UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Desenvolvimento da técnica PLIC-VOF no sistema HiG-Flow para simulação de escoamentos bifásicos

Aquisson Theyllon Gomes da Silva

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências
de Computação e Matemática Computacional (PPG-CMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Aquisson Theyllon Gomes da Silva

Desenvolvimento da técnica PLIC-VOF no sistema HiG-Flow para simulação de escoamentos bifásicos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Antonio Castelo Filho

**USP – São Carlos
Maio de 2020**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

d111d da Silva, Aquisson Theyllon Gomes
Desenvolvimento da técnica PLIC-VOF no sistema
HiG-Flow para simulação de escoamentos bifásicos /
Aquisson Theyllon Gomes da Silva; orientador
Antonio Castelo Filho. -- São Carlos, 2020.
87 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2020.

1. VOF. 2. Escoamento multifásico. 3. Tensão
superficial contínua. 4. HiG-Flow. 5. Diferenças
finitas. I. Filho, Antonio Castelo, orient. II.
Título.

Aquisson Theyllon Gomes da Silva

**Development of the VOF technique in the HiG-Flow system
for two-phase flow simulation**

Dissertation submitted to the Institute of Mathematics
and Computer Sciences – ICMC-USP – in
accordance with the requirements of the Computer
and Mathematical Sciences Graduate Program, for
the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Antonio Castelo Filho

USP – São Carlos
May 2020

À minha família.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida, por ter me capacitado e permitido chegar até aqui. Aos meus pais, Pedro e Elenita, e minha irmã, Thayne, que sempre me apoiaram e acreditaram na minha capacidade.

Agradeço aos professores que estiveram dispostos a ajudar e contribuir para um melhor aprendizado, em especial ao meu orientador, Antonio Castelo Filho.

Ao Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) por ter me dado a chance e todas as ferramentas que permitiram chegar hoje ao final deste ciclo de maneira satisfatória.

Agradeço, também, a todos os colegas e amigos do Laboratório de Matemática Aplicada e Computação Científica (LMACC). E um agradecimento especial aos amigos, Gustavo Miziara e Juniormar Organista, por terem trabalhado comigo de forma assídua nos últimos meses.

Finalmente, agradeço à PETROBRAS e CAPES pelo suporte financeiro durante o desenvolvimento deste projeto.

*“Consagre ao Senhor tudo o que você faz,
e os seus planos serão bem-sucedidos.”
(A Bíblia (PROVÉRBIOS, 16:3))*

RESUMO

SILVA, A. T. G. **Desenvolvimento da técnica PLIC-VOF no sistema HiG-Flow para simulação de escoamentos bifásicos**. 2020. 87 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Teve-se como objetivo estudar e implementar técnicas numéricas para a simulação computacional de escoamentos bifásicos de fluidos newtonianos, incompressíveis, imiscíveis e isotérmicos. Nesse tipo de escoamento, uma descrição consistente da interface entre os fluidos é de suma importância para garantir uma simulação numérica precisa. O método da fração de volume de fluidos (VOF) foi utilizado para localizar e reconstruir a interface. Considerou-se o modelo de tensão interfacial contínua (CSF) para o balanço de forças na interface. As equações de Navier-Stokes foram resolvidas utilizando o sistema HiG-Flow, no qual o método VOF foi acoplado. Para a análise dos resultados numéricos obtidos, os métodos implementados foram verificados utilizando dados e problemas clássicos da literatura, e apresentaram boa concordância.

Palavras-chave: Escoamentos Multifásicos, Método VOF, Método PLIC-VOF, HiG-Flow, HiG-Tree, Força de Tensão Superficial.

ABSTRACT

SILVA, A. T. G. **Development of the VOF technique in the HiG-Flow system for two-phase flow simulation.** 2020. 87 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

The purpose of this master thesis was to study and implement numerical techniques for the computational simulation of incompressible, immiscible and isothermal two-phase flows of Newtonian fluids. In two-phase flows, a consistent description of the interface between fluids is very important to ensure accurate numerical simulation. The volume of fluid (VOF) method was used to locate and reconstruct the interface. The continuum interfacial force (CSF) model was considered for the balance of forces at the interface. Navier-Stokes equations were solved using the HiG-Flow solver, in which the VOF method was coupled. For the analysis of numerical results, the implemented methods were verified using data and classical problems of the literature, and proved to be consistent.

Keywords: Multiphase Flow, VOF Method, PLIC-VOF Method, HiG-Flow, HiG-Tree, Surface Tension.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura de dados HiG-Tree.	30
Figura 2 – Estêncil do método de diferenças finitas de segunda ordem, onde as incógnitas estão dispostas no centro das células. x_c, x_l, x_r, x_b e x_t são aproximações de valores posicionados no centro, à esquerda, à direita, em cima e embaixo no estêncil de 5 pontos.	31
Figura 3 – Exemplo de uma malha deslocada, em que os triângulos representam $\mathbf{u}$, e os pontos centrados nas células representam p, ρ, μ, f	37
Figura 4 – Os pontos representam as pressões necessárias para o cálculo do gradiente da pressão, avaliado em $(i + \frac{1}{2}, j)$, indicado por $\mathbf{x}$	38
Figura 5 – Pontos necessários para o cálculo do termo viscoso avaliado em $(i + \frac{1}{2}, j)$, indicado por $\mathbf{x}$. Os triângulos representam as velocidades necessárias à discretização. Os pontos (vermelhos) indicam a posição das derivadas da velocidade intermediária.	39
Figura 6 – Pontos necessários para o cálculo da equação de Poisson para pressão avaliada em (i, j) , indicada por $\mathbf{x}$. Os triângulos representam as velocidades necessárias à discretização. Os pontos (vermelhos) indicam a posição das derivadas da pressão.	40
Figura 7 – Frações de volume de fluido em um volume de controle.	44
Figura 8 – Interface definida por $x^2 + y^2 = R^2$, em uma malha cartesiana, onde os pontos internos (azuis) representam o fluido de referência para a inicialização de f	46
Figura 9 – Possíveis interseções da interface com a célula computacional.	47
Figura 10 – Stencil 3x3 entorno de uma célula interfacial.	48
Figura 11 – Stencil 3x7 entorno de uma célula interfacial, onde as alturas, h_j , são representadas pelas linhas tracejadas.	49
Figura 12 – stencil adaptativo com funções de altura independentes.	50
Figura 13 – Stencil adaptativo com alturas, h_i , com origem em comum.	50
Figura 14 – Stencil com célula interfacial isolada.	51
Figura 15 – Stencil descentrado com célula interfacial isolada.	52
Figura 16 – Área sob a reta geral cortando a célula computacional.	54
Figura 17 – Áreas fora da célula computacional que devem ser subtraídas na eq.(4.24).	55
Figura 18 – Representação geométrica da interface PLIC quando $R_x = 0$ e $R_y = 0$	56
Figura 19 – Representação geométrica da interface PLIC quando $R_x = 0$ e $R_y = 1$	57
Figura 20 – Representação geométrica da interface PLIC quando $R_x = 1$ e $R_y = 1$	58

Figura 21 – Representação geométrica da interface PLIC quando $\mathbf{n} = (0, -n_y)$	59
Figura 22 – Fração de área de fluido (região escura) a ser advectada para a célula vizinha (direita).	61
Figura 23 – Sobreposição de duas regiões de uma célula que serão advectadas em sentidos opostos.	62
Figura 24 – Configuração inicial.	66
Figura 25 – Erros de inicialização da fração de volume, f , em função da resolução da malha, dividindo cada célula interfacial em 8^2 subcélulas.	67
Figura 26 – Erros na estimativa da curvatura interfacial, em função da resolução da malha, usando o método HF, com $n_{sc} = 8^2$	68
Figura 27 – Erros na estimativa da curvatura interfacial, em função da resolução da malha, via método HF, com $n_{sc} = 64^2$	69
Figura 28 – Erros na estimativa do vetor normal à interface, em função da resolução da malha, usando o método HF, com $n_{sc} = 8^2$	70
Figura 29 – Advecção de uma interface, inicialmente circular, com campo de velocidade prescrito, no tempo $t = 1$	71
Figura 30 – Campo de pressão oriundo do teste da gota estática em uma malha 64×64 , no tempo $t = 1$	73
Figura 31 – Análise de convergência do erro no salto da pressão sobre uma gota de etanol em diferentes resoluções de malha, no tempo $t = 1$	74
Figura 32 – Comparação dos resultados da oscilação da posição de um ponto, localizado ao norte de uma gota elíptica, obtidos com o HiG-Flow (32×32 , 64×64 e 128×128 células) e Štrubelj, Tiselj e Mavko (2009) (128×128 células).	75
Figura 33 – Configuração inicial do teste da bolha ascendente.	76
Figura 34 – Bolha ascendente para diferentes resoluções de malha no tempo $t = 3$ s. A linha azul (em (b), (c) e (d)) corresponde ao centro do canal na direção y	77
Figura 35 – Evolução do centro de massa da bolha na direção vertical (y), para diferentes resoluções de malha, até o tempo $t = 3$ s.	79
Figura 36 – Velocidade de ascensão da bolha, para diferentes resoluções de malha, até o tempo dimensional $t = 3$ s.	80
Figura 37 – Comparação dos resultados do teste da bolha ascendente no tempo $t = 0.5$ s. A linha contínua (preta) representa a bolha gerada no HiG-Flow e a linha tracejada (azul) representa a bolha reportada por Kalland (2008). A linha azul dividindo o domínio ($[0, 2] \times [0, 4]$) corresponde ao centro do domínio na direção y	81

LISTA DE TABELAS

Tabela 1	–	Análise de convergência do erro oriundo da inicialização de f , com diferentes tamanhos de malha, na norma L_∞	67
Tabela 2	–	Análise de convergência do erro oriundo do cálculo da curvatura, via método HF, para diferentes resoluções de malha, com $nsc = 8^2$	68
Tabela 3	–	Análise de convergência do erro oriundo do cálculo da curvatura, via método HF, para diferentes resoluções de malha, com $nsc = 64^2$	69
Tabela 4	–	Análise de convergência do erro oriundo do cálculo do vetor normal à interface, via método HF, para diferentes resoluções de malha, com $nsc = 8^2$. . .	70
Tabela 5	–	Análise do erro entre a área exata e numérica dentro da interface, em diferentes resoluções de malha, no tempo $t = 1$	72
Tabela 6	–	Parâmetros físicos para a simulação da gota estática.	72
Tabela 7	–	Análise de convergência do erro no salto da pressão sobre uma gota de etanol em diferentes resoluções de malha, no tempo $t = 1$	73
Tabela 8	–	Parâmetros físicos para a simulação da gota oscilante.	75
Tabela 9	–	Parâmetros físicos e adimensionais para a simulação da bolha ascendente. .	77
Tabela 10	–	Velocidade máxima de ascensão da bolha e posição final do centro de massa no eixo y , usando os softwares: TP2D, FreeLIFE e MooNMD	78
Tabela 11	–	Velocidade máxima de ascensão da bolha e posição final do centro de massa no eixo y	79
Tabela 12	–	Parâmetros físicos para a simulação da bolha ascendente.	80

LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões
BCGS	BiConjugate Gradient Squared
CSF	Continuum Surface Force
HF	Height Function
HiG	Hierarchical Grid
PLIC	Piecewise-Linear Interface Calculation
VOF	Volume of Fluid

LISTA DE SÍMBOLOS

ρ — densidade

$\mathbf{u}$ — campo de velocidade

∇ — operador diferencial

p — pressão

μ — viscosidade

$\mathbf{g}$ — campo gravitacional

σ — coeficiente de tensão superficial

κ — curvatura

$\mathbf{n}$ — vetor normal

$\mathbf{x}$ — vetor posição

Re — número de Reynolds

Bo — número de Bond

f — fração de volume de fluido

t — tempo

h — função altura

κ — curvatura

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Justificativa	25
1.2	Objetivo Geral	26
1.3	Objetivo Específico	27
1.4	Procedimentos metodológicos	27
1.5	Esboço da dissertação	27
2	SISTEMAS HIG-TREE E HIG-FLOW	29
2.1	O sistema HiG-Tree	29
2.2	O sistema HiG-Flow	31
2.2.1	<i>Discretização espaço-temporal</i>	31
2.2.2	<i>Acoplamento velocidade-pressão</i>	32
2.2.3	<i>Discretização do termo convectivo</i>	32
3	METODOLOGIA	33
3.1	Formulação matemática	33
3.1.1	<i>Introdução teórica</i>	33
3.1.2	<i>Adimensionalização</i>	34
3.2	Métodos numéricos	35
3.3	Discretizações	36
3.3.1	<i>Gradiente da pressão</i>	37
3.3.2	<i>Densidade</i>	37
3.3.3	<i>Termos viscosos</i>	38
3.3.4	<i>Equação de Poisson para a pressão</i>	39
4	TÉCNICA DA FRAÇÃO DO VOLUME DE FLUIDOS (VOF)	43
4.1	Introdução teórica	43
4.2	Inicialização da fração de volume	45
4.2.1	<i>Método de inicialização</i>	45
4.3	Cálculo da curvatura e do vetor normal à interface	47
4.3.1	<i>O método da Função Altura (HF)</i>	48
4.4	Método PLIC-VOF	52
4.4.1	<i>Reconstrução da interface</i>	53
4.4.1.1	<i>Expressão para o cálculo da área abaixo da interface reconstruída</i>	54

4.4.1.2	<i>Descrição do algoritmo de reconstrução da interface</i>	55
4.4.2	<i>Advecção da interface</i>	59
4.4.2.1	<i>Restrição para o passo de tempo</i>	62
4.5	<i>Implementação da tensão interfacial</i>	63
5	RESULTADOS NUMÉRICOS	65
5.1	Testes de verificação dos métodos	65
5.1.1	<i>Análise do método de inicialização de f</i>	66
5.1.2	<i>Análise do cálculo da curvatura e do vetor normal à interface</i>	67
5.1.3	<i>Análise do algoritmo de reconstrução e advecção PLIC-VOF</i>	71
5.2	Simulações numéricas de escoamentos bifásicos	72
5.2.1	<i>Gota estática</i>	72
5.2.2	<i>Gota oscilante</i>	74
5.2.3	<i>Bolha ascendente</i>	76
6	CONCLUSÕES	83
6.1	Considerações finais	83
6.2	Trabalhos futuros	84
	REFERÊNCIAS	85

INTRODUÇÃO

1.1 Justificativa

Deparar-se com fenômenos envolvendo escoamentos de fluidos com propriedades físicas diferentes (e.g., escoamentos bifásicos) é comum em Mecânica dos Fluidos Computacional. O estudo numérico desses fenômenos tem por objetivo desenvolver e/ou aprimorar técnicas e métodos numéricos robustos, visando aplicações na indústria. Áreas como Engenharia Nuclear, Física ou Química esbarram frequentemente com esses fenômenos. Destaca-se também o aparecimento de escoamentos bifásicos na indústria de petróleo, onde esse fenômeno aparece em várias fases do processo de produção, desde a extração do petróleo dos reservatórios até os processos de refino. Em muitos casos, o tratamento desses fenômenos demandam escalas de tempo e espaço extremamente pequenas (ou grandes), tornando sua análise, experimentalmente, inviável. O uso de simulações computacionais pode garantir a viabilidade dessas análises.

