig. 4.2).
- **alphaEqn.H**: resolve a equação da advecção ou transporte a partir da equação de continuidade da fase dispersa, com a finalidade de prever a distribuição da fase dispersa na mistura.
- **alphaEqnSubCycle.H**: resolve a equação da advecção ou transporte e obtém uma nova distribuição.
- **correctPhi.H**: neste arquivo são resolvidas funções de correção de fluxo para garantir a conservação da massa.
- **createFields.H**: na qual são definidos todos os campos escalar e vetoriais do problema em estudo.
- **pEqn.H**: após configurar a equação de velocidade, o *solver* vai para o passo de solução de pressão neste arquivo. Primeiramente a velocidade é predita para depois ajustar a equação de pressão.
- **setDeltaT.H**: reajusta o passo de tempo para manter um número de Courant máximo constante.

- **Ueqn.H:** resolve a equação de quantidade de movimento. Usa o algoritmo PIMPLE para o acoplamento pressão-velocidade.
- **InterFoam.C:** é o *solver* propriamente dito. Nele são chamadas as bibliotecas a serem usadas no *solver* e o procedimento para os cálculos nos arquivos anteriores.

Apêndice 3

3.1 Refinamento dinâmico com o Solver *interDyMFoam* e arquivos do código fonte relacionados com sua funcionalidade

O solver *interDyMFoam* pode fazer o refinamento apenas para as células hexaédricas 3D dividindo as células igualmente em todas as três direções. O refinamento da malha no *interDyMFoam* é conseguido usando as bibliotecas *dynamicFvMesh* e *dynamicMesh* e é inicializado chamando a função *mesh.update()*. Esta função é definida na biblioteca *dynamicFvMesh* e pode ser encontrado no arquivo */opt/openFoam240/src/dynamicFvMesh/dynamicFvMesh.C* no código fonte do OpenFOAM (vide Fig. 4.3). Esta função pode fazer o refinamento como segue (BANIABEDALRUHMAN, 2015):

- a) Lê o arquivo *dynamicMeshDict* localizado na pasta *constant* do diretório de caso (vide Fig. 4.2);
- b) Determina as células candidatas a serem refinadas chamando a função *selecRefineCandidates()* localizado no arquivo *dynamicRefineFvMesh.C*;
- c) Seleciona um subconjunto de células candidatas para refinamento chamando a função *selecRefineCells()* localizada no arquivo *dynamicRefineFvMesh.C*. O subconjunto é escolhido com base no número máximo de células permitido, definido no arquivo *dynamicMeshDict*.
- d) Executa o refinamento chamando a função *refine()* localizado no arquivo *dynamicFvMesh.C*. A função *refine()* chama a função *setRefinement()* definida na biblioteca *dynamicMesh* e pode ser encontrada no arquivo */opt/openFoam240/src/dynamicMesh/polyTopoChange/polyTopoChange/hexRef8.C*, no código fonte do OpenFOAM. Esta função faz o refinamento adicionando pontos no centro de cada célula, de cada face e de cada aresta. A partir destes pontos pode ser construído no interior de cada célula oito sub-células.
- e) Determina os pontos que podem ser não refinados chamando a função *selectUnrefinePoints()* localizado no mesmo arquivo *dynamicRefineFvMesh.C*. Estes pontos são escolhidos com base dos números *PointLevel* e *nBufferLayers* definidos no arquivo *dynamicMeshDict* (vide Fig. 4.2).
- f) Finalmente os campos são projetados da malha antiga para a nova.

Cada célula pode ser refinada até o número de refinamento máximo definido no arquivo *dynamicMeshDict* por meio do número *maxCell* e se as células totais excederem o número máximo de células, o refinamento pára. No Anexo B é apresentado o arquivo *dynamicMeshDict* no caso de simulação de uma gota em um micronal. O número *unrefineLevel* controla os pontos que não podem ser refinados.

Apêndice 4

4.1 Explicação das funcionalidades dos arquivos do código fonte do *solver interFoam*

alphaCourantNo.H: calcula o número de Courant máximo

alphaEqn.H: descreve a solução da equação de conservação de massa. Neste arquivo a equação convectiva de difusão é derivada da equação de continuidade da fase dispersa.

alphaEqnSubCycle.H: resolve a equação de transporte alpha

correctPhi.H: neste arquivo são calculadas funções de correção de fluxo para garantir a continuidade.

createFields.H: na qual são definidos todos os campos escalar e vetoriais do problema em estudo.

pEqn.H: na qual é resolvida a equação da pressão.

setDeltaT.H: arquivo onde é reajustado o passo do tempo para manter um número de Courant constante e evitar oscilações instáveis.

