

ANA PAULA MOREIRA DE FREITAS

**SIMULAÇÃO DO ESCOAMENTO BIFÁSICO EM UM
TANQUE DE COMBUSTÍVEL**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2019

ANA PAULA MOREIRA DE FREITAS

**SIMULAÇÃO DO ESCOAMENTO BIFÁSICO EM UM TANQUE DE
COMBUSTÍVEL**

Projeto de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **BACHAREL** em **ENGENHARIA MECATRÔNICA**.

Área de concentração: Mecânica dos Fluidos.

Orientador: Prof. Dr. Francisco José de Souza

Uberlândia - MG

2019

"A nossa ciência é parcial, a nossa profecia é imperfeita. Quando chegar o que é perfeito, o imperfeito desaparecerá.

Quando eu era criança, falava como criança, pensava como criança, raciocinava como criança. Desde que me tornei homem, eliminei as coisas de criança.

Hoje vemos como por um espelho, confusamente; mas então veremos face a face. Hoje conheço em parte; mas então conhecerei totalmente, como eu sou conhecido.

Por ora subsistem a fé, a esperança e a caridade - as três. Porém, a maior delas é a caridade."

1 Coríntios 13:9-13

AGRADECIMENTOS

Primeiramente agradeço a Deus, pois sem Ele nada disso seria possível. Ele que sempre me abençoou e iluminou os momentos mais escuros.

À minha amada mãe, Marlene, que nunca mediu esforços ao lutar pelas filhas, abdicando de si mesma inúmeras vezes. Ela quem nunca deixou de procurar possibilidades mesmo quando tudo parecia impossível. Ao meu igualmente amado pai, Celino, que foi o primeiro a, singelamente, plantar em mim o amor pela engenharia. Quem me inspirou a ir rumo ao meu sonho. Aos dois juntos, agradeço pelos exemplos de humildade, empatia e honestidade. Por me apoiarem em projetos pessoais, mesmo sem compreendê-los. Agradeço por todo amor, confiança e, claro, por me corrigirem sempre que necessário.

À minha estimada irmã, Michelet, quem aprecio a doçura, inteligência e persistência. Agradeço por ser o ótimo modelo de dedicação em que me apoiei e por todo reforço cedido nesta etapa.

Ao meu querido Arthur, que entrou em minha vida no momento exato, inspirando e dando cor aos meus dias. Quem tanto me compreende, apoia, motiva e auxilia a dar movimento aos meus sonhos. Quem me apaixona todos os dias.

Aos meus valiosos amigos, Raissa, Matheus e Guilherme que, próximos ou distantes, fizeram-se presentes com toda harmonia de uma amizade real. Por partilharem comigo o dia-a-dia da graduação, facilitando assim cada etapa.

Ao Rafael, por ser o primeiro a me apresentar esta tão apreciada área de pesquisa. Por ter me apoiado e estado comigo em um momento turbulento, auxiliando-me a dar o primeiro passo que deu sentido aos meus dias como pessoa e profissional.

À Renata Regina que, ao conhecê-la na fase final de minha graduação, trouxe conforto, conselhos significativos e auxílio, levando de mim uma afeição singular.

Ao Prof. Dr. Santos Alberto Enriquez Remigio, pela primeira oportunidade na pes-

quisa, fundamental para moldar meus anseios profissionais, pela amizade e ensinamentos. Ao meu orientador Prof. Dr. Francisco José de Souza, pela oportunidade, orientação e paciência ao longo deste trabalho. Ao Prof. Dr. Rubens Gedraite pelo incentivo e sabedoria tão bem colocados.

À assistência estudantil da Universidade Federal de Uberlândia pelo bom trabalho prestado que, sem tal, não seria possível a permanência na graduação.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de realizar essa importante etapa de um sonho maior, a graduação em Engenharia Mecatrônica. E, por fim, aos professores e técnicos que, de uma forma ou de outra, me auxiliaram.

FREITAS, A. P. M., **Simulação do Escoamento Bifásico em um Tanque de Combustível**. 2019. Projeto de Fim de Curso, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

Este trabalho tem como objetivo simular o escoamento bifásico em um tanque de combustível usando o Método Lattice Boltzmann. A simulação foi realizada através da utilização de um código aberto chamado *Palabos*. O *sloshing* no tanque foi aproximado por um problema de *dam break*. As fases líquido e gás no escoamento foi representado por água e ar, respectivamente. A pressão de impacto foi monitorada em uma parede do tanque e os resultados foram comparados com os encontrados na literatura, mostrando que o método se comportou bem para o tipo de problema.

Palavras Chave: Método Lattice Boltzmann, Sloshing, Escoamento Bifásico, Palabos

FREITAS, A. P. M., **Simulation of Biphasic Flow in a Fuel Tank**. 2019. Projeto de Fim de Curso, Universidade Federal de Uberlândia, Uberlândia.