A modelagem de escoamentos bifásicos não é uma tarefa fácil, tanto do ponto de vista físico quanto numérico. De acordo com [Marek, Aniszewski e Boguslawski \(2008\)](#), a complexidade desse fenômeno advém da presença de uma interface em que propriedades físicas (i.e. densidade, viscosidade, pressão) mudam de forma descontínua. Ademais, fenômenos que abrangem escoamentos bifásicos tendem a gerar interfaces com geometrias complexas, podendo, ainda, apresentar grandes variações geométricas com a evolução do tempo. Logo, uma descrição consistente da interface é de suma importância para garantir uma simulação numérica precisa.

Devido ao avanço da computação, várias técnicas e métodos numéricos vêm sendo desenvolvidos para a captura da dinâmica de escoamentos bifásicos, visando a representação precisa da interface. Destacam-se como os mais conhecidos na literatura os métodos: Front-Tracking ([TRYGGVASON *et al.*, 2001](#)), Level-Set ([OSHER; FEDKIW; PIECHOR, 2004](#)) e o VOF ([HIRT; NICHOLS, 1981](#)).

De acordo com [Lages \(2016\)](#), o método Front-Tracking consiste em resolver as equações

de Navier-Stokes em uma malha fixa (euleriana) e a interface entre as fases do escoamento é representada e acompanhada por uma malha independente (lagrangeana), que armazena a posição da interface.

No método Level-Set, uma função suave, chamada função distância, é usada para representar a interface. A informação da função distância é utilizada para advectar a interface através da resolução de uma equação de transporte.

Neste projeto, optou-se trabalhar com o método VOF, descrito detalhadamente no capítulo 4, por apresentar vantagens em relação aos demais métodos supracitados, tais como relativa simplicidade para sua implementação e propriedades de conservação de volume. De forma análoga ao método Level-Set, a advecção da interface no método VOF também é feita resolvendo-se uma equação de transporte, porém, usando a informação da fração de volume. Visando garantir a máxima eficiência do método VOF, foi utilizado um algoritmo geométrico para a advecção da interface, o chamado PLIC (do inglês, *Piecewise Linear Interface Calculation*) do VOF, uma vez que o VOF não apresenta bom comportamento com algoritmos de advecção algébricos.

Para resolver os problemas envolvendo escoamentos bifásicos, utilizou-se o sistema HiG-Flow (software ainda em desenvolvimento), onde o método VOF foi implementado, para fazer todas as simulações numéricas deste projeto.

O sistema HiG-Flow está sendo desenvolvido com base em simulações numéricas, para a modelagem de escoamentos bifásicos de fluidos incompressíveis, utilizando plataformas multiprocessadas. Esse sistema está acoplado ao sistema HiG-Tree, e, até a presente data, está operacional para simular escoamentos monofásicos dos tipos: newtoniano, newtoniano generalizado e viscoelástico, e agora, com a conclusão deste projeto, escoamentos bifásicos, em duas dimensões (2D), de fluidos newtonianos com densidades e viscosidades variáveis.

Os sistemas HiG-Tree e HiG-Flow estão sendo projetados para simular tanto problemas simples quanto complexos, isso em termos geométricos ou modelos constitutivos e métodos numéricos empregados. Esses sistemas vêm sendo desenvolvido pelo grupo de Mecânica dos Fluidos Computacional do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP), sob orientação do professor Dr. Antonio Castelo Filho.

Todas as implementações foram feitas utilizando a linguagem de programação C. Além disso, os softwares MATLAB e GeoGebra foram utilizados para a produção dos gráficos e figuras desta dissertação.

1.2 Objetivo Geral

Neste projeto de mestrado, teve-se como objetivo geral tornar o sistema HiG-Flow operacional para simulações numéricas de escoamentos bifásicos, tendo como foco o desenvolvimento e verificação de métodos numéricos utilizados na modelagem de escoamentos dessa natureza.

Para isso, foi implementado no sistema HiG-Flow o método VOF, optando-se por um algoritmo geométrico para a advecção da interface conhecido como PLIC-VOF.

1.3 Objetivo Específico

Adquirir um amplo conhecimento na área de Mecânica dos Fluidos Computacional, por meio da análise de problemas físicos e matemáticos, bem como no desenvolvimento de habilidades em programação computacional.

Desenvolver capacidades necessárias à execução de atividades em grupos de pesquisa científica, mas também a capacidade de trabalho independente e criativo.

1.4 Procedimentos metodológicos

Abaixo, seguem as etapas que foram cumpridas, ao longo de todo o período de mestrado, para que os objetivos propostos, neste projeto, fossem atingidos satisfatoriamente. As etapas do projeto foram divididas da seguinte forma:

1. Estudar os sistemas HiG-Tree e HiG-Flow.
2. Levantamentos bibliográficos para a verificação do estado da arte, buscando publicações relevantes sobre o método VOF.
3. Implementação do método VOF no sistema HiG-Flow, com as implementações de:
 - Técnica de inicialização do VOF.
 - Técnicas para o cálculo da curvatura e do vetor normal à interface.
 - Técnica de reconstrução e advecção da interface (PLIC-VOF).
 - Modelo para o cálculo da tensão interfacial.
4. Verificação dos resultados por meio de:
 - Experimentos numéricos.
 - Comparação com dados e testes clássicos da literatura.

1.5 Esboço da dissertação

Esta dissertação está estruturada da seguinte maneira:

- Capítulo 1: é feita uma contextualização do projeto, começando com uma justificativa ao trabalho, apresentando os propósitos que se pretendia alcançar, e, finalmente, as etapas de execução para atingir os objetivos pretendidos.

- Capítulo 2: são apresentados os sistemas HiG-Tree e HiG-Flow, bem como os módulos e métodos numéricos que constituem esses sistemas.
- Capítulo 3: é apresentado o sistema de equações que modela os escoamentos de fluidos estudados neste projeto, além das discretizações utilizadas.
- Capítulo 4: encontra-se uma descrição detalhada do método do volume de fluidos (VOF), onde é descrito o método de inicialização do VOF, técnicas de estimativa da curvatura e do vetor normal à interface, algoritmo de reconstrução e advecção da interface, e, finalmente, modelo de representação da tensão interfacial.
- Capítulo 5: são mostrados os resultados alcançados com este trabalho, por meio da apresentação de verificações e análises dos métodos numéricos empregados.
- Capítulo 6: são apresentadas as conclusões do trabalho, bem como suas principais contribuições.

SISTEMAS HiG-TREE E HiG-FLOW

Todas as simulações numéricas contidas neste trabalho foram realizadas utilizando os sistemas HiG-Tree e HiG-Flow. Este capítulo dedica-se a apresentação desses sistemas. Para o estudo do HiG-Tree, seguiu-se como principal referência o trabalho de (SOUSA *et al.*, 2019).

2.1 O sistema HiG-Tree

O sistema HiG-Tree é responsável pela estrutura de dados, domínios, resolvedores de sistemas lineares e não lineares, aproximações e interpolações. Atualmente, esse sistema é formado por quatro módulos, os quais são: estrutura de dados HiG-Tree, resolvedores de sistemas lineares e não lineares, particionamento de domínios e sub-domínios, aproximações e interpolações.

Descrição dos módulos do sistema HiG-Tree

1. **Estrutura de dados HiG-Tree:** a estrutura de dados HiG-Tree representa uma malha cartesiana do tipo hierárquico, podendo ser constituída por elementos de tamanhos diferentes. Essa estrutura tem por base as subdivisões (refinamentos) espaciais, de forma recursiva, dos seus elementos, possibilitando, assim, sua representação em uma árvore *m-tree* (vide figura 1). Na estrutura de dados HiG-Tree, processos de refinamentos arbitrários da malha podem ser aplicados em qualquer nível da árvore, gerando uma malha não-estruturada. Destacam-se como exemplos clássicos desse tipo de malha as estruturas de dados *Quadtree* (bidimensional) e *Octree* (tridimensional). Apesar da representação ilustrada na figura 1 ser bidimensional, a extensão para qualquer número de dimensão é simples, uma vez que a estrutura é suficientemente genérica para isso.

Figura 1 – Estrutura de dados HiG-Tree.

Fonte: [Sousa et al. \(2019\)](#)

2. **Resolvedores de sistemas lineares e não lineares:** esse módulo é composto por duas bibliotecas de alta performance com rotinas para a resolução de sistema lineares e não lineares, baseadas em computação paralela, com memória compartilhada e distribuída (arquitetura híbrida). Estas bibliotecas são a **HYPRE** (do inglês, *High Performance Pre-conditioners*), que, segundo [Falgout e Yang \(2002\)](#), é uma biblioteca de software para a solução de sistemas lineares grandes e esparsos em computadores maciçamente paralelos, e a **PETSc** (do inglês, *Portable, Extensible Toolkit for Scientific Computation*), que oferece um conjunto de estrutura de dados e rotinas para a solução de problemas modelados por equações diferenciais parciais. Tais bibliotecas possuem grandes vantagens no desenvolvimento de códigos em paralelo, e já foram otimizadas e testadas em diversos códigos de simulação de escoamentos de fluidos.
3. **Particionamento de domínios e subdomínios:** os domínios gerados são compostos por blocos que podem ter complexidade para simular problemas em canais com geometrias complexas. Cada bloco é discretizado pela estrutura HiG-Tree, que permite refinamento espacial adequado aos problemas simulados. Tais domínios são particionados utilizando a biblioteca ZOLTAN-TRILINOS que permite uma boa distribuição de carga entre os processos durante a execução. Este módulo também permite o armazenamento e a enumeração de propriedades representadas nas células e facetas da malha.
4. **Aproximações e interpolações:** responsável pelas aproximações de propriedades descritas nas células e facetas, sendo possível escolher a ordem de aproximação polinomial que é utilizada pelo método dos mínimos quadrados moveis (MQM), proposto por [Sousa et al. \(2019\)](#).

Em se tratando de malhas compostas por elementos com diferentes níveis de refinamento, métodos numéricos baseados em aproximações de diferenças finitas em malhas uniformes cartesianas tornam-se limitados, pois demandarão interpolações espaciais em pontos incógnitas do estêncil, como ilustrado na figura 2. A precisão desse tipo de método depende vigorosamente das características geométricas da malha. O método MQM evita

essa dependência geométrica, usando uma nuvem de pontos vizinhos à incógnita para aplicar a interpolação.

Figura 2 – Estêncil do método de diferenças finitas de segunda ordem, onde as incógnitas estão dispostas no centro das células. x_c , x_l , x_r , x_b e x_t são aproximações de valores posicionados no centro, à esquerda, à direita, em cima e embaixo no estêncil de 5 pontos.

Fonte: Elaborada pelo autor.

2.2 O sistema HiG-Flow

O sistema HiG-Flow possibilita a simulação numérica de escoamentos monofásicos, em qualquer dimensão, de fluidos newtonianos, newtonianos generalizados e viscoelásticos (com técnicas de estabilização numérica), e escoamentos bifásicos, 2D, utilizando a técnica VOF (vide capítulo 4). Esse sistema vem sendo projetado de forma modular, permitindo que novas técnicas e métodos possam ser implementados, no mesmo, de forma fácil. Além disso, implementou-se o código de forma que o usuário possa escolher as seguintes características à simulação: dimensão, módulos a serem utilizados em seu programa (e.g., monofásico, newtoniano, newtoniano generalizado, viscoelástico e bifásico) e técnicas numéricas por arquivo de entrada de dados (e.g., método da projeção, esquemas numéricos para o termo convectivo, modelo da equação constitutiva para escoamentos viscoelásticos).

Nas subseções abaixo, encontram-se as técnicas e métodos numéricos implementados no sistema HiG-Flow:

2.2.1 Discretização espaço-temporal

Foram implementados os mais tradicionais métodos para avanço temporal. Os métodos explícitos implementados são: Euler explícito, Runge-Kutta TVD de segunda ordem (ou mé-

todo de Euler modificado) e Runge-Kutta TVD de terceira ordem. Já os métodos implícitos implementados no sistema são: Euler implícito, Crank-Nicolson e BDF (do inglês, *Backward Differentiation Formula*) de segunda ordem.

2.2.2 Acoplamento velocidade-pressão

Atualmente, dois esquemas estão disponíveis no sistema para a obtenção do campo de pressão, estes são: método da projeção incremental (segunda ordem no tempo) e método da projeção não incremental (primeira ordem no tempo).

2.2.3 Discretização do termo convectivo

Em se tratando dos termos convectivos, estão implementados, até o momento, quatro métodos clássicos para a discretização dos mesmos, estes são: central de diferenças finitas, Upwind de primeira ordem, Quick de segunda ordem (descrito no trabalho de [Leonard \(1979\)](#)) e Cubista de segunda ordem (descrito no trabalho de [Alves, Oliveira e Pinho \(2003\)](#)).

METODOLOGIA

Este capítulo dedica-se à apresentação do sistema de equações que modela o tipo de escoamento abordado nesta dissertação, bem como os métodos numéricos e discretizações utilizados para resolver esse sistema.

3.1 Formulação matemática

Para a apresentação do conjunto de equações que modela os escoamentos tratados nesta dissertação, esta seção foi dividida em duas subseções, estas são: Introdução teórica e Adimensionalização.

3.1.1 Introdução teórica

Fenômenos envolvendo escoamento de fluidos podem ser modelados por um sistema de equações diferenciais não lineares, conhecido como equações de Navier-Stokes. Esse sistema de equações é constituído por três leis de conservação, estas são: conservação da massa, conservação da quantidade de movimento linear e conservação da energia. Além disso, problemas envolvendo escoamentos bifásicos, requerem a inclusão, no sistema de equações, de uma equação que modele a advecção da interface, esta equação será descrita no capítulo 4.

Neste trabalho, todas as formulações foram feitas para escoamentos de fluidos incompressíveis e isotérmicos. Assim sendo, a densidade e viscosidade são constantes em cada fase, mas variam de fase para fase, e a lei de conservação de energia foi omitida no sistema de equações. Portanto, o conjunto de equações que modela escoamentos bifásicos, isotérmicos, de fluidos incompressíveis e imiscíveis é descrito pelas seguintes expressões:

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

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}(\nabla \cdot \mathbf{u}) \right) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})] + \rho \mathbf{g} + \sigma \kappa \mathbf{n} \delta, \quad (3.2)$$

onde (3.1) e (3.2) representam as conservações de massa e quantidade de movimento linear, respectivamente, sendo $\mathbf{u} = (u, v)$ o campo de velocidade do escoamento, ρ a densidade, p a pressão, μ a viscosidade absoluta, $\mathbf{g}$ o campo gravitacional, σ o coeficiente de tensão interfacial, κ e $\mathbf{n}$ a curvatura e o vetor normal à interface, respectivamente, e δ é uma função que assume valor um na interface e zero no restante do domínio. Todas as variáveis estão em função do tempo e do espaço.

3.1.2 Adimensionalização

O sistema HiG-Flow opera tanto na forma dimensional quanto adimensional. Os parâmetros adimensionais são dados por:

- $\mathbf{x} = L\mathbf{x}^*$, em que L é o comprimento característico;
- $\mathbf{u} = (gL)^{\frac{1}{2}} \mathbf{u}^*$, em que g é a constante gravitacional;
- $t = \left(\frac{L}{g}\right)^{\frac{1}{2}} t^*$;
- $\rho = \rho_1 \rho^*$, em que ρ_1 representa a densidade característica;
- $p = \rho_1 g L p^*$;
- $\mu = \mu_1 \mu^*$, em que μ_1 representa a viscosidade característica;
- $\kappa = \kappa^* L^{-2}$;
- $\nabla = \frac{1}{L} \nabla^*$,

os termos com super-índice (*) denotam os termos adimensionais.