Ueqn.H: arquivo no qual é resolvida a equação da quantidade de movimento. Aplicação do algoritmo PIMPLE e acoplamento pressão-velocidade.

InterFoam.C: é o *solver*. Neste arquivo é controlada a compilação do *solver*. Nele são definidos todas as bibliotecas e utilitários usados para a compilação.

4.2 Arquivos adicionais no caso do *solver S-CLSVOF*.

CalcNewCurvature.H: Calcula a normal a interface e a curvatura da função *Level-Set*.

solveLSFunction.H: Reinicializa a função *Level-Set* e ϕ . Calcula a função Delta de Dirac e a Função de Heaviside (H).

uodateFlux.H: Atualiza o fluxo convectivo nas equações de Navier-Stokes.

mappingPsi.H: Inicializa as funções *Level-Set* e ϕ .

No arquivo *createFields.H* são definidas as novas variáveis e constantes (Tab. A4.1) do *solver S-CLSVOF*, além de corrigidos os valores da massa específica e viscosidade com LS (Eqs 3.22 e 3.23).

Tabela A4.1. Campos e constantes inicializadas para implementação do novo solver

Campos	Definição
ψ, ϕ	Função <i>Level Set</i> (LS)
ψ_0, ϕ_0	Função inicial LS
H	Função de Heaviside
Δ, δ	Função de Dirac
C	Curvature
Constantes	Definição
$\Delta x, \Delta x$	Tamanho da célula
γ, γ	Número adimensional de inicialização da função LS
ϵ, ϵ	Espessura da interface
$\Delta \tau, \Delta \tau$	Intervalo do tempo artificial
dimChange	Fração para a re-inicialização da função ϕ
σ, σ	Tensão superficial calculada com LS
ν_1, ν_1 e ν_2, ν_2	Correção da viscosidade com S-CLSVOF

ANEXOS

ANEXO A

A.1 Arquivo dynamicMeshDict

```

/*-----*- C++ -*-----
-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox
| \\      / O peration  | Version: 2.4.0
| \\      / A nd        | Web:      www.OpenFOAM.org
|  \\/      M anipulation |
|
\*-----*
-----*/
FoamFile
{
    version      2.4;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       dynamicMeshDict;
}
// * * * * *
* * * * //

dynamicFvMesh    dynamicRefineFvMesh;

dynamicRefineFvMeshCoeffs
{
    // How often to refine
    refineInterval 1;
    // Field to be refinement on
    field          alpha.phase1;
    // Refine field inbetween lower..upper
    lowerRefineLevel 0.001;
    upperRefineLevel 0.999;
    // If value < unrefineLevel unrefine
    unrefineLevel 10;
    // Have slower than 2:1 refinement
    nBufferLayers 1;
    // Refine cells only up to maxRefinement levels
    maxRefinement 2;
    // Stop refinement if maxCells reached
    maxCells      20000000;
    // Flux field and corresponding velocity field. Fluxes on
    changed
    // faces get recalculated by interpolating the velocity. Use
    'none'
}

```

```
// on surfaceScalarFields that do not need to be reinterpolated.
correctFluxes
(
    (phi none)
    (nHatf none)
    (rhoPhi none)
    (rho*phi none)
    (ghf none)

);
// Write the refinement level as a volScalarField
dumpLevel      true;
}

//
*****
***** //
```

ANEXO B

B.1 Fotografia do Cluster do Laboratório de Astrofísica Computacional da UNIFEI



ANEXO D

D.1 Arquivo snappyHexMesh

```

FoamFile
{
    version      2.4;
    format       ascii;
    class        dictionary;
    object       snappyHexMeshDict;
}
// * * * * *
* * * * * //

// Which of the steps to run
castellatedMesh true;
snap             true;
addLayers        false;

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
    T.stl
    {
        type      triSurfaceMesh;
        name      wing;
    }

    /* refinementBox
    {
        type      searchableBox;
        min       (-1.5 -1.5 -1);
        max       ( 2.5  1.5  1);
    }*/
};

// Settings for the castellatedMesh generation.
castellatedMeshControls
{
    // Refinement parameters
    // ~~~~~

    // If local number of cells is >= maxLocalCells on any processor
    // switches from from refinement followed by balancing
    // (current method) to (weighted) balancing before refinement.
    maxLocalCells 2000000;