ABSTRACT

This work aims to simulate biphasic flow in a fuel tank using the Lattice Boltzmann Method. The simulation was performed using an open source code called Palabos. The sloshing in the tank was approached by a dam break problem. The liquid and gas phases in the flow were represented by water and air respectively. The impact pressure was monitored on a wall of the tank and the results were compared with those found in the literature, showing that the method behaved well for the type of problem.

Keywords: Lattice Boltzmann Method, Sloshing, Biphasic Flow, Palabos.

Lista de Figuras

1.1	Simulação de uma turbina eólica, com movimento das pás, realizada com o software XFLOW (http://www.xflowcfcd.com/industries/view/energy). . . .	4
2.1	Técnicas de simulação.	8
2.2	Vetor posição e velocidade em uma partícula em movimento.	9
2.3	Uma partícula movendo-se livremente dentro de uma caixa.	10
2.4	Vista em corte de duas esferas com raios c e $c + dc$	14
2.5	Função distribuição de Maxwell-Boltzmann.	16
2.6	Vetor posição e velocidade para uma partícula antes e após a aplicação de uma força F	19
2.7	Arranjos de <i>lattice</i> para problemas unidimensionais. Fonte: Mohamad (2011). .	23
2.8	Arranjos de <i>lattice</i> para problemas bidimensionais. Fonte: Mohamad (2011). .	24
2.9	Arranjo de Lattice D2Q9 para problemas bidimensionais. Fonte: Mohamad (2011).	24
2.10	Arranjo de <i>lattice</i> D3Q15 para problemas tridimensionais. Fonte: Mohamad (2011).	25
2.11	Arranjo de <i>lattice</i> D3Q19 para problemas tridimensionais. Fonte: Mohamad (2011).	26
2.12	Placa com comprimento de 100 unidades e temperatura $T = 1$ na extremidade esquerda.	32
2.13	Nós distribuídos unidimensionalmente na abordagem de Diferenças Finitas. . . .	32
2.14	Diagrama esquemático para arranjo de <i>lattice</i> unidimensional. Fonte: Mohamad (2011).	35
2.15	<i>Lattices</i> para um problema de difusão unidimensional. Fonte: Mohamad (2011).	35
2.16	Esquema para a etapa de advecção. Fonte: Mohamad (2011).	36

2.17	Comparação dos resultados obtidos através do LBM e FDM para uma placa com condições de contorno de Dirichlet.	37
3.1	Danos causados à plataforma de combate a incêndios da <i>Emerald</i> FSU por água verde. Fonte: Kleefsman (2005)	39
3.2	Danos causados na proa do FPSO Varg. Fonte: Kleefsman (2005)	40
3.3	Impressão artística da nave espacial NEAR. Fonte: Veldman (2006)	41
3.4	Representação do tanque com a posição de interesse para monitoramento da pressão.	47
3.5	Representação do tanque com o volume de fluido a ser liberado.	48
4.1	Histórico de densidade para diferentes valores de tempo da simulação.	50
4.2	Evolução da velocidade entre $t = 0$ s e $t = 1$ s.	51
4.3	Evolução da velocidade entre $t = 1,15$ s e $t = 1,7$ s.	52
4.4	Evolução da velocidade entre $t = 2,0$ s e $t = 2,5$ s.	53
4.5	Evolução da velocidade entre $t = 3,0$ s e $t = 4,0$ s.	54
4.6	Pressão de impacto obtida ao longo do tempo e pressão após filtragem.	55
4.7	Histórico da pressão exercida pelo impacto de ondas na parede vertical esquerda após filtragem.	56
4.8	Comportamento para a pressão causada pelo impacto de uma onda sobre uma parede vertical. Fonte: Peregrine (2003).	57
4.9	Momento em que há impacto de fluido pela segunda vez na parede vertical esquerda em $t = 5,62$ s.	58

Lista de Símbolos

x	- Posição no eixo x do sistema de coordenadas cartesiano
y	- Posição no eixo y do sistema de coordenadas cartesiano
z	- Posição no eixo z do sistema de coordenadas cartesiano
c	- Velocidade resultante das partículas
c_x	- Componente da velocidade na direção x
c_y	- Componente da velocidade na direção y
c_z	- Componente da velocidade na direção z
c_s	- Velocidade do som
c_{avg}	- Velocidade média de um conjunto de partículas
c_{RMS}	- Raiz da velocidade quadrática média
$\vec{F}$	- Soma das forças interpartículas e forças externas
$\vec{r}$	- Vetor posição da partícula
$\vec{c}$	- Vetor velocidade da partícula
$\vec{a}$	- Vetor aceleração da partícula
$\vec{u}$	- Vetor velocidade do fluido
$\vec{u}_a$	- Velocidade da partícula em relação a velocidade do fluido
m	- Massa da partícula
M	- Massa molecular
P	- Pressão
T	- Temperatura
F	- Força resultante
N	- Número de partículas
V	- Volume
n_p	- Número de partículas por unidade de volume
K_e	- Energia cinética
n	- Número de mols
N_A	- Número de Avogrado
R	- Constante universal dos gases
k	- Constante de Boltzmann
k_T	- Condutividade térmica
C	- Calor específico
E	- Energia total
S	- Entropia termodinâmica
W	- Número de estados microscópicos
e	- Energia interna
l	- Escala de distância