Aplicando esses parâmetros à equação (3.2) e manipulando-a algebricamente, tem-se:

$$\rho^* \frac{\partial \mathbf{u}^*}{\partial t^*} + \rho^* \mathbf{u}^* \cdot \nabla^* \mathbf{u}^* = -\nabla^* p^* + \left(\frac{\mu_1}{\rho_1 g^{\frac{1}{2}} L^{\frac{3}{2}}} \right) \nabla^* \cdot [\mu^* (\nabla^* \mathbf{u}^* + (\nabla^*)^T \mathbf{u}^*)] + \left(\frac{\sigma}{\rho_1 g L^2} \right) \cdot \kappa^* \mathbf{n} \delta(x - x_f) + \rho^* \frac{\mathbf{g}}{g}. \quad (3.3)$$

Logo,

$$\rho^* \frac{\partial \mathbf{u}^*}{\partial t^*} + \rho^* \mathbf{u}^* \cdot \nabla^* \mathbf{u}^* = -\nabla^* p^* + \frac{1}{Re} \nabla^* \cdot \{ \mu^* [\nabla^* \mathbf{u}^* + (\nabla^*)^T \mathbf{u}^*] \} + \frac{1}{Bo} \kappa \mathbf{n} \delta(x - x_f) + \rho^* \frac{\mathbf{g}}{g}. \quad (3.4)$$

Assim, omitindo * por conveniência, a equação de Navier-Stokes adimensionalizada é dada por:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho Re} \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla^T \mathbf{u})] + \frac{1}{\rho Bo} \kappa \mathbf{n} \delta(x - x_f) + \frac{\mathbf{g}}{g}, \quad (3.5)$$

em que os parâmetros adimensionais Re e Bo representam, na devida ordem, o número de Reynolds, que descreve a razão entre as forças inerciais e viscosas, e o número de Bond (também conhecido como número de Eötvös), que descreve a razão entre a força devido ao campo gravitacional e a tensão superficial.

3.2 Métodos numéricos

No sistema HiG-Flow, para resolver a equação de Navier-Stokes, adotou-se métodos de projeção (não incremental e incremental) que possibilitam o desacoplamento dos campos de pressão e velocidade. De acordo com [Lages \(2016\)](#), a versão não incremental do método é $O(\Delta t)$, e a versão incremental é $O(\Delta t^2)$, em se tratando do erro proveniente do desacoplamento pressão-velocidade. Optando-se pela discretização da equação de Navier-Stokes por um esquema explícito no tempo, ambos os métodos são $O(\Delta t)$. É possível obter métodos de projeção com maior ordem de precisão optando-se por esquemas de alta ordem para discretização temporal. Abaixo, encontram-se uma breve descrição de ambos os métodos de projeção.

Discretizando as equações de Navier-Stokes (3.1 e 3.2) pelo método de Euler explícito, têm-se:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla \mathbf{u}^n) + \frac{1}{\rho^n} \{ -\nabla p^{n+1} + \nabla \cdot [\mu (\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T)] + \mathbf{F}^n \}, \quad (3.6)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (3.7)$$

em que $\mathbf{F}^n$ representa as forças de tensão interfacial e gravitacional, e $\mathbf{u}^n$ é a velocidade, no n -ésimo passo de tempo.

No método da projeção não incremental, ignorando, inicialmente, o termo do gradiente da pressão, calcula-se a velocidade intermediária, $\mathbf{u}^*$, como segue:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla \mathbf{u}^n) + \frac{1}{\rho^n} \{ \nabla \cdot [\mu (\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T)] + \mathbf{F}^n \}, \quad (3.8)$$

Em seguida, corrige-se a velocidade intermediária, acrescentando o gradiente da pressão, para obter-se a solução final para $\mathbf{u}^{n+1}$:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho^n} \nabla p^{n+1}, \quad (3.9)$$

ou ainda,

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^n} \nabla p^{n+1}. \quad (3.10)$$

Aplicando o divergente à equação (3.10) e considerando a condição de incompressibilidade dos fluidos (i.e., $\nabla \cdot \mathbf{u}^{n+1} = 0$), a pressão pode ser obtida resolvendo-se a equação de Poisson:

$$\nabla \cdot \left(\frac{1}{\rho^n} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (3.11)$$

e, sabendo-se o valor de p , pode-se atualizar a velocidade $\mathbf{u}^{n+1}$ na equação (3.10).

Para o método da projeção incremental, considera-se p^n no lugar de p^{n+1} , na equação (3.6), obtendo assim:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla \mathbf{u}^n) + \frac{1}{\rho^n} \{ -\nabla p^n + \nabla \cdot [\mu (\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T)] + \mathbf{F}^n \}. \quad (3.12)$$

No segundo passo, subtraindo a equação (3.12) de (3.6), tem-se:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho^n} \nabla (p^{n+1} - p^n) = -\frac{1}{\rho^n} \nabla \Delta p. \quad (3.13)$$

Assim como no método não incremental, a pressão pode ser obtida resolvendo-se a equação de Poisson:

$$\nabla \cdot \left(\frac{1}{\rho^n} \nabla \Delta p \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \quad (3.14)$$

Em ambos os métodos, existe uma equação de Poisson para a pressão da forma:

$$\nabla \cdot \left(\frac{1}{\rho^n} \nabla \phi \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (3.15)$$

que demanda certos cuidados durante sua resolução, uma vez que, neste trabalho, ρ não é constante em todo o domínio. Na próxima seção, encontra-se uma opção para a resolução de (3.15).

3.3 Discretizações

No sistema HiG-Flow, as equações (3.6) e (3.15) são discretizadas por esquemas de diferenças finitas em uma malha deslocada. Um exemplo desse tipo de malha pode ser observado na figura 3, de modo que as variáveis estão ordenadas da seguinte forma: as velocidades, $\mathbf{u}$, são avaliadas nas facetas da célula, e as demais variáveis (p, ρ, μ, f) são avaliadas no centro. A variável f representa a fração de volume de fluidos e será descrita detalhadamente no capítulo 4. De acordo com Lages (2016), o uso desse tipo de malha evita instabilidades numéricas oriundas do acoplamento da pressão com a velocidade em escoamentos incompressíveis. Uma vez que todas as simulações deste projeto foram realizadas utilizando malhas cartesianas uniformes (i.e., todos os elementos têm o mesmo nível de refinamento), todas as discretizações que serão apresentadas, a partir daqui, foram feitas utilizando a notação (i, j) no lugar de (x_c) (como ilustrado na figura 2) para indicar o centro de uma célula computacional.

Figura 3 – Exemplo de uma malha deslocada, em que os triângulos representam $\mathbf{u}$, e os pontos centrados nas células representam p, ρ, μ, f .

Fonte: Elaborada pelo autor.

Seguem abaixo as descrições de alguns esquemas implementados no sistema HiG-Flow, que foram utilizados nas simulações numéricas contidas neste trabalho. Em vista disso, sem perda de generalidade, considera-se um ponto localizado na faceta de uma célula com índice $(i + \frac{1}{2}, j)$, a exemplo da figura 4.

3.3.1 Gradiente da pressão

Na equação (3.6), o gradiente da pressão é dado por:

$$\left(\frac{1}{\rho} \nabla p \right)_{i+\frac{1}{2},j} = \left[\frac{1}{\rho_{i+\frac{1}{2},j}} \left(\frac{\partial p}{\partial x} \right)_{i+\frac{1}{2},j} \right]. \quad (3.16)$$

Logo, discretizando (3.16) por diferenças centradas, tem-se:

$$\frac{\partial p_{i+\frac{1}{2},j}}{\partial x} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x}. \quad (3.17)$$

A representação geométrica dessa discretização pode ser observada na figura 4.

3.3.2 Densidade

Inicialmente, a informação da densidade está armazenada no centro das células. Nas discretizações realizadas no presente trabalho, necessita-se da avaliação de ρ nas facetas das células, e, para isso, calculou-se uma aproximação pela média dos valores da densidade das células vizinhas:

$$\rho_{i+\frac{1}{2},j} \approx \frac{\rho_{i+1,j} + \rho_{i,j}}{2}. \quad (3.18)$$

Figura 4 – Os pontos representam as pressões necessárias para o cálculo do gradiente da pressão, avaliado em $(i + \frac{1}{2}, j)$, indicado por $\mathbf{x}$.

Fonte: Elaborada pelo autor.

3.3.3 Termos viscosos

Destaca-se que a discretização dos termos viscosos tem um papel fundamental nos resultados numéricos. Isso vem do fato de que em escoamentos bifásicos a viscosidade, μ , tal como a densidade, não é constante em todo o domínio, variando de uma fase para outra. Neste trabalho, adotou-se a seguinte estratégia para a discretização desses termos:

$$\nabla \cdot (\mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})) = \begin{bmatrix} \partial_x(2\mu \partial_x u) + \partial_y[\mu(\partial_y u + \partial_x v)] \\ \partial_x[\mu(\partial_x v + \partial_y u)] + \partial_y(2\mu \partial_y v) \end{bmatrix} \quad (3.19)$$

Logo, o termo não-misto, na primeira linha de (3.19), é aproximado por um esquema de diferenças centradas, como segue:

$$\left[\frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) \right]_{i+\frac{1}{2},j} \approx \frac{2}{\Delta x} \left[\mu_{i+1,j} \left(\frac{\partial u}{\partial x} \right)_{i+1,j} - \mu_{i,j} \left(\frac{\partial u}{\partial x} \right)_{i,j} \right], \quad (3.20)$$

onde as derivadas de primeira ordem são aproximadas por:

$$\left(\frac{\partial u}{\partial x} \right)_{i,j} \approx \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}. \quad (3.21)$$

A aproximação da derivada avaliada em $(i + 1, j)$, em (3.20), é obtida de forma análoga.

Já o termo misto, na primeira linha de (3.19), é calculado como:

$$\left\{ \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \right\}_{i+\frac{1}{2},j} \approx \frac{1}{\Delta y} \left[\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)_{i+\frac{1}{2},j+\frac{1}{2}} - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)_{i+\frac{1}{2},j-\frac{1}{2}} \right], \quad (3.22)$$

onde as derivadas são calculadas como:

$$\left(\frac{\partial u}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta y}, \quad (3.23)$$

$$\left(\frac{\partial v}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x}. \quad (3.24)$$

As derivadas avaliadas no índice $(i + \frac{1}{2}, j - \frac{1}{2})$, em (3.22), são obtidas de forma análoga. Para o cálculo dos termos remanescentes em (3.19), executa-se os mesmos procedimentos descritos acima.

Figura 5 – Pontos necessários para o cálculo do termo viscoso avaliado em $(i + \frac{1}{2}, j)$, indicado por $\mathbf{x}$. Os triângulos representam as velocidades necessárias à discretização. Os pontos (vermelhos) indicam a posição das derivadas da velocidade intermediária.

Fonte: Elaborada pelo autor.

Sabe-se que, inicialmente, os valores das viscosidades são ordenados no centro das células. No entanto, podem ocorrer casos em que μ deverá ser estimada fora do centro de uma célula, à semelhança da equação (3.22) e ilustrado na figura 5, a título de exemplo. Em vista disso, assim como Sousa *et al.* (2004), os valores de μ nesses pontos foram estimados da seguinte maneira:

$$\mu_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{4}{\frac{1}{\mu_{i,j+1}} + \frac{1}{\mu_{i+1,j+1}} + \frac{1}{\mu_{i,j}} + \frac{1}{\mu_{i+1,j}}}. \quad (3.25)$$

3.3.4 Equação de Poisson para a pressão

Outro termo que merece atenção especial durante a resolução da equação de Navier-Stokes é a equação de Poisson para a pressão, definido pela equação (3.15), uma vez que a

densidade não é constante em todo o domínio. Em vista disso, a equação (3.15) foi discretizada da seguinte forma:

$$\left(\frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \right)_{i,j} \approx \frac{1}{\Delta t} \left(\frac{u_{i+\frac{1}{2},j}^* - u_{i-\frac{1}{2},j}^*}{\Delta x} + \frac{v_{i,j+\frac{1}{2}}^* - v_{i,j-\frac{1}{2}}^*}{\Delta y} \right) \quad (3.26)$$

e

$$\begin{aligned} \nabla \cdot \left(\frac{1}{\rho^n} \nabla \phi \right)_{i,j} &= \left[\frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{1}{\rho} \frac{\partial \phi}{\partial y} \right) \right]_{i,j} \\ &\approx \left\{ \frac{1}{2\Delta x} \left[\frac{1}{\rho_{i+1,j}} \left(\frac{\partial \phi}{\partial x} \right)_{i+1,j} - \frac{1}{\rho_{i-1,j}} \left(\frac{\partial \phi}{\partial x} \right)_{i-1,j} \right] \right. \\ &\quad \left. + \frac{1}{2\Delta y} \left[\frac{1}{\rho_{i,j+1}} \left(\frac{\partial \phi}{\partial y} \right)_{i,j+1} - \frac{1}{\rho_{i,j-1}} \left(\frac{\partial \phi}{\partial y} \right)_{i,j-1} \right] \right\}, \end{aligned} \quad (3.27)$$

em que

$$\left(\frac{\partial \phi}{\partial x} \right)_{i+1,j} \approx \frac{\phi_{i+2,j} - \phi_{i,j}}{2\Delta x}, \quad (3.28)$$

$$\left(\frac{\partial \phi}{\partial x} \right)_{i-1,j} \approx \frac{\phi_{i,j} - \phi_{i-2,j}}{2\Delta x}. \quad (3.29)$$

As derivadas na direção y são calculadas de forma análoga.

Figura 6 – Pontos necessários para o cálculo da equação de Poisson para pressão avaliada em (i, j) , indicada por $\mathbf{x}$. Os triângulos representam as velocidades necessárias à discretização. Os pontos (vermelhos) indicam a posição das derivadas da pressão.

Fonte: Elaborada pelo autor.

Como a velocidade intermediária é conhecida, neste passo, a equação (3.26) pode ser calculada. Por outro lado, de acordo com Kalland (2008), na equação (3.27) não é possível calcular, explicitamente, valores de p em um único ponto, sendo necessário a resolução de um sistema linear de equações, onde p é tratada, em cada ponto, como uma incógnita. Contudo, é necessária certa cautela durante a escolha dos métodos numéricos para a resolução desse sistema, visto que a variação de ρ de uma fase para outra torna essa resolução mais difícil, limitando a escolha dos métodos. Neste trabalho, assim como Lages (2016), utilizou-se o método BCGS (do inglês, *BiConjugate Gradient Squared*), porém, como pré-condicionador, fez-se o uso do método bloco iterativo de Jacobi, que é um dos resolvidores de sistema linear implementado na biblioteca PETSc, acoplada ao sistema HiG-Tree.

Por fim, para a discretização do termo advectivo, em (3.8), utilizou-se um método de diferenças centradas padrão.

TÉCNICA DA FRAÇÃO DO VOLUME DE FLUIDOS (VOF)

Neste capítulo, encontra-se o detalhamento do método VOF. Dividiu-se o capítulo da seguinte forma: Introdução teórica; Inicialização da fração de volume; Cálculo da curvatura e do vetor normal à interface, e Método PLIC-VOF. As técnicas que serão descritas neste capítulo foram implementadas para o caso 2D, desse modo, por conveniência, quando o termo *volume* for usado, subentende-se por *área*.

4.1 Introdução teórica

O volume de fluidos (VOF) (tradução direta do inglês, *Volume of Fluid*) é um método euleriano usado para localizar e reconstruir interfaces que separam fluidos imiscíveis presentes em um escoamento bifásico. Esse método foi originalmente introduzido por [Hirt e Nichols \(1981\)](#), sendo um dos mais aplicados para simulação desse tipo de escoamento, pois é relativamente simples de ser implementado, tem boa acurácia e apresenta bom comportamento no tratamento de mudanças substanciais na topologia da interface.