```

```

    // Overall cell limit (approximately). Refinement will stop
    immediately
    // upon reaching this number so a refinement level might not
    complete.
    // Note that this is the number of cells before removing the
    part which
    // is not 'visible' from the keepPoint. The final number of
    cells might
    // actually be a lot less.
    maxGlobalCells 2000000;

    // The surface refinement loop might spend lots of iterations
    refining just
    // a few cells. This setting will cause refinement to stop if <=
    // minimumRefine are selected for refinement. Note: it will at
    least do one
    // iteration (unless the number of cells to refine is 0)
    minRefinementCells 200;

    // Number of buffer layers between different levels.
    // 1 means normal 2:1 refinement restriction, larger means
    slower
    // refinement.
    nCellsBetweenLevels 8;

    // Explicit feature edge refinement
    // ~~~~~

    // Specifies a level for any cell intersected by its edges.
    // This is a featureEdgeMesh, read from constant/triSurface for
    now.
    features (
        {
            file "T.eMesh";
            level 3;
        }
    );

    // Surface based refinement
    // ~~~~~

    // Specifies two levels for every surface. The first is the
    minimum level,
    // every cell intersecting a surface gets refined up to the
    minimum level.
    // The second level is the maximum level. Cells that 'see'
    multiple
    // intersections where the intersections make an
    // angle > resolveFeatureAngle get refined up to the maximum
    level.

```

```

refinementSurfaces
{
    wing
    {
        // Surface-wise min and max refinement level
        level (2 3);
    }
}

// Resolve sharp angles on fridges
resolveFeatureAngle 90;

// Region-wise refinement
// ~~~~~

// Specifies refinement level for cells in relation to a
surface. One of
// three modes
// - distance. 'levels' specifies per distance to the surface
the
// wanted refinement level. The distances need to be specified
in
// descending order.
// - inside. 'levels' is only one entry and only the level is
used. All
// cells inside the surface get refined up to the level. The
surface
// needs to be closed for this to be possible.
// - outside. Same but cells outside.

refinementRegions
{
    /* refinementBox
    {
        mode inside;
        levels ((1e15 2));
    }*/
}

// Mesh selection
// ~~~~~

// After refinement patches get added for all refinementSurfaces
and
// all cells intersecting the surfaces get put into these
patches. The
// section reachable from the locationInMesh is kept.
// NOTE: This point should never be on a face, always inside a
cell, even
// after refinement.
locationInMesh (0.000000112 0.001511 0.000011111);

```



```

    // Whether any faceZones (as specified in the
refinementSurfaces)
    // are only on the boundary of corresponding cellZones or also
allow
    // free-standing zone faces. Not used if there are no faceZones.
allowFreeStandingZoneFaces false;
}

```

```

// Settings for the snapping.
snapControls
{
    //- Number of patch smoothing iterations before finding
correspondence
    // to surface
nSmoothPatch 8;

    //- Relative distance for points to be attracted by surface
feature point
    // or edge. True distance is this factor times local
    // maximum edge length.
tolerance 1.0;

    //- Number of mesh displacement relaxation iterations.
nSolveIter 40;

    //- Maximum number of snapping relaxation iterations. Should
stop
    // before upon reaching a correct mesh.
nRelaxIter 8;
}

```

```

// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the
undistorted
    // size of the refined cell outside layer (true) or absolute
sizes (false).
relativeSizes true;

    // Per final patch (so not geometry!) the layer information
layers
    {
        wing
        {
            nSurfaceLayers 4;
        }
    }

    // Expansion factor for layer mesh
expansionRatio 1.2;

```

```

// Wanted thickness of final added cell layer. If multiple
layers
// is the thickness of the layer furthest away from the wall.
// Relative to undistorted size of cell outside layer.
// See relativeSizes parameter.
finalLayerThickness 0.7;

// Minimum thickness of cell layer. If for any reason layer
// cannot be above minThickness do not add layer.
// Relative to undistorted size of cell outside layer.
// See relativeSizes parameter.
minThickness 0.25;

// If points get not extruded do nGrow layers of connected faces
that are
// also not grown. This helps convergence of the layer addition
process
// close to features.
// Note: changed(corrected) w.r.t 17x! (didn't do anything in
17x)
nGrow 0;

// Advanced settings

// When not to extrude surface. 0 is flat surface, 90 is when
two faces
// are perpendicular
featureAngle 30;

// Maximum number of snapping relaxation iterations. Should stop
// before upon reaching a correct mesh.
nRelaxIter 5;

// Number of smoothing iterations of surface normals
nSmoothSurfaceNormals 1;

// Number of smoothing iterations of interior mesh movement
direction
nSmoothNormals 3;

// Smooth layer thickness over surface patches
nSmoothThickness 10;

// Stop layer growth on highly warped cells
maxFaceThicknessRatio 0.5;

// Reduce layer growth where ratio thickness to medial
// distance is large
maxThicknessToMedialRatio 0.3;

// Angle used to pick up medial axis points
// Note: changed(corrected) w.r.t 16x! 90 degrees corresponds to
130 in 16x.
minMedianAxisAngle 90;

// Create buffer region for new layer terminations