t^*	-	Tempo adimensional
x^*	-	Comprimento adimensional
u_n	-	Componente normal da velocidade do fluido
u_t	-	Componente tangencial da velocidade do fluido
p_0	-	Pressão atmosférica
$f(c_x)$	-	Função distribuição de probabilidade de velocidade na direção x
$f(c_y)$	-	Função distribuição de probabilidade de velocidade na direção y
$f(c_z)$	-	Função distribuição de probabilidade de velocidade na direção z
f^{eq}	-	Função distribuição de equilíbrio
Δt	-	Tempo entre colisões das partículas
Δx	-	Comprimento de ligação do <i>lattice</i>
Φ	-	Função distribuição
π	-	Número pi
Ω	-	Operador de colisão
∇	-	Operador gradiente
ρ	-	Densidade do fluido
ω	-	Frequência de colisão
τ	-	Tempo de relaxamento
β	-	Fator de ponderação
λ	-	Parâmetro escalar qualquer
Ψ	-	Variável dependente na equação da difusão
α	-	Difusividade térmica
τ	-	Tempo de relaxamento
ν	-	Viscosidade cinemática do fluido
Γ	-	Coefficiente de difusão
ϵ	-	Número de Knudson
θ	-	Temperatura adimensionalizada
σ	-	Tensão superficial
κ	-	Curvatura total da superfície livre
μ	-	Viscosidade dinâmica do fluido
γ	-	Fator de filtragem no filtro passa-baixa

Lista de Abreviações e Siglas

CFD	-	Computational Fluid Dynamics
FDP	-	Função Distribuição de Probabilidade
FPSO	-	Floating Production, Storage and Offloading Unit
FSO	-	Floating Storage and Offloading Unit
LBM	-	Lattice Boltzmann Method
FDM	-	Finite Difference Method
MD	-	Molecular Dynamics
NS	-	Navier-Stokes
GNL	-	Gás Natural Liquefeito
FSU	-	Floating Storage Unit
NEAR	-	Near Earth Asteroid Rendezvous
SI	-	Sistema Internacional

SUMÁRIO

1	INTRODUÇÃO	1
2	REVISÃO BIBLIOGRÁFICA	6
2.1	Cinética das Partículas	6
2.1.1	Dinâmica das Partículas	8
2.1.2	Pressão e Temperatura	10
2.1.3	Função Distribuição	12
2.1.4	Distribuição de Boltzmann	17
2.2	A Equação de transporte de Boltzmann	18
2.3	A Aproximação BGKW	21
2.4	Arranjos de Lattice	22
2.4.1	Unidimensional	22
2.4.2	Bidimensional	24
2.4.3	Tridimensional	25
2.5	Função de Distribuição de Equilíbrio	26
2.6	Difusão de Calor em uma placa infinita submetida a uma temperatura constante	27
2.6.1	Equação da Difusão	27
2.6.2	Função Distribuição de Equilíbrio	28
2.6.3	Expansão Chapman-Enskog	29
2.6.4	Modelagem com condição de contorno de Dirichlet	31
3	METODOLOGIA	38
3.1	Estudo do Sloshing	38

	1
3.2 Utilização da Palabos	41
3.3 Modelagem Numérica	44
3.3.1 Condições de Contorno	45
3.3.2 Superfície Livre	45
3.3.3 Monitoramento da Pressão no tanque	45
3.3.4 Hipóteses para a simulação	48
4 RESULTADOS E DISCUSSÕES	49
5 CONCLUSÕES E TRABALHOS FUTUROS	59
5.1 Conclusões	59
5.2 Trabalhos Futuros	60
REFERÊNCIAS BIBLIOGRÁFICAS	61
APÊNDICES	63

CAPÍTULO I

INTRODUÇÃO

Ao longo dos últimos anos, a Dinâmica dos Fluidos Computacional (CFD, do inglês *Computational Fluid Dynamics*) tem tido um papel fundamental no desenvolvimento de produtos mais eficientes, tais como aeronaves, automóveis e trens de alta velocidade. Além de permitir a otimização do projeto, há também a vantagem de reduzir consideravelmente o custo e o ciclo de desenvolvimento, já que muitos experimentos dispendiosos e complexos não precisam ser realizados.

Do ponto de vista científico, a Dinâmica dos Fluidos Computacional também revela-se como ferramenta essencial à compreensão e análise de vários fenômenos cuja investigação experimental é difícil ou mesmo impossível em alguns casos.