A ideia básica do método consiste em introduzir uma função indicadora, f , dentro de um volume de controle, representando, em cada passo de tempo, a fração de volume de fluido dentro de cada célula na malha computacional. Para exemplificar o funcionamento do método, considere um escoamento bifásico em uma interação gás-líquido, onde a análise é feita em coordenadas cartesianas. Dentro de cada célula computacional é definida a função $f \in [0, 1]$, tal que

$$f_{i,j} = \begin{cases} 0, & \text{se a célula (i,j) estiver cheia de líquido (Fluido 1),} \\ 1, & \text{se a célula (i,j) estiver cheia de gás (Fluido 2),} \\ f \in (0, 1), & \text{se a célula (i,j) contém, simultaneamente, líquido e gás.} \end{cases} \quad (4.1)$$

A representação geométrica da equação (4.1) pode ser observada na figura 7:

Figura 7 – Frações de volume de fluido em um volume de controle.

Fonte: Elaborada pelo autor.

Para a imposição da viscosidade e da densidade em cada célula computacional, as seguintes interpolações foram usadas:

$$\mu = (1 - f)\mu_1 + f\mu_2, \quad (4.2)$$

$$\rho = (1 - f)\rho_1 + f\rho_2, \quad (4.3)$$

com μ_1 , μ_2 e ρ_1 , ρ_2 representando, nesta ordem, as viscosidades e densidades de cada um dos fluidos.

A advecção das frações de volume de fluido, no método VOF, é modelada pela seguinte equação de transporte:

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}f) = 0. \quad (4.4)$$

É importante destacar que a precisão do método VOF depende vigorosamente da técnica escolhida para modelar a evolução da interface. Apesar da relativa simplicidade e baixo custo computacional das técnicas algébricas para resolver (4.4), uma vez que a interface não precisa ser reconstruída a cada passo de tempo, o método VOF apresenta maior eficiência com o uso de formulações geométricas, isso pode ser observado com mais detalhes no trabalho de [Rudman \(1997\)](#). Na seção 4.4, será descrita, de forma detalhada, a estratégia numérica escolhida para resolver essa equação.

Algumas vantagens do método VOF são:

- De acordo com [Hirt e Nichols \(1981\)](#), apresenta maior flexibilidade e eficiência que outros métodos para tratar interfaces com configurações complexas, fornecendo uma maneira simples e econômica de rastrear interfaces em formas bidimensionais ou tridimensionais.

- É possível escrever algoritmos precisos para advectar a fração de volume, de modo que a massa seja conservada enquanto ainda mantém uma representação nítida da interface (SUSSMAN; PUCKETT, 2000).
- De acordo com Scardovelli e Zaleski (1999), o tratamento de mudanças topológicas (coalescência ou quebra de interfaces) está implícito no algoritmo.
- De acordo com Scardovelli e Zaleski (1999), é um esquema local no sentido de que apenas os valores de f das células vizinhas são necessárias para atualizar f na célula (i, j) . Por esse motivo, é relativamente simples de ser implementado em paralelo.

4.2 Inicialização da fração de volume

O primeiro passo do método VOF é a inicialização da função f . Para uma superfície desejada que objetiva-se analisar, em um volume de controle, deve-se transformá-la em volumes fracionários para cada célula computacional, de forma que $f \in [0, 1]$. Destaca-se que uma má inicialização dessa função, conduzirá a uma estimativa imprecisa da curvatura e do vetor normal à interface, degradando a simulação numérica.

Neste trabalho, a inicialização de f foi feita baseando-se em funções implícitas, devido a forma fácil de verificar se um ponto está dentro ou fora da superfície. Uma função implícita, ϕ , de uma superfície é uma equação da forma $\phi(x, y) = 0$. Logo, para um dado ponto $\mathbf{P} = (x, y)$, basta verificar se este ponto está dentro ($\phi > 0$), fora ($\phi < 0$), ou na interface ($\phi = 0$).

4.2.1 Método de inicialização

Para a inicialização de f , tomou-se como base o método proposto por López *et al.* (2009). Esse método segue os seguintes passos:

1. Em um escoamento bifásico, considera-se um dos fluidos como referência, como ilustrado por Fig. 8.
2. Calcula-se a distância dos vértices de cada célula até a interface que separa os fluidos.
3. Se todos os vértices da célula têm distância com valores negativos (está fora do fluido de referência) ou valores positivos (está dentro do fluido de referência), a função indicadora, f , na célula é inicializada com 0 ou 1, respectivamente.
4. Para as células interfaciais (isto é, células que possuem vértices positivos e negativos, simultaneamente), a área de um polígono é calculada. Neste passo, a divisão da célula em subcélulas (nsc) menores aumenta a precisão do método de inicialização. Na seção 5.1.2, é possível notar a influência da divisão das células na precisão do método usado para calcular a curvatura.

Descrição dos passos de inicialização:

- *Passo 1* - Para exemplificar o procedimento, considere uma interface circular definida como $(x - x_0)^2 + (y - y_0)^2 = R^2$, sendo (x_0, y_0) o centro da circunferência, em que o fluido de referência (Fluido 2) encontra-se na parte interna à interface.

Figura 8 – Interface definida por $x^2 + y^2 = R^2$, em uma malha cartesiana, onde os pontos internos (azuis) representam o fluido de referência para a inicialização de f .

Fonte: Elaborada pelo autor.

- *Passo 2* - Seja $\mathbf{P}_i = (x_i, y_i)$ os vértices da célula computacional. Segue que a distância, ϕ_i , de cada vértice da célula até a interface é calculada analiticamente como: $\phi_i = R^2 - [(x_i - x_0)^2 + (y_i - y_0)^2]$.
- *Passos 3 e 4* - Para executar os passos 3 e 4, tomou-se como base o algoritmo Marching Squares (versão 2D do algoritmo Marching Cubes proposto por Lorensen e Cline (1987)). Este algoritmo é utilizado para localizar interfaces, com base no sinal da função distância armazenado nos vértices das células. Assim, conhecendo-se a posição relativa dos vértices, após a execução do *passo 2*, deve-se analisar todas as possibilidades de interseção entre a interface e a célula computacional, como ilustrado em Fig. 9. Os pontos de interseção gerados pelo encontro da interface com a célula são calculados por meio de uma interpolação linear. Se, por exemplo, a interface intersecta uma aresta definida pelos vértices P_1 e P_2 de uma célula, cujas distâncias ϕ_1 e ϕ_2 têm sinais opostos, então o ponto de interseção, I_P , é obtido da seguinte forma:

$$I_P = P_1 - \frac{\phi_1}{\phi_2 - \phi_1}(P_2 - P_1), \quad i = 1, 2. \quad (4.5)$$

Figura 9 – Possíveis interseções da interface com a célula computacional.

Fonte: Elaborada pelo autor.

O resultado das interseções da interface com as células formam polígonos convexos com vértices conhecidos. Portanto, para a obtenção das áreas desses polígonos, conhecendo-se as coordenadas de todos os seus vértices, fez-se o uso da seguinte expressão:

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right|, \quad (4.6)$$

onde n é o número de facetas e (x_i, y_i) , $i = 1, 2, \dots, n$, são os vértices do polígono. É importante observar que, para que o cálculo da área seja feito corretamente, os vértices devem ser dispostos no sentido horário ou anti-horário, exclusivamente, para todas as células.

Por fim, a fração de volume (neste caso, fração de área) de fluido da célula é obtida como:

$$f = \frac{A}{\Delta x \Delta y}. \quad (4.7)$$

4.3 Cálculo da curvatura e do vetor normal à interface

A curvatura e o vetor normal à interface são necessários para determinar a tensão interfacial. Para tanto, deve-se utilizar a informação da fração de volume para obtê-los. Algumas técnicas podem ser encontradas na literatura, tais como: Parker e Youngs (Parker e Youngs (1992)), Mínimos Quadrados (Scardovelli e Zaleski (2003)) e Função Altura (HF) (tradução direta do inglês, *Height Function*) (TORREY *et al.*, 1985). Neste trabalho, adotou-se o método

da Função Altura por sua implementação relativamente simples e por apresentar boa precisão. Tomou-se como base para a implementação desse método os trabalhos de [Popinet \(2009\)](#) e [Bornia et al. \(2011\)](#).

4.3.1 O método da Função Altura (HF)

Seja uma interface definida pela equação implícita $\phi(x,y)$. O método da função altura é uma técnica geométrica para estimar a curvatura interfacial e o vetor normal à interface, através da construção de funções explícitas locais $y = h(x)$ (ou, $x = h(y)$), chamadas de alturas, onde essas funções são construídas na mesma direção da maior componente do vetor normal à interface. Para determinar a direção da função altura sem utilizar a informação do vetor normal à interface, visto que inicialmente o vetor normal é desconhecido, pode-se aplicar a seguinte estratégia descrita por [Lam \(2009\)](#):

Constrói-se um stencil 3×3 entorno de cada célula interfacial, como ilustrado na figura 10.

Figura 10 – Stencil 3x3 entorno de uma célula interfacial.

Fonte: Elaborada pelo autor.

Posteriormente, soma-se as frações de volume das colunas à esquerda ($i-1$) e à direita ($i+1$) da célula interfacial, e calcula-se o valor absoluto da diferença entre $i-1$ e $i+1$, i.e.,

$$f_{i-1} = 1.0 + 1.0 + 0.62 = 2.62, \quad (4.8)$$

$$f_{i+1} = 0.0 + 0.04 + 0.01 = 0.05, \quad (4.9)$$

$$\Delta h_{horizontal} = |f_{i-1} - f_{i+1}| = 2.57. \quad (4.10)$$

De forma similar, calcula-se o valor absoluto da diferença entre $j - 1$ e $j + 1$, i.e.,

$$f_{j-1} = 1.0 + 0.8 + 0.01 = 1.81, \quad (4.11)$$

$$f_{j+1} = 0.62 + 0.34 + 0.0 = 0.96, \quad (4.12)$$

$$\Delta h_{vertical} = |f_{j-1} - f_{j+1}| = 0.85. \quad (4.13)$$

Logo, $\Delta h_{horizontal} > \Delta h_{vertical}$. Portanto, a função altura tem direção horizontal no exemplo ilustrado na figura 10.

Dado isso, considere uma malha uniforme de tamanho Δ . Sobre uma célula interfacial, na mesma direção da maior componente do vetor normal, $\mathbf{n} = (n_x, n_y)$, à interface, uma função h é localmente construída como uma aproximação da localização da interface e calculada a partir da soma das frações de volume, isto é,

$$h_j = \sum_{i=-N}^{i+N} f_{i,j} \Delta x \quad \text{para} \quad j = j' - 1, j', j' + 1, \quad (4.14)$$

para o caso em que a maior componente do vetor normal está na direção x (i.e., $|n_x| > |n_y|$), por exemplo.

Na literatura, por via de regra, o índice N , em (4.14), é tomado com valor $N = 3$, obtendo um stencil com tamanho fixo 7×3 (ou 3×7) entorno de cada célula interfacial, como ilustrado por Fig. 11:

Figura 11 – Stencil 3x7 entorno de uma célula interfacial, onde as alturas, h_j , são representadas pelas linhas tracejadas.

Fonte: Elaborada pelo autor.

No entanto, de acordo com Popinet (2009), em muitos casos não é necessário construir um stencil com sete células (i. e., $N = 3$) na direção da maior componente do vetor normal, enquanto que para outros casos necessita-se de stencils com mais de sete células para se ter uma estimativa consistente da altura da interface. Em vista disso, neste trabalho optou-se pela implementação do método da Função Altura com stencil variável, adaptado a geometria da interface. Nesse sentido, dada uma célula interfacial, (i, j) , aplica-se (4.14) para a obtenção

das funções altura, porém, como o método escolhido é dinâmico, cada altura será construída, independentemente, de modo que as linhas (ou colunas) do stencil tenham uma célula cheia ($f = 1$) em uma extremidade e uma célula vazia ($f = 0$) na extremidade oposta, e isso é usado como critério de parada no somatório em (4.14), obtendo um stencil não necessariamente simétrico, como ilustrado em Fig. 12:

Figura 12 – stencil adaptativo com funções de altura independentes.

Fonte: Elaborada pelo autor.

As funções altura devem ser definidas com respeito a um sistema de coordenadas fixo, logo, é necessário armazenar um valor auxiliar, $sgn(f) = \pm 1$, às alturas indicando o seu sentido (i. e., as alturas "crescem" da célula cheia para a célula vazia) e em seguida transladá-las para uma origem em comum, possibilitando o cálculo adequado das derivadas. Na figura 12, por exemplo, é necessário que a origem das alturas h_j e h_{j+1} estejam dispostas na mesma origem da altura h_{j-1} , para isso, acrescenta-se mais células às respectivas alturas, vide fig. 13:

Figura 13 – Stencil adaptativo com alturas, h_i , com origem em comum.

Fonte: Elaborada pelo autor.

Com isso, as alturas são usadas para estimar a curvatura (κ) e o vetor normal ($\mathbf{n}$) de cada

célula computacional:

$$\kappa = \text{sgn}(f) \cdot \frac{h_{yy}}{(1 + h_y^2)^{3/2}}, \quad (4.15)$$

$$\mathbf{n} = \begin{bmatrix} \text{sgn}(f) \\ -\text{sgn}(f) \cdot h_y \end{bmatrix}, \quad (4.16)$$

onde as derivadas h_y e h_{yy} são discretizadas utilizando diferenças finitas centradas de segunda ordem:

$$h_y = \frac{h_{i,j+1} - h_{i,j-1}}{2\Delta y}, \quad (4.17)$$

$$h_{yy} = \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{(\Delta y)^2}. \quad (4.18)$$

É importante destacar que podem existir células interfaciais isoladas onde o stencil, obtido a partir de (4.14), não dará uma boa estimativa da curvatura, mesmo no caso do stencil variável. Quando é construída uma função altura centrada em tais células isoladas, o critério de parada no somatório em (4.14) não é satisfeito, isto é, a função não é limitada por uma célula cheia em uma extremidade e uma célula vazia na extremidade oposta, impossibilitando a definição do sentido da função altura e, consequentemente, de uma origem em comum, como ilustrado por 14. Isso pode ser visto com mais detalhes no trabalho de [Bornia et al. \(2011\)](#). Assim, para tratar esses casos, o cálculo da curvatura nessas células deve ser feito de uma forma alternativa. No trabalho de [Bornia et al. \(2011\)](#), por exemplo, a estimativa da curvatura nessas células é feita pelo método de [Parker e Youngs \(1992\)](#), já [Popinet \(2009\)](#) utiliza um método baseado no ajuste da curvatura à parábola.

Figura 14 – Stencil com célula interfacial isolada.

Fonte: Elaborada pelo autor.

Neste trabalho, para dar uma estimativa consistente da curvatura interfacial nessas células, optou-se pelo método da função altura modificado (descentrado), como ilustrado por Fig.15:

Figura 15 – Stencil descentrado com célula interfacial isolada.

Fonte: Elaborada pelo autor.

obtendo, assim, as seguintes fórmulas para o cálculo da curvatura:

$$h_y = -\text{sgn}(f) \cdot \frac{1.5h_i - 2h_{i+1} + 0.5h_{i+2}}{\Delta y}, \quad (4.19)$$

$$h_{yy} = -\text{sgn}(f) \cdot \frac{h_i - 2h_{i+1} + h_{i+2}}{(\Delta y)^2}. \quad (4.20)$$

Com as formulações apresentadas acima, a curvatura é definida no centro das células interfaciais. Entretanto, a representação das forças de tensão interfacial, descrita com mais detalhes na seção 4.5, requer a curvatura centrada nas facetas das células. Desse modo, no trabalho de [Figueiredo \(2016\)](#), por exemplo, a curvatura é estimada nas facetas aplicando-se a seguinte média ponderada:

$$\kappa_{i+\frac{1}{2},j} = \frac{w_{i,j}\kappa_{i,j} + w_{i+1,j}\kappa_{i+1,j}}{w_{i,j} + w_{i+1,j}}, \quad (4.21)$$

onde o fator de ponderação é definido como $w_{i,j} = f_{i,j}(1 - f_{i,j})$.