```

```

nBufferCellsNoExtrude 0;

// Overall max number of layer addition iterations. The mesher
will exit
// if it reaches this number of iterations; possibly with an
illegal
// mesh.
nLayerIter 50;
}

// Generic mesh quality settings. At any undoable phase these
determine
// where to undo.
meshQualityControls
{
    //- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 65;

    //- Max skewness allowed. Set to <0 to disable.
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //- Max concaveness allowed. Is angle (in degrees) below which
concavity
    // is allowed. 0 is straight face, <0 would be convex face.
    // Set to 180 to disable.
    maxConcave 80;

    //- Minimum pyramid volume. Is absolute volume of cell pyramid.
    // Set to a sensible fraction of the smallest cell volume
expected.
    // Set to very negative number (e.g. -1E30) to disable.
    minVol 1e-13;

    //- Minimum quality of the tet formed by the face-centre
    // and variable base point minimum decomposition triangles and
    // the cell centre. Set to very negative number (e.g. -1E30)
to
    // disable.
    // <0 = inside out tet,
    // 0 = flat tet
    // 1 = regular tet
    minTetQuality 1e-30;

    //- Minimum face area. Set to <0 to disable.
    minArea -1;

    //- Minimum face twist. Set to <-1 to disable. dot product of
face normal
    // and face centre triangles normal
    minTwist 0.05;

    //- minimum normalised cell determinant

```

```
//- 1 = hex, <= 0 = folded or flattened illegal cell
minDeterminant 0.001;

//- minFaceWeight (0 -> 0.5)
minFaceWeight 0.05;

//- minVolRatio (0 -> 1)
minVolRatio 0.01;

//must be >0 for Fluent compatibility
minTriangleTwist -1;

// Advanced

//- Number of error distribution iterations
nSmoothScale 4;
//- amount to scale back displacement at error points
errorReduction 0.75;
}

// Advanced

// Merge tolerance. Is fraction of overall bounding box of initial
mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1e-6;

//
*****
***** //
```

ANEXO E

E.1 Resultados apresentados por Hoang *et al.* (2013)

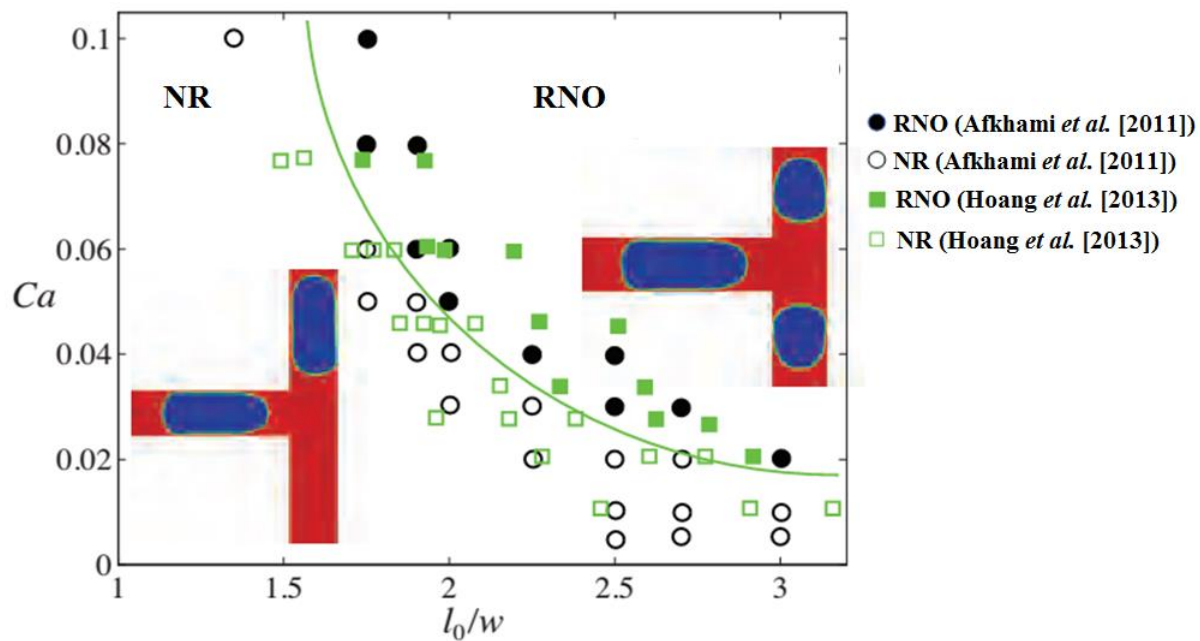


Figura E1: Resultados apresentados por Hoang *et al.* (2013) em comparação com os resultados de Afkhami *et al.* (2011). Fonte: Adaptado de Hoang *et al.* (2013).