Apesar de sua extensiva aplicação no mundo moderno, os métodos comumente empregados em CFD não têm uma base matemática sólida, e foram derivados nas décadas de 80 a 90 com base nas equações de Navier-Stokes (Ferziger; Peric, 2013). Sabe-se que tais métodos foram desenvolvidos principalmente por engenheiros que procuravam soluções práticas para problemas envolvendo escoamentos. De forma bastante empírica, os métodos de discretização das equações de Navier-Stokes foram propostos e testados, buscando-se evitar os problemas comuns de instabilidade numérica devidos à precisão finita dos processadores digitais. Infelizmente, uma consequência do desenvolvimento destes métodos foi o aumento da complexidade dos algoritmos envolvidos. Mesmo para problemas simples como o escoamento laminar de um fluido em um canal, vários sistemas lineares envolvendo milhares de incógnitas devem ser resolvidos, resultando

em um custo computacional considerável. Considerando escoamentos em regime turbulento, o tempo de CPU cresce conforme uma potência do número de Reynolds, e a solução precisa em tempo hábil exige clusters de processadores. Os algoritmos mais popularmente utilizados, da família SIMPLE (Ferziger; Peric, 2013), são inerentemente ineficientes em função do chamado acoplamento pressão-velocidade. O avanço destas variáveis deve ser contido para evitar divergência do conjunto de equações, o que exige várias iterações.

Outra desvantagem das equações de Navier-Stokes é que sua aplicabilidade é limitada pela hipótese do contínuo, utilizada em sua derivação. Embora essa hipótese seja válida para diversos problemas de interesse prático, há várias situações na engenharia que demandam uma abordagem mais adequada, como nanotecnologia, voo supersônico e dinâmica molecular. Nestes casos, a aplicação da hipótese do contínuo pode gerar resultados irrealistas, já que a escala de interesse é da ordem da distância média entre partículas/moléculas que compõem o fluido.

Como uma alternativa interessante à simulação das equações de Navier-Stokes, surge a abordagem do Método Lattice Boltzmann (Succi, 2001; Wolf-Gladrow, 2005). Assim, este trabalho tem como objetivo principal a simulação do escoamento bifásico em um tanque de combustível através da utilização do método Lattice Boltzmann. Tal método foi escolhido devido ao fácil tratamento de fluxos multifásicos e multicomponentes sem a necessidade de rastrear as interfaces entre as diferentes fases e, além disso, a obtenção da pressão é possível a partir de uma simples equação algébrica.

Resumidamente, o método Lattice Boltzmann consiste em resolver a equação de Boltzmann (Chapman; Cowling, 1970), que por sua vez pode ser compreendida como análoga das equações de Navier-Stokes em nível molecular. Essencialmente, a equação de Boltzmann descreve a dinâmica espacial-temporal de uma grandeza estatística chamada de função distribuição de probabilidade (FDP). A principal vantagem em relação às equações de Navier-Stokes é não estar limitada pela hipótese do contínuo, o que permite sua aplicação a uma gama de problemas mais vasta que as primeiras. É possível, portanto, descrever o movimento de fluidos que não se estejam no chamado regime hidrodinâmico.

Outra grande vantagem do método Lattice Boltzmann é sua eficiência. O algoritmo é consideravelmente mais simples que aquele necessário para a solução das equações do contínuo, já que sistemas lineares não precisam ser resolvidos. Também não há o chamado acoplamento

pressão-velocidade, pois a pressão é obtida de uma equação de estado algébrica. De forma sintética, apenas partículas com velocidade discreta e determinados pesos descrevem a FDP em cada ponto. O avanço no tempo de um escoamento é descrito por duas operações explícitas: colisão e advecção das partículas, as quais serão explicadas na próxima seção. As propriedades macroscópicas de interesse como velocidade são simplesmente calculadas com base em médias em cada ponto. Conclui-se assim que é possível descrever escoamentos complexos com este método com recursos mais modestos que aqueles normalmente necessários para a solução das equações de Navier-Stokes. Em função de sua simplicidade, a extensão do método para aplicação em computação de alto desempenho (paralelização) é uma tarefa bastante realizável, diferentemente das equações do contínuo.

Apesar de suas aparentes vantagens, o método Lattice Boltzmann encontra certa resistência na comunidade científica, principalmente entre engenheiros. No entanto, nas comunidades internacionais envolvendo físicos, o método tem sido muito sucedido quando aplicado a problemas similares ao realizado no presente trabalho. Possivelmente, o ceticismo atual resulta da bem conhecida dificuldade de se representar bem geometrias complexas. Neste sentido, já há soluções efetivas, como o refinamento adaptativo (www.palabos.org/). Há inclusive códigos comerciais de uso bastante simples quando comparado aos softwares convencionais. Para exemplificar, a Fig. 1.1 mostra a simulação de uma turbina eólica, realizada pelo software XFLOW (<http://www.xflowcfld.com/>), com o movimento rotativo das pás. Esta aplicação ainda representa um desafio para as técnicas convencionais de CFD, já que há a necessidade de geração da malha a cada passo de tempo em função da movimentação das fronteiras.