4.4 Método PLIC-VOF

Com o objetivo de resolver a equação (4.4), para o transporte da interface durante a dinâmica do escoamento bifásico, optou-se pela implementação do método geométrico PLIC (do inglês, *Piecewise-Linear Interface Calculation*) do VOF, em que a interface é aproximada por um segmento de reta em cada célula interfacial. Esse método foi introduzido por [Youngs \(1982\)](#), sendo um dos mais utilizados na literatura. Tomou-se como base principalmente os trabalhos de [Puckett et al. \(1997\)](#) e [Figueiredo \(2016\)](#).

O esquema PLIC é dividido em duas etapas:

1. Reconstrução da interface;
2. Advecção da interface.

4.4.1 Reconstrução da interface

No passo de reconstrução no método PLIC-VOF, dada a fração de volume de fluido, f , e uma estimativa do vetor normal à interface, $\mathbf{n}$, em cada célula computacional, obtém-se unicamente a interface linear aproximada naquela célula, de modo que se tenha o volume de fluido apropriado. Essa interface linear é representada por um segmento de reta descrito por:

$$m_x x + m_y y = \alpha, \quad (4.22)$$

em que $m_k = \frac{n_k}{\|\mathbf{n}\|_2}$, $k = x, y$, com $\|\cdot\|_2$ representando a norma euclidiana, x e y é a posição de algum ponto sobre o segmento de reta e α é a menor distância do segmento à origem da célula.

Como f é conhecida inicialmente, visto que nesta etapa o passo de inicialização de f (vide seção 4.2) já foi aplicado, o problema, então, consiste em estimar $\mathbf{n}$ para se obter a interface linear. Na literatura, existem diversos algoritmos de reconstrução que dão uma estimativa para o vetor normal à interface, baseados nas aproximações de diferenças finitas do gradiente da fração de volume, ∇F , satisfazendo, em geral, algum critério de minimização. Como exemplo, destacam-se os métodos: Parker and Youngs de [Parker e Youngs \(1992\)](#), Least Squares Volume-of-Fluid Interface Reconstruction Algorithm (LVIRA) de [Puckett \(1991\)](#) e Efficient Least Squares VOF Interface Reconstruction Algorithm (ELVIRA) de [Pilliod \(1992\)](#). Porém, neste trabalho, fez-se o uso do método da Função Altura, descrito na seção 4.3, para estimar $\mathbf{n}$, uma vez que esse método apresenta eficiência computacional, possui implementação relativamente simples e assegura uma estimativa consistente para $\mathbf{n}$.

Tendo em vista que f é definida pela equação (4.7), então a área de fluido da fase de referência em cada célula é obtido por:

$$A = f \cdot \Delta x \Delta y. \quad (4.23)$$

Uma vez que f , a área total da célula, a área da fase de referência e o vetor normal são conhecidos inicialmente, o problema do método PLIC-VOF consiste em ajustar a interface linear reconstruída, visando determinar a área, A_{plic} , abaixo dessa interface, tal que A_{plic} seja igual a A . Em contra partida, deve-se efetuar também o cálculo inverso, isto é, dado que o vetor normal e a área da fase de referência são conhecidos, deve-se calcular a distância da interface à origem da célula. As relações acima podem ser sumarizadas da seguinte forma:

$$\begin{aligned} A_{plic} &= A_{plic}(\mathbf{m}, \alpha), \\ \alpha &= \alpha(\mathbf{m}, A). \end{aligned}$$

De acordo com [Scardovelli e Zaleski \(2000\)](#), dependendo do método numérico escolhido, essas relações podem ocorrer muitas vezes durante a reconstrução e propagação da interface.

4.4.1.1 Expressão para o cálculo da área abaixo da interface reconstruída

Considere uma célula computacional com dimensões $\Delta = (\Delta x, \Delta y)$ e o vetor normal à interface apontando para dentro da região onde o fluido de referência está localizado, ou seja, o fluido de referência encontra-se a esquerda da interface linear. O valor de A_{plic} , dentro de qualquer célula, (i, j) , pode ser obtido pela seguinte expressão:

$$\begin{aligned}
 2 \cdot A_{plic} &= \frac{|d|^2}{|n_x| \cdot |n_y|} \left[1 - R_x \cdot \left(1 - \frac{|n_x| \Delta x}{|d|} \right)^2 - R_y \cdot \left(1 - \frac{|n_y| \Delta y}{|d|} \right)^2 \right] \\
 &= \frac{|d|^2}{|n_x| \cdot |n_y|} \left[1 - R_x \cdot \left(1 - \frac{2|n_x| \Delta x}{|d|} + \frac{|n_x|^2 \Delta x^2}{|d|^2} \right) - R_y \cdot \left(1 - \frac{2|n_y| \Delta y}{|d|} + \frac{|n_y|^2 \Delta y^2}{|d|^2} \right) \right] \\
 &= |d|^2 \left(\frac{1 - R_x - R_y}{|n_x| \cdot |n_y|} \right) + |d| \left(\frac{2R_x \Delta x}{|n_y|} + \frac{2R_y \Delta y}{|n_x|} \right) - \frac{R_x |n_x| \Delta x^2}{|n_y|} - \frac{R_y |n_y| \Delta y^2}{|n_x|}, \quad (4.24)
 \end{aligned}$$

onde

$$R_x = R(|d| - |n_x| \Delta x), \quad (4.25)$$

$$R_y = R(|d| - |n_y| \Delta y), \quad (4.26)$$

e a função $R(x)$ é definida como:

$$R(x) = \begin{cases} 0, & \text{se } x \leq 0 \\ 1, & \text{se } x > 0. \end{cases} \quad (4.27)$$

Na equação (4.24), o parâmetro d representa a distância da interface linear à origem da célula. No entanto, para a aplicação da fórmula, a origem deve ser tomada nos vértices da célula. Logo, d e α são iguais a menos de uma translação. Ademais, o termo $[d^2 / (2 \cdot |n_x| \cdot |n_y|)]$ representa a área do polígono, AEH , abaixo da interface linear, como ilustrado na fig. 16:

Figura 16 – Área sob a reta geral cortando a célula computacional.

Fonte: Elaborada pelo autor.

O primeiro e o segundo termo subtraídos do lado direito da primeira igualdade em (4.24), representam as áreas DGH e BEF , respectivamente, sob a interface linear fora da célula computacional, como ilustrado na fig. 17:

Figura 17 – Áreas fora da célula computacional que devem ser subtraídas na eq.(4.24).

Fonte: Elaborada pelo autor.

A função $R(x)$ garante que esses termos são algebricamente subtraídos apenas se os vértices E e H estiverem fora da célula computacional.

4.4.1.2 Descrição do algoritmo de reconstrução da interface

Suponha, neste momento, que f e $\mathbf{n}$ são conhecidos. Assim, o passo seguinte para a reconstrução da interface consiste em calcular os argumentos de (4.25) e (4.26), determinar o valor de $R(x)$ em (4.27) e, por fim, estimar a distância, d , da interface à origem da célula.

Para exemplificar o funcionamento do método, considere, sem perda de generalidade, $n_x \cdot n_y > 0$, sendo $n_x, n_y < 0$. Com o intuito de avaliar (4.25) e (4.26), deve-se tomar duas retas auxiliares passando pelos vértices superior esquerdo (LT) e inferior direito (RB), respectivamente, e a origem de (i, j) no vértice inferior esquerdo (LB), como ilustrado por fig. 18:

- $R_x = 0$ e $R_y = 0$

Figura 18 – Representação geométrica da interface PLIC quando $R_x = 0$ e $R_y = 0$.

Fonte: Elaborada pelo autor.

De (4.25) e (4.26), têm-se:

$$arg_x = d - |n_x|\Delta x = |d| - dRB < 0 \Rightarrow R_x = 0, \quad (4.28)$$

$$arg_y = d - |n_y|\Delta y = |d| - dLT < 0 \Rightarrow R_y = 0. \quad (4.29)$$

Logo, a partir de (4.24), pode-se estimar a distância da interface à origem pela seguinte expressão:

$$d = |\sqrt{2 \cdot A_{PLIC} \cdot |n_x| |n_y|}|. \quad (4.30)$$

- $R_x = 0$ e $R_y = 1$

Figura 19 – Representação geométrica da interface PLIC quando $R_x = 0$ e $R_y = 1$.

Fonte: Elaborada pelo autor.

De (4.25) e (4.26), têm-se:

$$arg_x = d - |n_x|\Delta x = d - dRB < 0 \Rightarrow R_x = 0, \quad (4.31)$$

$$arg_y = d - |n_y|\Delta y = d - dLT > 0 \Rightarrow R_y = 1. \quad (4.32)$$

Logo, a partir de (4.24), pode-se estimar a distância da interface à origem pela seguinte expressão:

$$d = \frac{2A_{PLIC}|n_x| + R_y|n_y|\Delta y^2}{2R_y\Delta y}. \quad (4.33)$$

- $R_x = 1$ e $R_y = 1$

Figura 20 – Representação geométrica da interface PLIC quando $R_x = 1$ e $R_y = 1$.

Fonte: Elaborada pelo autor.

De (4.25) e (4.26), têm-se:

$$arg_x = d - |n_x|\Delta x = d - dRB > 0 \Rightarrow R_x = 1, \quad (4.34)$$

$$arg_y = d - |n_y|\Delta y = d - dLT > 0 \Rightarrow R_y = 1. \quad (4.35)$$

Logo, a partir de (4.24), pode-se estimar a distância da interface à origem pela seguinte expressão:

$$B = \left(\frac{2R_x\Delta x}{|n_y|} + \frac{2R_y\Delta y}{|n_x|} \right)^2 - 4 \left(\frac{1 - R_x - R_y}{|n_x| \cdot |n_y|} \right) \cdot \left(-\frac{R_x|n_x|\Delta x^2}{|n_y|} - \frac{R_y|n_y|\Delta y^2}{|n_x|} - 2A_{PLIC} \right). \quad (4.36)$$

$$d = \min \left[\frac{-\left(\frac{2R_x\Delta x}{|n_y|} + \frac{2R_y\Delta y}{|n_x|} \right) \pm \sqrt{B}}{2 \left(\frac{1 - R_x - R_y}{|n_x| \cdot |n_y|} \right)} \right]. \quad (4.37)$$

Para os casos em que $n_x, n_y > 0$, os cálculos seguem de forma análoga aos apresentados acima, mas agora a área é definida como $(\Delta x \Delta y - A_{PLIC})$. Além disso, se $n_x \cdot n_y < 0$, os cálculos também são feitos de forma semelhante aos apresentados acima, mas agora considera-se duas retas auxiliares passando pelos vértices inferior esquerdo (LB) e superior direito (RT), respectivamente, e origem no vértice inferior direito (RB).

Também é possível que ocorram casos em que uma das coordenadas de $\mathbf{n}$ seja nula. Geometricamente, isso diz que a interface linear deve ser paralela a uma das facetas da célula, vide fig. 21:

Figura 21 – Representação geométrica da interface PLIC quando $\mathbf{n} = (0, -n_y)$.

Fonte: Elaborada pelo autor.

Na figura 21, $\mathbf{n} = (0, -n_y)$. Desse modo, a área da célula é calculada trivialmente como:

$$A_{plic} = \Delta x \cdot d. \quad (4.38)$$

4.4.2 Advecção da interface

Uma vez que a interface foi reconstruída em todas as células interfaciais, seu movimento deve ser modelado por um algoritmo de advecção que resolva a equação (4.4) adequadamente. Essa equação pode ser resolvida com o uso de diferentes categorias de esquemas numéricos, os quais são: algébricos e geométricos. Como dito na seção 4.1, o método VOF não apresenta bom comportamento quando acoplado a esquemas algébricos, gerando resultados não físicos na vizinhança da interface. De acordo com Ivey e Moin (2012), por exemplo, o esquema algébrico Upwind de primeira ordem conserva globalmente F e a mantém limitada, porém, ocorre a rápida difusão da interface. Já esquemas Upwind de alta ordem ajudam a manter a integridade da interface, no entanto, não mantêm F limitada, culminando na perda de precisão do VOF. A fim de minimizar esses problemas, neste trabalho, a advecção é feita através do fluxo geométrico nas facetas das células. Para determinar o fluxo através da faceta em que o campo de velocidade está direcionado, deve-se utilizar informações geométricas obtidas no passo de reconstrução do PLIC-VOF. De acordo com Figueiredo (2016), a acurácia na resolução da equação (4.4) está diretamente ligada à precisão do método de reconstrução. No presente projeto, o passo de reconstrução foi feito como descrito na seção 4.4.1. A verificação do acoplamento entre as técnicas de reconstrução e advecção adotadas será apresentada na seção 5.

No trabalho de Puckett *et al.* (1997), são apresentados dois métodos geométricos para a

atualização da fração de volume em 4.4, estes são: Algoritmo de advecção com passo fracionário (*split*) e Algoritmo de advecção de segunda ordem com passo único (*unsplit*). No algoritmo *split*, a advecção é feita inicialmente na direção horizontal e, posteriormente, na direção vertical. Essa estratégia torna o método mais simples e evita que uma mesma porção de fluido seja advectada duas vezes, como ocorre nos métodos de passo único (FIGUEIREDO, 2016). De acordo com Puckett *et al.* (1997), alternando-se a direção de advecção em cada passo de tempo, esse algoritmo atinge segunda ordem de precisão no tempo. Em vista disso, neste trabalho, optou-se pela implementação do método *split*.

Assim sendo, discretizando a equação (4.4) com diferenças finitas centradas no espaço e progressiva no tempo, tem-se:

$$\begin{aligned} f_{i,j}^{n+1} &= f_{i,j}^n - \frac{\Delta t}{\Delta x} [(fu)_{i+1/2,j}^n - (fu)_{i-1/2,j}^n] - \frac{\Delta t}{\Delta y} [(fv)_{i,j+1/2}^n - (fv)_{i,j-1/2}^n] \\ &= f_{i,j}^n - \frac{\Delta t}{\Delta x} [F_{i+1/2,j} - F_{i-1/2,j}] - \frac{\Delta t}{\Delta y} [G_{i,j+1/2} - G_{i,j-1/2}], \end{aligned} \quad (4.39)$$

onde $f_{i,j}^n$ é o valor da fração de volume na (i, j) – ésima célula, no tempo $t^n = n\Delta t$, u e v são as componentes horizontal e vertical da velocidade, respectivamente, e $F_{i+1/2,j}$ denota o fluxo de f na faceta direita de (i, j) .