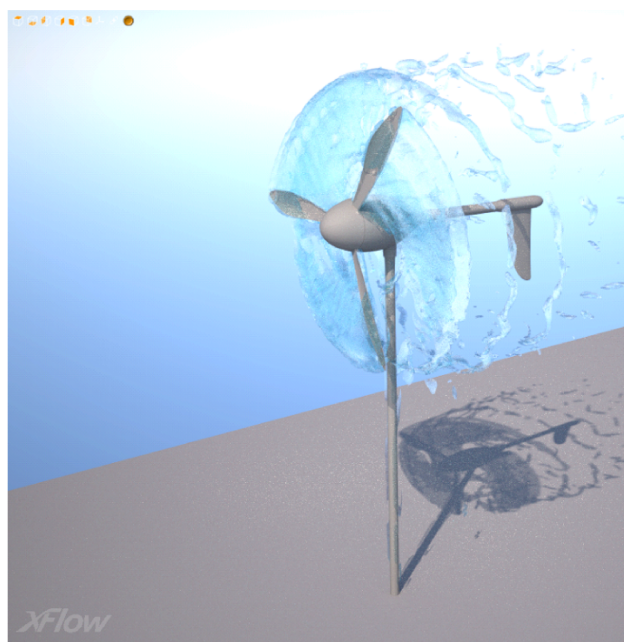


Figura 1.1 – Simulação de uma turbina eólica, com movimento das pás, realizada com o software XFLOW (<http://www.xflowcfd.com/industries/view/energy>).

O fenômeno simulado no presente trabalho é conhecido por *sloshing* que, de forma simplificada, significa qualquer movimento da superfície livre do líquido no interior do seu recipiente (Ibrahim, 2005). Este fenômeno físico é de grande importância devido a sua aplicabilidade. Por isso, pesquisadores de diversas áreas, como, geofísica, sismologia e engenharias buscam estudá-lo e compreendê-lo. O problema de *sloshing* em recipientes móveis ou estáticos é uma grande preocupação da indústria aeroespacial e civil, para engenheiros nucleares, físicos, projetistas de caminhões tanques e navios petroleiros, e também para matemáticos (Ibrahim, 2005). Para exemplificar algumas aplicações é possível citar: projetos de reservatórios de líquidos posicionados no topo de edifícios elevados com a finalidade de controlar oscilações devidas a terremotos, estudos da influência do *sloshing* em veículos lançadores de satélites e em espaçonaves, efeitos desestabilizantes em navios petroleiros e plataformas do tipo FPSO (*Floating Production, Storage and Offloading Unit*) e FSO (*Floating Storage and Offloading Unit*).

Em um reservatório parcialmente cheio de líquido, a geração e propagação de ondas devido a uma aceleração lateral gera um movimento oscilatório por conta do deslocamento deste líquido em busca da posição de equilíbrio. Essas ondas gravitacionais são extremamente energéticas e devido a distribuição de forças, momentos, pressões e frequências naturais da superfície

livre é provocado efeitos sobre a estabilidade dinâmica e desempenho dos reservatórios móveis (Ibrahim, 2005). Por isso a importância de se investigar tais grandezas, pois o conhecimento de antemão delas torna possível a realização do projeto dos recipientes móveis evitando, por exemplo, explosões em tanques de combustível devido a fraturas por fragilização dos membros estruturais. Além disso, é possível aplicar a previsão destes parâmetros na estabilidade dinâmica de navios tanque de carga de gás natural liquefeito e petroleiros devido ao impacto hidrodinâmico de carga líquida. Já nas espaçonaves, a presença de líquido na estrutura de veículos espaciais requer cuidadosa análise uma vez que seu movimento dissipa energia interna no veículo além de alterar as propriedades de inércia e simetria da espaçonave (Junior, 2017). Um exemplo deste problema foi o primeiro pouso lunar da missão Apollo 11 quando, pouco antes do pouso, o *sloshing* causado pelo combustível restante provocou um movimento oscilatório do Módulo Lunar (de cerca de 2-3 graus), prejudicando seriamente a segurança da manobra de aterrissagem (Veldman, 2006).

Assim, o estudo das forças sobre um tanque em movimento acelerado permite prever o movimento do combustível, tornando possível a redução de ruídos, otimização da forma geométrica do tanque e sua localização.

Este trabalho foi organizado em quatro partes. A Seção 2 contém uma revisão bibliográfica do método empregado na simulação do problema proposto, juntamente com a implementação de um exemplo simples da difusão de calor em uma placa com condições de contorno de Dirichlet utilizando o Método Lattice Boltzmann. A Seção 3 apresenta as hipóteses assumidas na modelagem matemática e suas equações governantes, bem como informações sobre as simulações desenvolvidas. Na Seção 4 são analisados os resultados obtidos, comparando-os com os resultados encontrados na literatura. São monitorados qualitativamente os perfis de velocidade no tanque e quantitativamente as pressões exercidas em uma das paredes do tanque. Por fim, na Seção 5 encontra-se as principais conclusões e sugestões para trabalhos futuros.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

2.1 Cinética das Partículas

Resumidamente, a cinética estuda as relações entre forças não equilibradas e na mudança de movimento por elas provocada. No presente trabalho, os termos partículas e moléculas são usados sem distinção.

Na abordagem contínua de simulação das equações de transporte (massa, momento e energia), primeiro são identificadas os modelos matemáticos, sejam elas equações diferenciais ordinárias ou parciais. Logo, o domínio é discretizado em nós, volumes ou elementos, dependendo do método de solução adotado. Cada nó ou volume ou elemento contém uma grande quantidade de partículas, na ordem de 10^{16} , tratando de uma escala macroscópica. Assim, a velocidade, pressão e temperatura de todas as partículas são representadas por um valor nodal, ou calculadas em média em um volume finito, ou simplesmente assumidas linear ou bilinearmente variando de um nó para outro (Mohamad, 2011). Vale ressaltar que as propriedades termofísicas, tais como a viscosidade, condutividade térmica, capacidade de calor, entre outras, são em geral, parâmetros conhecidos, ou seja, parâmetros de entrada. Porém, em problemas inversos, uma ou mais dessas propriedades podem ser desconhecidas.

Também, pode-se considerar que o meio é formado por pequenas partículas que colidem umas com as outras. Referindo-se agora da microescala. Dessa forma, é necessário identificar as

forças interpartículas para resolver a equação diferencial ordinária da segunda lei de Newton que representa o princípio fundamental da dinâmica. Em cada passo de tempo, é necessário identificar a trajetória de cada partícula, isto é possível através do conhecimento de suas respectivas posições e velocidades. Percebe-se que nesse caso não há definição de temperatura, pressão e propriedades termofísicas. Por exemplo, a temperatura e a pressão estão relacionadas à energia cinética das partículas (massa e velocidade) e frequência de bombardeamento de partículas nas fronteiras, respectivamente (Mohamad, 2011). O método de solução para este caso é chamado de simulações de dinâmica molecular (MD, do inglês *Molecular Dynamics*). Porém, é impossível resolver problemas com escala de comprimento na ordem de *cm* com o MD, pois seria necessário identificar a posição (x, y, z) e as componentes de velocidade (c_x, c_y, c_z) , de cada partícula. Além disso, é fundamental que cada passo de tempo de simulação seja inferior ao tempo de colisão das partículas, que está na ordem de picossegundos ($10^{-12}s$).

O método Lattice Boltzmann (LBM, do inglês *Lattice Boltzmann Method*) é um método intermediário aos dois mencionados anteriormente. Ao não considerar o comportamento de cada partículas isoladamente, mas o comportamento de um conjunto de partículas como uma unidade remete a idéia principal de Boltzmann que é preencher a lacuna entre a microescala e a macroescala (Mohamad, 2011). As propriedades deste conjunto de partículas são representadas por funções de distribuição. Assim, esse método está na chamada mesoescala, ou seja, na escala intermediária entre o macroscópico e o microscópico. A figura 2.1 exemplifica os três métodos brevemente discutidos.

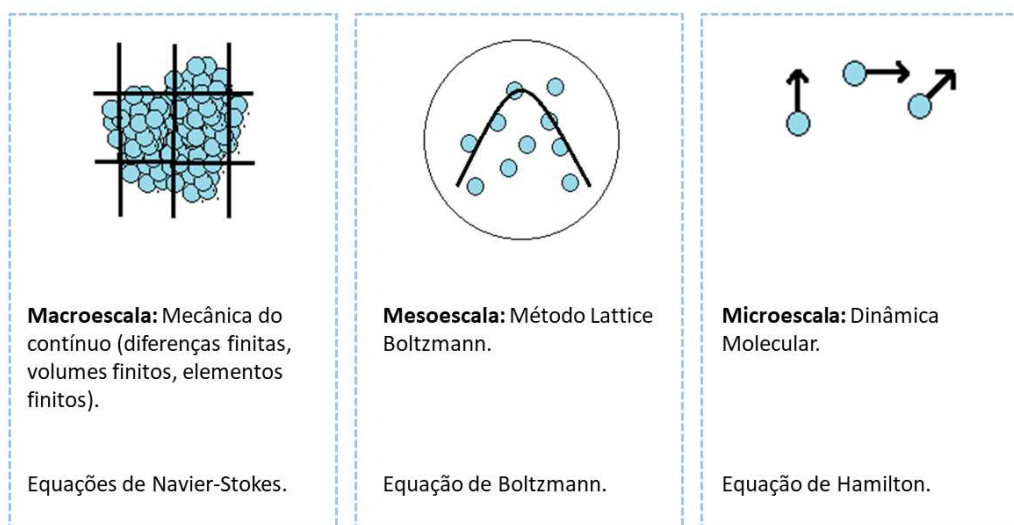


Figura 2.1 – Técnicas de simulação.