Desde que no método *split* o passo de advecção é dividido em duas etapas para a resolução de (4.4), deve-se, inicialmente, calcular uma fração de volume intermediária, $f_{i,j}^*$, para finalmente calcular a fração de volume final, $f_{i,j}^{n+1}$. Logo, têm-se:

$$f_{i,j}^* = f_{i,j}^n - \frac{\Delta t}{\Delta x} (F_{i+1/2,j}^n - F_{i-1/2,j}^n), \quad (4.40)$$

$$f_{i,j}^{n+1} = f_{i,j}^* - \frac{\Delta t}{\Delta y} (G_{i,j+1/2}^n - G_{i,j-1/2}^n). \quad (4.41)$$

Suponha, sem perda de generalidade, que a velocidade $u_{i+1/2,j}$ é positiva. Assim, a célula (i, j) pode ser dividida em dois retângulos, cujas áreas direita e esquerda são definidas como $u_{i+1/2,j}\Delta t\Delta y$ e $(\Delta x - u_{i+1/2,j}\Delta t)\Delta y$, respectivamente. Durante o passo de tempo Δt , todo a área de fluido contido no retângulo direito (especificamente, a área escura, $\bar{A}_{i+1/2,j}$, abaixo da interface linear, vide figura 22) passará pela faceta à direita da célula.

Figura 22 – Fração de área de fluido (região escura) a ser advectada para a célula vizinha (direita).

Fonte: Elaborada pelo autor.

A área $\bar{A}_{i+1/2,j}$ pode ser calculada usando uma aproximação análoga à apresentada na seção anterior (vide seção 4.4.1). Para isto, deve-se tomar as seguintes informações geométricas: vetor normal à interface, comprimentos $u_{i+1/2,j}\Delta t$ e Δy das facetas do retângulo, e distância local da interface à origem. Com isso, o fluxo de fluido, $F_{i+1/2,j}^n$, é dado por:

$$F_{i+1/2,j}^n = \frac{u_{i+1/2,j}^n \cdot \bar{A}_{i+1/2,j}}{u_{i+1/2,j}^n \cdot \Delta t \cdot \Delta y} = \frac{\bar{A}_{i+1/2,j}}{\Delta t \cdot \Delta y}. \quad (4.42)$$

Feito isso, a fração de volume intermediária, $f_{i,j}^*$, pode ser calculada via equação (4.40). Em seguida, antes do cálculo de $f_{i,j}^{n+1}$, uma nova interface deverá ser reconstruída. Finalmente, o fluxo vertical, $G_{i,j+1/2}^n$ pode ser obtido com o procedimento análogo ao utilizado para a obtenção do fluxo horizontal, e, assim, a fração de volume no tempo t^{n+1} pode ser estimada via equação (4.41).

Seguindo a estratégia proposta por Puckett *et al.* (1997), a fim de manter a conservação de f , levou-se em consideração a condição de incompressibilidade (i.e., $\nabla \cdot \mathbf{u} = 0$), gerando a seguinte modificação na equação (4.4):

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}f) = f \nabla \cdot \mathbf{u}, \quad (4.43)$$

ou ainda

$$\frac{\partial f}{\partial t} + \nabla \cdot (uf) = f \nabla \cdot \mathbf{u} \quad (4.44)$$

$$\frac{\partial f}{\partial t} + \nabla \cdot (vf) = f \nabla \cdot \mathbf{v}. \quad (4.45)$$

Rider e Kothe (1998) mostram que, com a adição de $f \nabla \cdot \mathbf{u}$ na equação (4.4), as restrições de volume locais e globais são respeitadas muito mais de perto, tornando a conservação de f mais precisa.

Assim, discretizando f implicitamente à direita de (4.44) e explicitamente à direita de (4.45), têm-se:

$$f_{i,j}^* = \frac{f_{i,j}^n + \frac{\Delta t}{\Delta x}(F_{i-1/2,j}^n - F_{i+1/2,j}^n)}{1 - \frac{\Delta t}{\Delta x}(u_{i+1/2,j} - u_{i-1/2,j})} \quad (4.46)$$

$$f_{i,j}^{n+1} = f_{i,j}^* + \frac{\Delta t}{\Delta y}[f_{i,j}^*(v_{i,j+1/2}^n - v_{i,j-1/2}^n) - (G_{i,j+1/2}^n - G_{i,j-1/2}^n)]. \quad (4.47)$$

4.4.2.1 Restrição para o passo de tempo

Durante o passo de advecção da interface com o PLIC-VOF, deve-se considerar algumas restrições para que o método funcione satisfatoriamente. Assim como Kalland (2008), para que o método seja estável numericamente, as seguintes condições devem ser consideradas:

- A área que deverá ser advectada não pode ser maior do que uma célula.
- Se os vetores velocidade de duas facetas opostas de uma célula tiverem sentidos diferentes, então as duas áreas de uma célula que deverão ser advectadas não podem apresentar sobreposição (vide figura 23).

Figura 23 – Sobreposição de duas regiões de uma célula que serão advectadas em sentidos opostos.

Fonte: Elaborada pelo autor.

Neste trabalho, durante a implementação do método, para que as duas condições fossem satisfeitas simultaneamente, a seguinte condição foi imposta:

$$|\Delta t \cdot u| < \frac{\Delta x}{2} \quad (4.48)$$

$$|\Delta t \cdot v| < \frac{\Delta y}{2}. \quad (4.49)$$

De acordo com Kalland (2008), pode-se usar a seguinte restrição para o passo de tempo:

$$\Delta t < \min\left(\frac{\Delta x}{2|u|_{\max}}, \frac{\Delta y}{2|v|_{\max}}\right) \equiv \Delta t_{advec}. \quad (4.50)$$

4.5 Implementação da tensão interfacial

Optando-se por modelar escoamentos bifásicos por um único conjunto de equações, além de considerar os termos viscosidade e densidade variáveis de fase para fase, há de se levar em conta a força que age sobre a interface entre os fluidos presentes no escoamento, comumente conhecida como força de tensão interfacial. Desse modo, acrescenta-se o termo da tensão interfacial como um termo fonte à equação do momentum (3.2).

Dado isso, no presente trabalho, adotou-se o método CSF (do inglês, *Continuum Surface Force*), de Brackbill, Kothe e Zemach (1992), para modelar a força de tensão interfacial. Nesse modelo, a força de tensão por unidade de área interfacial pode ser expressa como:

$$F_\sigma = \sigma \kappa \mathbf{n} \delta. \quad (4.51)$$

No contexto do método VOF, assim como feito nos trabalhos de Popinet (2009) e Figueiredo (2016), a equação (4.51) pode ser rescrita como:

$$F_\sigma = \sigma \kappa \frac{\nabla f}{\|\nabla f\|} \|\nabla f\| = \sigma \kappa \nabla f. \quad (4.52)$$

Assim, F_σ é não nula apenas em células onde o gradiente da fração de volume, ∇f , é não nulo.

Como dito no capítulo 3, empregou-se um esquema de malha deslocada (vide figura 3). Com isso, ∇f é avaliada no centro das células e, conseqüentemente, F_σ também é. Assim, o termo interfacial presente na equação do momentum, (3.2), deverá ser interpolado para as facetas. Logo, define-se F_σ nas facetas discretizando (4.52) da seguinte forma:

$$F_{\sigma x} = \sigma \kappa_{i-\frac{1}{2},j} \frac{f_{i,j} - f_{i-1,j}}{\Delta x} \quad \text{e} \quad F_{\sigma y} = \sigma \kappa_{i,j-\frac{1}{2}} \frac{f_{i,j} - f_{i,j-1}}{\Delta y}, \quad (4.53)$$

onde a curvatura na faceta é aproximada como descrito na equação (4.21).

A inclusão do termo de tensão interfacial na equação de Navier-Stokes demanda mais uma restrição para o passo de tempo. De acordo com Popinet (2018), a discretização explícita do termo de tensão interfacial conduz para uma condição de estabilidade temporal da forma:

$$\Delta t < \sqrt{\frac{(\rho_0 + \rho_1) \Delta^3}{4\pi\sigma}} \equiv \Delta t_\sigma. \quad (4.54)$$

Como justificativa física para a restrição, Popinet (2018) afirma que o passo de tempo deve ser suficientemente pequeno para resolver a propagação da onda de capilaridade mais rápida do sistema. Ondas de capilaridade são pequenas ondas onde a dinâmica é dominada pela tensão interfacial (KALLAND, 2008).

Tomando a razão entre as condições de estabilidade (4.50) e (4.54), obtém-se:

$$\frac{\Delta t_\sigma}{\Delta t_{advec}} = \sqrt{\frac{(\rho_0 + \rho_1) |\mathbf{u}|^2 \Delta^2}{\pi\sigma}}. \quad (4.55)$$

RESULTADOS NUMÉRICOS

Neste capítulo, encontram-se os resultados numéricos alcançados com este trabalho. Neste cenário, tomou-se como estratégia a apresentação dos resultados em duas etapas. Na primeira etapa, faz-se a verificação dos métodos e técnicas acoplados ao método VOF, e, na segunda etapa, são apresentados testes clássicos da literatura envolvendo as simulações numéricas de escoamentos bifásicos em um canal.

1. Testes de verificação dos métodos implementados:

- Análise do método de inicialização;
- Análise do cálculo da curvatura e do vetor normal à interface;
- Análise do algoritmo de reconstrução e advecção PLIC-VOF.

2. Simulações numéricas de escoamentos bifásicos:

- Gota estática;
- Gota oscilante;
- Bolha ascendente.

5.1 Testes de verificação dos métodos

Para as análises de convergência dos métodos empregados, considerou-se um círculo dado pela equação implícita:

$$\phi(x,y) = (x - x_0)^2 + (y - y_0)^2 - R^2 = 0, \quad (5.1)$$

com raio $R = 0.25$, centrado em $(x_0, y_0) = (0.5, 0.5)$, em um domínio $\Omega = [0, 1] \times [0, 1]$, como ilustrado na figura 24:

Figura 24 – Configuração inicial.

Fonte: Elaborada pelo autor.

5.1.1 Análise do método de inicialização de f

Nesta seção, é apresentado um estudo de convergência do método de inicialização descrito na seção 4.2. Como dito previamente, a técnica de inicialização implementada neste trabalho foi baseada no método proposto por [López et al. \(2009\)](#), desse modo, espera-se que esse método atinja segunda ordem de precisão.

À semelhança de [Ribera et al. \(2015\)](#), para analisar o método de inicialização, comparou-se a área exata da superfície com a soma das áreas inicializadas em cada célula computacional, obtidas a partir das informações de f . Para isso, as seguintes expressões foram consideradas:

$$\text{Área}_{\text{exata}} = \pi \times r^2, \quad (5.2)$$

$$\text{Área}_{\text{inicializada}} = \sum (F_{i,j} \times A_{i,j}), \quad (5.3)$$

onde $F_{i,j}$ e $A_{i,j}$ representam a fração de área e a área de cada célula (i, j) , respectivamente.

O erro de inicialização (ε) e a ordem de convergência (p) são calculados como:

$$\varepsilon(\Delta_m) = \frac{\max(|\text{Área}_{\text{inicializada}} - \text{Área}_{\text{exata}}|)}{\text{Área}_{\text{exata}}}, \quad (5.4)$$

$$p \approx \frac{\log(\varepsilon(\Delta_m)/\varepsilon(\Delta_{m+1}))}{\log(\Delta_m/\Delta_{m+1})}, \quad (5.5)$$

nessa ordem, sendo Δ_m o espaçamento da malha m .

Para a inicialização da interface, definida pela equação implícita (5.1), no método VOF, as células interfaciais foram uniformemente divididas em 8^2 subcélulas. Como descrito na tabela 1 e ilustrado na figura 25, o método de inicialização implementado apresenta segunda ordem de convergência para essa interface, estando em boa concordância com os resultados apresentados por [López et al. \(2009\)](#).

Tabela 1 – Análise de convergência do erro oriundo da inicialização de f , com diferentes tamanhos de malha, na norma L_∞ .

Malha	Passo (Δ)	Erro (ϵ)	Razão	Ordem (p)
32x32	0.03125	2.00029e-05	-	-
64x64	0.01562	5.01253e-06	3.99058	1.99660
128x128	0.00781	1.26866e-06	3.95104	1.98223
256x256	0.00391	3.17152e-07	4.00017	2.00006

Fonte: Dados da pesquisa.

Figura 25 – Erros de inicialização da fração de volume, f , em função da resolução da malha, dividindo cada célula interfacial em 8^2 subcélulas.

Fonte: Elaborada pelo autor.

5.1.2 Análise do cálculo da curvatura e do vetor normal à interface

Análise da curvatura interfacial

Para analisar o método utilizado para estimar a curvatura, comparou-se a curvatura exata do círculo (curvatura média) com a curvatura obtida a partir do método da função altura (HF), descrito na seção 4.3. A curvatura exata é obtida a partir da seguinte expressão:

$$\kappa_{exata} = 1/r. \quad (5.6)$$

O erro (ϵ) oriundo da estimativa da curvatura é calculado da seguinte forma:

$$\epsilon(\Delta) = \frac{\max(|\kappa_{aproximada} - \kappa_{exata}|)}{\kappa_{exata}}, \quad (5.7)$$

a ordem de convergência é calculada usando a expressão 5.5.

A razão de convergência do erro no cálculo da curvatura, via método HF, é apresentada nas tabelas 2 e 3. Na tabela 2, os resultados foram obtidos a partir do cálculo da curvatura, tomando-se informações do campo de fração de volume inicializado dividindo-se cada célula interfacial por 8^2 subcélulas. Pode-se notar que a ordem de convergência do método oscila com o refinamento da malha, como ilustrado na figura 26:

Tabela 2 – Análise de convergência do erro oriundo do cálculo da curvatura, via método HF, para diferentes resoluções de malha, com $n_{sc} = 8^2$.

Malha	Passo (Δ)	Erro (ϵ)	Razão	Ordem (p)
32x32	0.03125	1.25195e-02	-	-
64x64	0.01562	6.03125e-03	2.07586	1.05371
128x128	0.00781	3.18600e-03	1.89297	0.92065
256x256	0.00391	7.30000e-04	4.36438	2.12578
512x512	0.00195	3.28150e-04	2.22490	1.15374

Fonte: Dados da pesquisa.

Figura 26 – Erros na estimativa da curvatura interfacial, em função da resolução da malha, usando o método HF, com $n_{sc} = 8^2$.

Fonte: Elaborada pelo autor.

Agora, dividindo cada célula interfacial por 64^2 subcélulas na inicialização do campo de fração de volume, obtêm-se resultados mais precisos no cálculo da curvatura, isso pode ser observado na tabela 3 e na figura 27:

Tabela 3 – Análise de convergência do erro oriundo do cálculo da curvatura, via método HF, para diferentes resoluções de malha, com $n_{sc} = 64^2$.

Malha	Passo (Δ)	Erro (ϵ)	Razão	Ordem (p)
32x32	0.03125	1.25082e-02	-	-
64x64	0.01562	6.02725e-03	2.07528	1.05331
128x128	0.00781	3.18550e-03	1.89209	0.91998
256x256	0.00391	7.30500e-04	4.36071	2.12456
512x512	0.00195	1.62750e-04	4.48848	2.16623

Fonte: Dados da pesquisa.

Figura 27 – Erros na estimativa da curvatura interfacial, em função da resolução da malha, via método HF, com $n_{sc} = 64^2$.

Fonte: Elaborada pelo autor.

Esses resultados corroboram a grande influência do método de inicialização do VOF na precisão do cálculo da curvatura interfacial. De acordo com [Popinet \(2009\)](#), o método padrão da função altura atinge segunda ordem de precisão na estimativa da curvatura quando a fração de volume é inicializada exatamente.

Análise do vetor normal à interface

A análise do vetor normal à interface definida pela equação implícita (5.1) é feita comparando-se o vetor normal obtido numericamente, oriundo do método da função altura, com o vetor normal teórico, dado pela seguinte expressão:

$$\mathbf{n}_t = \frac{(x - x_0, y - y_0)}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}. \quad (5.8)$$

Suponha, sem perda de generalidade, que o estêncil usado no método da função de altura seja vertical (i. e., o vetor normal à interface tem direção vertical). Logo, para calcular (5.8), o vetor normal teórico deve ser avaliado no ponto $P_t = (x_c - x_0, \pm \sqrt{R^2 - (x_c - x_0)^2})$, onde x_0 representa a coordenada x do centro da interface circular e x_c é a coordenada x do centro de uma célula interfacial. Se P_t estiver localizado no semicírculo superior, então $P_t = (x_c - x_0, \sqrt{R^2 - (x_c - x_0)^2})$, senão $P_t = (x_c - x_0, -\sqrt{R^2 - (x_c - x_0)^2})$.

Dado o vetor normal teórico e o vetor normal aproximado, o erro é definido como:

$$\varepsilon(\Delta) = \max(|\mathbf{n} - \mathbf{n}_t|), \quad (5.9)$$

e, de forma análoga a análise da curvatura, a ordem de convergência é calculada usando a expressão 5.5. Os resultados podem ser observados na tabela 4 e na figura 28 abaixo:

Tabela 4 – Análise de convergência do erro oriundo do cálculo do vetor normal à interface, via método HF, para diferentes resoluções de malha, com $n_{sc} = 8^2$.

Malha	Passo (Δ)	Erro (ε)	Razão	Ordem (p)
32x32	0.03125	1.39330e-02	-	-
64x64	0.01562	1.24150e-02	1.12227	0.16642
128x128	0.00781	9.80830e-03	1.26576	0.34001
256x256	0.00391	2.71090e-03	3.61810	1.85523
512x512	0.00195	6.08700e-04	4.45359	2.15497

Fonte: Dados da pesquisa.

Figura 28 – Erros na estimativa do vetor normal à interface, em função da resolução da malha, usando o método HF, com $n_{sc} = 8^2$.

Fonte: Elaborada pelo autor.

5.1.3 Análise do algoritmo de reconstrução e advecção PLIC-VOF

Para finalizar a seção de testes de verificação dos métodos implementados, apresenta-se um teste onde é possível avaliar, além da correta implementação do método para o cálculo do vetor normal, o método de reconstrução e advecção da interface, PLIC-VOF 4.4.

Após a inicialização da interface circular (5.1), foi prescrito um campo de velocidade para advecção-la em um domínio retangular $([0, 2] \times [0, 1])$.

Seja $\Gamma_0^{num} = [x(s, 0), y(s, 0)]$ a interface numérica no tempo $t = 0$, e o campo de velocidade $\mathbf{u}(x, y)$ dado por:

$$\begin{aligned} u &= -4(y - 1)y, \\ v &= 0. \end{aligned}$$

O vetor posição da nova localização da interface é calculado como:

$$\mathbf{x}(s, t) = \mathbf{x}(s, 0) + \int_0^t \mathbf{u} \, dt = \mathbf{x}(s, 0) + \mathbf{u}\Delta t. \quad (5.10)$$

Na tabela 5, comparou-se a área da interface analítica $\Gamma_t = [\mathbf{x}(s, t)]$ com a área de Γ_t^{num} , em $t = 1$, vide figura 29. A área de Γ_t é obtido a partir de (5.2), uma vez que o escoamento é incompressível, e a área de Γ_t^{num} é obtido a partir de (5.3).

Figura 29 – Advecção de uma interface, inicialmente circular, com campo de velocidade prescrito, no tempo $t = 1$.

Fonte: Elaborada pelo autor.

Tabela 5 – Análise do erro entre a área exata e numérica dentro da interface, em diferentes resoluções de malha, no tempo $t = 1$.

Malha	Erro (ε)
32×32	1.560e-5
64×64	3.921e-6
128×128	9.800e-7
256×256	1.211e-7

Fonte: Dados da pesquisa.

5.2 Simulações numéricas de escoamentos bifásicos

5.2.1 Gota estática

Para validar a implementação da tensão superficial, verifica-se o balanço, na equação de Navier-Stokes, entre a força de tensão superficial e a pressão agindo na interface. Para isso, simula-se numericamente o problema de uma bolha (ou gota) imersa em um fluido, sem a influência do campo gravitacional (i.e., $\mathbf{g} = 0$). Esta é a popular relação de Laplace entre pressão e tensão superficial sobre uma bolha em equilíbrio (POPINET, 2018). Desse modo, o salto da pressão sobre a interface é calculado, analiticamente, pela equação:

$$\Delta p = \sigma \kappa = \sigma \left(\frac{1}{R} \right), \quad (5.11)$$

conhecida como equação de Young-Laplace.

Para fazer o teste da gota estática, avaliou-se o comportamento de uma gota de etanol (de raio $R = 25$ mm) centrada em uma cavidade quadrada ($[100 \text{ mm}]^2$) cheia de ar. Para isso, levou-se em consideração os seguintes parâmetros:

Tabela 6 – Parâmetros físicos para a simulação da gota estática.

ρ_1	ρ_2	μ_1	μ_2	$\mathbf{g}$	σ
1.1768	787.88	2×10^{-3}	2.4×10^{-2}	0	0.02361

Fonte: Štrubelj, Tiselj e Mavko (2009)

em que a unidade de medida das densidades é avaliada em kg/m^3 , as viscosidades são avaliadas em $\text{Pa} \cdot \text{s}$ e a tensão interfacial é avaliada em N/m . As propriedades físicas dos fluidos, descritas acima, foram retiradas de Štrubelj, Tiselj e Mavko (2009), em que os valores reais das viscosidades do ar e do etanol foram multiplicadas por 200 e 20, respectivamente, visando a diminuição da magnitude das velocidades espúrias e oscilações no formato da gota. Com isso, a partir da equação (5.11), espera-se que o salto da pressão seja $\Delta p = 0.9444$.

O erro relativo no salto da pressão é estimado pela expressão:

$$\varepsilon(\Delta p) = \frac{\max(|\Delta p_{aproximado} - \Delta p_{exato}|)}{\Delta p_{exato}}. \quad (5.12)$$

Foram feitas simulações numéricas com quatro diferentes resoluções de malha uniforme, estas são: 32, 64, 128, 256 células em cada direção. Como esperado, tem-se um decaimento do erro relativo com o refinamento da malha, fazendo com que o valor da pressão numérica convirja para o valor analítico, obtido em (5.11), vide tabela 7 e figura 31. Ademais, nota-se que a pressão é constante em cada fluido, exceto na interface, onde a pressão assume valor intermediário, como ilustrado na figura 30. Logo, o esquema de distribuição de forças, desenvolvido neste trabalho, apresenta boa precisão e consistência. Isso mostra que o trabalho em questão está de acordo com a literatura.

Figura 30 – Campo de pressão oriundo do teste da gota estática em uma malha 64×64 , no tempo $t = 1$.

Fonte: Elaborada pelo autor.

Tabela 7 – Análise de convergência do erro no salto da pressão sobre uma gota de etanol em diferentes resoluções de malha, no tempo $t = 1$.

Malha	Passo (Δ)	Erro (ε)	Razão	Ordem (p)
32x32	0.00313	7.86743e-03	-	-
64x64	0.00156	2.27658e-03	3.45581	1.78903
128x128	0.00078	7.72978e-04	2.94521	1.55837
256x256	0.00039	1.80008e-04	4.29412	2.10236

Fonte: Dados da pesquisa.

Figura 31 – Análise de convergência do erro no salto da pressão sobre uma gota de etanol em diferentes resoluções de malha, no tempo $t = 1$.

Fonte: Elaborada pelo autor.

5.2.2 Gota oscilante

Outro problema clássico na literatura é o teste da oscilação da gota, muito utilizado para a medição da tensão superficial e viscosidade dos fluidos. No trabalho de Sousa *et al.* (2004), por exemplo, esse teste é feito para verificar se a transferência entre a energia superficial e energia cinética é calculada corretamente, sendo importante quando o escoamento é dominado por efeitos de tensão superficial.

Esse teste consiste em verificar a frequência de oscilação de uma gota imersa em um fluido, sem a influência do campo gravitacional. Se uma gota tem um círculo como forma inicial, assim como no problema da gota estática (vide seção 5.2.1), ela tende a manter-se nessa forma mesmo com a evolução do tempo, visto que ela já está em seu ponto de equilíbrio, ou seja, a soma de todas as forças agindo sobre a gota tende a zero. O problema da gota oscilante consiste em causar uma pequena perturbação inicial na forma de equilíbrio da gota, e, desse modo, ela tende a oscilar até atingir sua forma de equilíbrio (círculo). A oscilação é amortecida pela viscosidade.

Tomou-se os dados apresentados no trabalho de Štrubelj, Tiselj e Mavko (2009), para a verificação dos resultados obtidos neste projeto. Logo, avaliou-se o comportamento de uma gota elíptica de etanol (com raios $R_x = 17.858$ mm e $R_y = 28.5$ mm) centrada em uma cavidade quadrada ($[75 \text{ mm}]^2$) cheia de ar. Para isso, levou-se em consideração as seguintes propriedades físicas:

Tabela 8 – Parâmetros físicos para a simulação da gota oscilante.

ρ_1	ρ_2	μ_1	μ_2	g	σ
1.1768	787.88	2×10^{-3}	2.4×10^{-2}	0	0.02361

Fonte: Štrubelj, Tiselj e Mavko (2009)

em que a unidade de medida das densidades é avaliada em kg/m^3 , as viscosidades são avaliadas em $Pa \cdot s$ e a tensão interfacial é avaliada em N/m . Os valores reais das viscosidades do ar e do etanol foram multiplicadas por 200 e 20, respectivamente. Os cálculos foram realizados em três diferentes resoluções de malha, estas são: 32×32 , 64×64 e 128×128 células.

Avaliando-se, então, a oscilação da posição de um ponto localizado ao norte da interface elíptica e comparando-a com o resultado de Štrubelj, Tiselj e Mavko (2009), na malha mais fina (128×128 células), obtém-se:

Figura 32 – Comparação dos resultados da oscilação da posição de um ponto, localizado ao norte de uma gota elíptica, obtidos com o HiG-Flow (32×32 , 64×64 e 128×128 células) e Štrubelj, Tiselj e Mavko (2009) (128×128 células).

Fonte: Elaborada pelo autor.

Com isso, os resultados do teste da gota oscilante, gerados no sistema HiG-Flow, mostram-se em conformidade com os resultados apresentados por Štrubelj, Tiselj e Mavko (2009).

5.2.3 Bolha ascendente

Por fim, para finalizar a bateria de testes, simulou-se o clássico problema da bolha ascendente, também muito utilizado para a verificação de resultados oriundos de simuladores de escoamentos bifásicos. Esse problema consiste em avaliar a ascensão de uma bolha dentro de uma coluna (canal) de fluidos. Nesse teste, considera-se a bolha com densidade menor do que o fluido que preenche a coluna, e, desse modo, com os efeitos do gradiente da pressão devido à gravidade, a bolha tende a se mover na direção oposta ao campo gravitacional.

Para o teste da ascensão da bolha, tomou-se como referência o trabalho de [Hysing et al. \(2007\)](#). Nesse trabalho, o autor considerou, inicialmente, uma bolha (de raio $R = 0.25$ cm) centrada no ponto $(0.5, 0.5)$ do domínio retangular $([0, 1] \times [0, 2])$, como ilustrado na figura 33:

Figura 33 – Configuração inicial do teste da bolha ascendente.

Fonte: Elaborada pelo autor.

Para as condições de contorno, levou-se em consideração a condição de contorno de Dirichlet para as velocidades, sendo $\mathbf{u} = 0$, e a condição de contorno de Neumann para a pressão.

Em geral, para a modelagem da ascensão da bolha, o número de Reynolds (Re), que representa a razão entre as forças inerciais e viscosas, e o número de Bond (Bo), que é a razão entre as forças devido ao campo gravitacional e a tensão interfacial, são levados em consideração como parâmetros adimensionais, sendo:

$$Re = \frac{\rho_0 \sqrt{g} D^{3/2}}{\mu_0} \quad \text{e} \quad Bo = \frac{\rho_0 g D^2}{\sigma}.$$

Assim, os seguintes parâmetros físicos e adimensionais foram considerados:

Tabela 9 – Parâmetros físicos e adimensionais para a simulação da bolha ascendente.

ρ_0	ρ_1	μ_0	μ_1	g	σ	Re	Bo
1000	100	10	1	0.98	24.5	35	10

Fonte: Hysing *et al.* (2007)

em que a unidade de medida das densidades é avaliada em kg/m^3 , as viscosidades são avaliadas em $Pa \cdot s$, a tensão interfacial é avaliada em N/m e a gravidade é avaliada em m/s^2 . Esses parâmetros foram escolhidos por Hysing *et al.* (2007) visando comparar resultados obtidos com diferentes softwares de simulação de escoamentos bifásicos, não tendo características realistas.

Os testes foram realizados para as seguintes resoluções de malha: 20×40 , 40×80 e 80×160 . Assim como no trabalho de Hysing *et al.* (2007), as simulações devem ser realizadas até o tempo dimensional $t = 3$ s, onde a bolha assume uma forma elipsoidal. Os resultados desses testes podem ser observados na figura 34:

Figura 34 – Bolha ascendente para diferentes resoluções de malha no tempo $t = 3$ s. A linha azul (em (b), (c) e (d)) corresponde ao centro do canal na direção y .

(a) Hysing *et al.* (2007) (b) 20×40 - HiG-Flow (c) 40×80 - HiG-Flow (d) 80×160 - HiG-Flow

Fonte: Elaborada pelo autor.

O formato da bolha, em todos os testes, é semelhante ao resultado apresentado por Hysing *et al.* (2007). Apenas a bolha resultante do teste feito com a resolução mais grosseira (20×40) apresentou uma pequena difusão na interface.

Além da avaliação qualitativa do problema em questão, por meio da análise da geometria da bolha (vide figura 34), fez-se também uma análise quantitativa, visando determinar a precisão

das simulações apresentadas nesta seção. Para isso, foram feitos dois testes visando avaliar quantitativamente a evolução temporal da bolha, estes são:

- análise da evolução do centro de massa da bolha, definido como sendo:

$$y_c = \frac{\int_{\Omega_1} y \, dy}{\int_{\Omega_1} 1 \, dy}, \quad (5.13)$$

sendo Ω_1 a região preenchida pela bolha. Em se tratando do método VOF, a fração de volume pode ser utilizada para calcular a equação (5.13). Assim, essa equação pode ser reescrita como:

$$y_c = \frac{\sum_{i=1}^N y_i \cdot f \cdot \Delta x \Delta y}{\sum_{i=1}^N f \cdot \Delta x \Delta y}, \quad (5.14)$$

onde N é o número de células que contêm fração de volume não nula.

- análise da velocidade de ascensão da bolha, definida como sendo:

$$v_c = \frac{\int_{\Omega_1} v \, dy}{\int_{\Omega_1} 1 \, dy}. \quad (5.15)$$

No contexto do VOF, a equação (5.15) pode ser reescrita como:

$$v_c = \frac{\sum_{i=1}^N v_i \cdot f \cdot \Delta x \Delta y}{\sum_{i=1}^N f \cdot \Delta x \Delta y}. \quad (5.16)$$

No trabalho de [Hysing et al. \(2007\)](#), em simulações numéricas utilizando os códigos TP2D (do inglês, Transport Phenomena in 2D), FreeLIFE e MooNMD, a posição final do centro de massa da bolha em relação ao eixo y , na malha 1/320, e a velocidade máxima de ascensão da bolha, na mesma malha, podem ser observadas na tabela 10:

Tabela 10 – Velocidade máxima de ascensão da bolha e posição final do centro de massa no eixo y , usando os softwares: TP2D, FreeLIFE e MooNMD

Software	TP2D	FreeLIFE	MooNMD
$V_{c,max}$	0.2417	0.2421	0.2417
$y_c(t=3)$	1.0813	1.0799	1.0817

Fonte: [Hysing et al. \(2007\)](#)

Abaixo, na tabela 11, seguem os resultados da evolução do centro de massa e velocidade de ascensão da bolha obtidos com o sistema HiG-Flow:

Tabela 11 – Velocidade máxima de ascensão da bolha e posição final do centro de massa no eixo y.