Diferentemente das soluções da equação de Navier-Stokes (NS), com o LBM não há necessidade de resolver a equação de Laplace em cada passo de tempo para satisfazer a equação da continuidade de fluxos incompressíveis e instáveis. Além disso, o LBM atende, com precisão confiável, problemas de microescala e macroescala, possui fácil aplicação para domínios complexos e é naturalmente adaptável a processos de computação paralela. Porém, o LBM demanda mais memória do computador em comparação com a solução por NS, o que não é uma limitação devido aos crescentes avanços computacionais.

2.1.1 Dinâmica das Partículas

De forma a simplificar, as partículas podem ser descritas por esferas sólidas que se movem aleatoriamente de forma conservatória em um espaço livre. Tal movimento satisfaz a conservação da massa, momento e energia. Logo, considerando uma massa constante, a segunda lei de Newton é aplicada e mostra que a taxa de mudança do momento linear é igual à força líquida aplicada

$$\vec{F} = \frac{m(d\vec{c})}{dt} = m\vec{a} \quad (2.1)$$

onde $\vec{F}$ são as forças interpartículas e forças externas, m representa a massa da partícula, $\vec{c}$ é o vetor de velocidades da partícula, $\vec{a}$ é o vetor de acelerações e t é o tempo.

A velocidade ($\vec{c}$) da partícula é determinada por:

$$\vec{c} = \frac{d\vec{r}}{dt} \quad (2.2)$$

onde $\vec{r}$ é o vetor posição da partícula.

Supondo uma força externa, $\vec{F}$, aplicada a uma partícula de massa m . A velocidade da partícula mudará de $\vec{c}$ para $\vec{c} + \vec{F}dt/m$ e sua posição mudará de $\vec{r}$ para $\vec{r} + \vec{c}dt$, conforme Fig. 2.2. Na ausência de uma força externa e assumindo que há colisão, a partícula se moverá livremente de um local para outro, sem alterar sua direção e velocidade, entre uma colisão e outra.

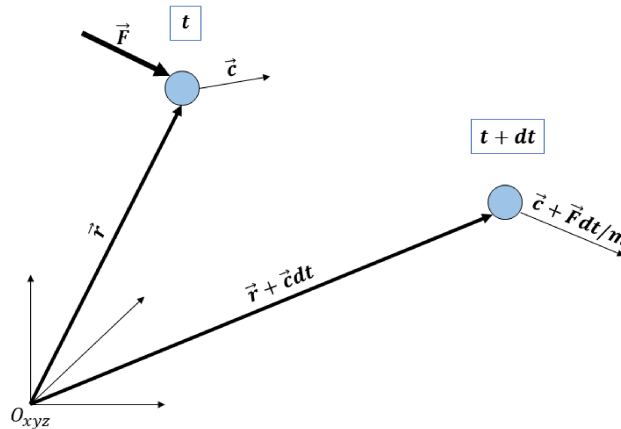


Figura 2.2 – Vetor posição e velocidade em uma partícula em movimento.

Com o aumento da energia interna do sistema, a magnitude da velocidade da partícula e a interação entre as partículas aumentam também. Macroscopicamente, aumentos na energia cinética das partículas são percebidos como aumentos de temperatura. Isso se dá pelo fato das partículas bombardearem continuamente as paredes dos recipientes em que estão contidas. A pressão nada mais é do que a força exercida por essas ações por unidade de área. Assim, observa-se a existência de uma relação entre temperatura e pressão, pois à medida que a energia cinética

das partículas aumenta, a temperatura aumenta e a probabilidade das partículas bombardearem a parede do recipiente, aumenta também (Mohamad, 2011).

2.1.2 Pressão e Temperatura

Considerando uma caixa de comprimento L e dentro dela, uma única partícula movendo-se na direção x com velocidade $\vec{c}_x$ colidindo continuamente com as paredes desta caixa. Assumindo que as colisões são perfeitamente elásticas e integrando a equação 2.1, a força exercida pela partícula em uma parede é igual a taxa de mudança da quantidade de movimento linear

$$F \Delta t = mc_x - (-mc_x) = 2mc_x \quad (2.3)$$

onde Δt é o tempo entre colisões.

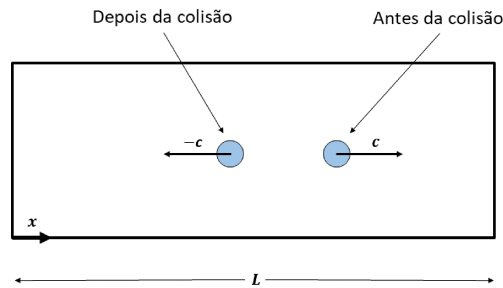


Figura 2.3 – Uma partícula movendo-se livremente dentro de uma caixa.