$1/\Delta_m$	1/20	1/40	1/80
$V_{c,max}$	0.30800	0.30340	0.30170
$y_c(t=3)$	1.0236	1.0182	1.0171

Fonte: Dados da pesquisa.

O centro de massa da bolha simulada no sistema HiG-Flow apresentou um pequeno deslocamento em relação ao centro de massa da bolha apresentada por [Hysing et al. \(2007\)](#), e isso gerou uma diferença na posição final do centro de massa no tempo $t = 3$ s. Além disso, a bolha gerada no sistema HiG-Flow também atingiu uma velocidade de ascensão maior do que a apresentada pela referência. Presumiu-se que essa diferença adveio do fato de os softwares utilizados por [Hysing et al. \(2007\)](#) empregarem o método Level-Set para representar a interface, ao passo que no sistema HiG-Flow utilizou-se o método VOF para isso. A representação geométrica da evolução do centro de massa e da velocidade de ascensão da bolha pode ser observada nas figuras 35 e 36 abaixo:

Figura 35 – Evolução do centro de massa da bolha na direção vertical (y), para diferentes resoluções de malha, até o tempo $t = 3$ s.

Fonte: Elaborada pelo autor.

Figura 36 – Velocidade de ascensão da bolha, para diferentes resoluções de malha, até o tempo dimensional $t = 3$ s.

Fonte: Elaborada pelo autor.

Essa diferença diminuiu consideravelmente quando comparou-se os resultados gerados no HiG-Flow com resultados de códigos que utilizam o mesmo esquema para a representação da interface. No teste abaixo, por exemplo, onde comparou-se os resultados gerados no sistema HiG-Flow com os resultados reportados por [Kalland \(2008\)](#), que também utilizou o método VOF para modelar a interface, observou-se uma compatibilidade maior nos resultados, vide figura 37. Nesse teste, simulou-se a ascensão de uma bolha de gás em uma coluna $([0, 2] \times [0, 4])$ de líquido, com as seguintes propriedades:

Tabela 12 – Parâmetros físicos para a simulação da bolha ascendente.

ρ_0	ρ_1	μ_0	μ_1	$\mathbf{g}$	σ
1	0.0013	0.002	3.2×10^{-5}	5	1.46

Fonte: [Kalland \(2008\)](#)

Figura 37 – Comparação dos resultados do teste da bolha ascendente no tempo $t = 0.5$ s. A linha contínua (preta) representa a bolha gerada no HiG-Flow e a linha tracejada (azul) representa a bolha reportada por [Kalland \(2008\)](#). A linha azul dividindo o domínio $([0, 2] \times [0, 4])$ corresponde ao centro do domínio na direção y .

Fonte: Elaborada pelo autor.

CONCLUSÕES

6.1 Considerações finais

Nesta dissertação de mestrado, teve-se como objetivo principal tornar o sistema HiG-Flow apto para a realização de simulações numéricas de escoamentos bifásicos. Desse modo, focou-se na implementação e verificação de métodos numéricos utilizados na modelagem da dinâmica de escoamentos desse tipo. A formulação matemática, deste trabalho, foi desenvolvida para a solução das equações de Navier-Stokes para escoamentos bifásicos bidimensionais.

Para a localização e mapeamento da interface foi implementado o método VOF no sistema HiG-Flow. O primeiro passo para o desenvolvimento do método VOF, consistiu na implementação da técnica para a inicialização do campo de fração de volume, descrito na seção 4.2. Para garantir a precisão do método, cada célula interfacial foi dividida em $n_{sc} = 8^2$ subcélulas. Para $n_{sc} < 6$, o método de inicialização não apresenta boa precisão, e isso pode culminar em erros no cálculo da curvatura e do vetor normal à interface. Os resultados obtidos com o método de inicialização desenvolvido neste projeto, vide seção 5.1.1, mostraram-se em conformidade com o que é reportado por [López *et al.* \(2009\)](#).

Para o cálculo da curvatura e do vetor normal à interface, optou-se pela implementação do método da função de altura (HF), pois, além de ser o mais tradicional da literatura, tem um bom equilíbrio entre custo computacional e precisão numérica. Por via de regra, o método HF é acoplado a outros métodos para o cálculo da curvatura interfacial, pois o método HF padrão pode apresentar inconsistência em alguns pontos do seu estêncil. Neste projeto, implementou-se o método HF modificado em vez de fazer tal acoplamento, vide seção 4.3. Pôde-se notar que os resultados do cálculo da curvatura apresentaram oscilações na razão de convergência do erro em função do refinamento da malha, quando a inicialização do VOF foi feita refinando cada célula interfacial por 8^2 subcélulas, mas atingiu ordem quadrática quando a inicialização foi feita refinando cada célula interfacial por 64^2 subcélulas, mostrando-se compatível com a literatura.

Para a advecção da interface, optou-se pelo algoritmo geométrico PLIC (*Piecewise-Linear Interface Calculation*), descrito na seção 4.4, uma vez que o VOF não apresenta bom comportamento quando acoplado a métodos algébricos para a advecção. No PLIC, a interface deve ser reconstruída a cada passo de tempo. Na literatura, existem diversos algoritmos de reconstrução que dão uma estimativa para o vetor normal à interface, baseados nas aproximações de diferenças finitas do gradiente da fração de volume satisfazendo, em geral, algum critério de minimização, o que aumenta a complexidade e o custo do processo. Para minimizar esse problema, neste projeto, utilizou-se o método da função de altura para estimar o vetor normal à interface durante o passo de reconstrução. A verificação do método PLIC-VOF foi apresentada na seção 5.1.3.

Foi implementado o modelo de tensão interfacial contínua para a descrição da força de tensão interfacial, descrito na seção 4.5. Esse esquema foi validado através das simulações dos clássicos problemas da gota estática 5.2.1 e gota oscilante 5.2.2, apresentando resultados satisfatórios em ambos os casos.

Os resultados obtidos, neste projeto, foram verificados e validados por meio de dados e problemas clássicos da literatura, tais como: gota estática 5.2.1, gota oscilante 5.2.2 e bolha ascendente 5.2.3, e mostraram-se compatíveis.

Todas as simulações numéricas deste trabalho foram realizadas utilizando o sistema HiG-Flow. As interpolações e aproximações de propriedades físicas na faceta e a resolução dos sistemas lineares foram feitas utilizando o sistema HiG-Tree. Esses sistemas mostraram-se robustos, eficientes e precisos.

Com isso, o objetivo de tornar o sistema HiG-Flow operacional para simular escoamentos bifásicos de fluidos foi atingido, e, assim, o trabalho pôde ser concluído.

6.2 Trabalhos futuros

Pretende-se aperfeiçoar o cálculo da curvatura, visto que a grande desvantagem do VOF em relação aos demais métodos de representação da interface encontra-se na estimativa da curvatura. Pretende-se, também, implementar o método VOF-3D, bem como modelos constitutivos poliméricos para a simulação de escoamentos bifásicos viscoelásticos 3D. E, por fim, implementar uma técnica de refinamento de malha adaptativa.

REFERÊNCIAS

ALVES, M.; OLIVEIRA, P.; PINHO, F. A convergent and universally bounded interpolation scheme for the treatment of advection. **International journal for numerical methods in fluids**, Wiley Online Library, v. 41, n. 1, p. 47–75, 2003. Citado na página 32.

BORNIA, G.; CERVONE, A.; MANSERVISI, S.; SCARDOVELLI, R.; ZALESKI, S. On the properties and limitations of the height function method in two-dimensional cartesian geometry. **Journal of Computational Physics**, Elsevier, v. 230, n. 4, p. 851–862, 2011. Citado nas páginas 48 e 51.

BRACKBILL, J. U.; KOTHE, D. B.; ZEMACH, C. A continuum method for modeling surface tension. **Journal of computational physics**, Elsevier, v. 100, n. 2, p. 335–354, 1992. Citado na página 63.

FALGOUT, R. D.; YANG, U. M. hypre: A library of high performance preconditioners. In: SPRINGER. **International Conference on Computational Science**. [S.l.], 2002. p. 632–641. Citado na página 30.

FIGUEIREDO, R. A. **Simulação numérica de escoamentos viscoelásticos multifásicos complexos**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado nas páginas 52, 59, 60 e 63.

HIRT, C. W.; NICHOLS, B. D. Volume of fluid (vof) method for the dynamics of free boundaries. **Journal of computational physics**, Elsevier, v. 39, n. 1, p. 201–225, 1981. Citado nas páginas 25, 43 e 44.

HYSING, S.; TUREK, S.; KUZMIN, D.; PAROLINI, N.; BURMAN, E.; GANESAN, S.; TOBISKA, L. Proposal for quantitative benchmark computations of bubble dynamics. **International Journal for Numerical Methods in Fluids**, 2007. Citado nas páginas 76, 77, 78 e 79.

IVEY, C.; MOIN, P. Conservative volume of fluid advection method on unstructured grids in three dimensions. **Center for Turbulence Research Annual Research Briefs**, p. 179–192, 2012. Citado na página 59.

KALLAND, K. M. **A Navier-Stokes Solver for Single-and Two-Phase Flow**. Dissertação (Mestrado), 2008. Citado nas páginas 16, 41, 62, 63, 80 e 81.

LAGES, C. F. A. **Métodos numéricos para escoamentos multifásicos em malhas hierárquicas**. Dissertação (Mestrado) — Universidade de São Paulo, 2016. Citado nas páginas 25, 35, 36 e 41.

LAM, K. **A numerical surface tension model for two-phase flow simulations**. Dissertação (Mestrado) — Faculty of Science and Engineering, 2009. Citado na página 48.

LEONARD, B. P. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. **Computer methods in applied mechanics and engineering**, Elsevier, v. 19, n. 1, p. 59–98, 1979. Citado na página 32.

- LÓPEZ, J.; ZANZI, C.; GÓMEZ, P.; ZAMORA, R.; FAURA, F.; HERNÁNDEZ, J. An improved height function technique for computing interface curvature from volume fractions. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 198, n. 33-36, p. 2555–2564, 2009. Citado nas páginas 45, 66 e 83.
- LORENSEN, W. E.; CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In: ACM. **ACM siggraph computer graphics**. [S.l.], 1987. v. 21, n. 4, p. 163–169. Citado na página 46.
- MAREK, M.; ANISZEWSKI, W.; BOGUSLAWSKI, A. Simplified volume of fluid method (svof) for two-phase flows. **Task Quarterly**, v. 12, n. 3, p. 255–265, 2008. Citado na página 25.
- OSHER, S.; FEDKIW, R.; PIECHOR, K. Level set methods and dynamic implicit surfaces. **Appl. Mech. Rev.**, v. 57, n. 3, p. B15–B15, 2004. Citado na página 25.
- PARKER, B.; YOUNGS, D. **Two and three dimensional Eulerian simulations of uid ow with material interfaces**. [S.l.], 1992. Citado nas páginas 47, 51 e 53.
- PILLIOD, J. E. **An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods**. [S.l.]: U. of Calif., Davis, 1992. Citado na página 53.
- POPINET, S. An accurate adaptive solver for surface-tension-driven interfacial flows. **Journal of Computational Physics**, Elsevier, v. 228, n. 16, p. 5838–5866, 2009. Citado nas páginas 48, 49, 51, 63 e 69.
- _____. Numerical models of surface tension. **Annual Review of Fluid Mechanics**, Annual Reviews, v. 50, p. 49–75, 2018. Citado nas páginas 63 e 72.
- PUCKETT, E. G. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In: **Proceedings of the Fourth International Symposium on Computational Fluid Dynamics**. [S.l.: s.n.], 1991. p. 933–938. Citado na página 53.
- PUCKETT, E. G.; ALMGREN, A. S.; BELL, J. B.; MARCUS, D. L.; RIDER, W. J. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. **Journal of computational physics**, Elsevier, v. 130, n. 2, p. 269–282, 1997. Citado nas páginas 52, 59, 60 e 61.
- RIBERA, R. L. *et al.* Understanding the dynamics of gas-liquid-solid contact points. Universidade Federal de Uberlândia, 2015. Citado na página 66.
- RIDER, W. J.; KOTHE, D. B. Reconstructing volume tracking. **Journal of computational physics**, Elsevier, v. 141, n. 2, p. 112–152, 1998. Citado na página 61.
- RUDMAN, M. Volume-tracking methods for interfacial flow calculations. **International journal for numerical methods in fluids**, Wiley Online Library, v. 24, n. 7, p. 671–691, 1997. Citado na página 44.
- SCARDOVELLI, R.; ZALESKI, S. Direct numerical simulation of free-surface and interfacial flow. **Annual review of fluid mechanics**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 31, n. 1, p. 567–603, 1999. Citado na página 45.
- _____. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. **Journal of Computational Physics**, Elsevier, v. 164, n. 1, p. 228–237, 2000. Citado na página 53.

_____. Interface reconstruction with least-square fit and split eulerian–lagrangian advection. **International Journal for Numerical Methods in Fluids**, Wiley Online Library, v. 41, n. 3, p. 251–274, 2003. Citado na página [47](#).

SOUSA, F. D.; MANGIAVACCHI, N.; NONATO, L. G.; CASTELO, A.; TOMÉ, M. F.; FERREIRA, V. G.; CUMINATO, J. A.; MCKEE, S. A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces. **Journal of Computational Physics**, Elsevier, v. 198, n. 2, p. 469–499, 2004. Citado nas páginas [39](#) e [74](#).

SOUSA, F. S. d.; LAGES, C.; ANSONI, J. L.; CASTELO, A.; SIMAO, A. A finite difference method with meshless interpolation for incompressible flows in non-graded tree-based grids. **Journal of Computational Physics**, Elsevier, v. 396, p. 848–866, 2019. Citado nas páginas [29](#) e [30](#).

ŠTRUBELJ, L.; TISELJ, I.; MAVKO, B. Simulations of free surface flows with implementation of surface tension and interface sharpening in the two-fluid model. **International Journal of Heat and Fluid Flow**, Elsevier, v. 30, n. 4, p. 741–750, 2009. Citado nas páginas [16](#), [72](#), [74](#) e [75](#).

SUSSMAN, M.; PUCKETT, E. G. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. **Journal of computational physics**, Elsevier, v. 162, n. 2, p. 301–337, 2000. Citado na página [45](#).

TORREY, M. D.; CLOUTMAN, L. D.; MJOLSNESS, R. C.; HIRT, C. Nasa-vof2d: a computer program for incompressible flows with free surfaces. **NASA STI/Recon Technical Report N**, v. 86, 1985. Citado na página [47](#).

TRYGGVASON, G.; BUNNER, B.; ESMAEELI, A.; JURIC, D.; AL-RAWAHI, N.; TAUBER, W.; HAN, J.; NAS, S.; JAN, Y.-J. A front-tracking method for the computations of multiphase flow. **Journal of computational physics**, Elsevier, v. 169, n. 2, p. 708–759, 2001. Citado na página [25](#).

YOUNGS, D. L. Time-dependent multi-material flow with large fluid distortion. **Numerical methods for fluid dynamics**, Academic Press, 1982. Citado na página [52](#).