Observando a Fig. 2.3, tem-se que o tempo entre colisões é igual a $2L/c_x$ e isso indica o tempo necessário que a partícula gasta para percorrer de uma extremidade para outra e retornar ao mesmo local. Assim

$$F = \frac{mc_x^2}{L} \quad (2.4)$$

Para N partículas, tem-se:

$$F = \frac{Nmc^2}{3L} \quad (2.5)$$

onde N é o número de partículas e $c^2 = c_x^2 + c_y^2 + c_z^2$, em que c_x , c_y e c_z são as componentes de velocidade nas direções x , y e z , respectivamente.

A pressão exercida por N partículas nas extremidades da caixa é igual a

$$P = \frac{Nmc^2}{3LA} = \frac{Nmc^2}{3V} \quad (2.6)$$

onde V é volume.

A relação entre pressão e energia cinética das moléculas é, então, dada por

$$P = \frac{2}{3}n_p K_e \quad (2.7)$$

onde n_p é o número de partículas por unidade de volume e $K_e = mc^2/2$ é a energia cinética.

É conhecido que a Lei dos gases ideais clássica é dada por

$$PV = nRT \quad (2.8)$$

em que $n = N/N_A$ é o número de mols, N_A é o número de Avogrado e R é a constante universal dos gases.

Das equações 2.7 e 2.8, tem-se:

$$\frac{NRT}{N_A} = \frac{mc^2}{2} \frac{2}{3} N \quad (2.9)$$

A constante de Boltzmann é,

$$k = \frac{R}{N_A} = 1,38 \times 10^{-23} J/K$$

Assim, a energia cinética K_e de um gás é dada por

$$K_e = \frac{mc^2}{2} = \frac{3}{2} kT \quad (2.10)$$

Retomando que temperatura e pressão no mundo macroscópico são referidas pela energia cinética no mundo microscópico.

2.1.3 Função Distribuição

Em 1859, Maxwell (1831-1879) notou que lidar com um grande número de partículas é difícil de formular, mesmo que a equação governante (segunda lei de Newton) seja conhecida. Assim, deu-se a ideia de média (Mohamad, 2011).

Logo, o conhecimento da posição e da velocidade de cada partícula em cada instante do tempo não é significativo e a função distribuição é o parâmetro fundamental para caracterizar o efeito das partículas. As partículas de um gás têm uma ampla gama de velocidades que colidem umas com as outras a todo instante, por isso as partículas rápidas transferem o impulso para as partículas lentas. Como resultado da colisão, o impulso é conservado. Considerando um gás em equilíbrio térmico uniformemente distribuído em um recipiente, a função distribuição não depende do tempo, assim o único desconhecido é a função distribuição de velocidade, já que o gás se encontra uniformemente distribuído no recipiente.

Para um gás com N partículas, o número de partículas com velocidades na direção x entre c_x e $c_x + dc_x$ é $Nf(c_x)dc_x$. Dessa forma, a função $f(c_x)$ representa a parcela das partículas com velocidades no intervalo entre c_x e $c_x + dc_x$, na direção x . De forma similar, para as direções y e z , a função distribuição de probabilidade pode ser definida como $f(c_y)$ e $f(c_z)$, respectivamente.

Logo, a probabilidade da velocidade existir entre c_x e $c_x + dc_x$, c_y e $c_y + dc_y$, e c_z e $c_z + dc_z$ será,

$$Nf(c_x)f(c_y)f(c_z)dc_xdc_ydc_z$$

Integrando a equação acima sobre todos os valores possíveis das velocidades, tem-se a probabilidade total da existência em toda a faixa de velocidades, que é por definição 1 (100%), que multiplicado por N , tem-se N

$$\int \int \int f(c_x)f(c_y)f(c_z)dc_xdc_ydc_z = 1 \quad (2.11)$$

Salienta-se que, em qualquer direção x , y ou z , a função distribuição não deve depender da

direção, mas apenas da velocidade das partículas. Portanto

$$f(c_x)f(c_y)f(c_z) = \Phi(c_x^2 + c_y^2 + c_z^2) \quad (2.12)$$

onde Φ é uma função desconhecida a ser determinada.

O valor da função distribuição é positivo e está entre zero e a unidade. Na equação 2.12, a velocidade é elevada ao quadrado justamente para evitar valores negativos. Uma função plausível que possui propriedade da equação 2.12 é a função logarítmica ou exponencial

$$\log A + \log B = \log(AB)$$

ou

$$e^A e^B = e^{(A+B)}$$

A equação apropriada para a função distribuição é,

$$f(c_x) = A e^{-B c_x^2} \quad (2.13)$$

onde A e B são constantes.

De forma direta, tem-se a função distribuição que dá o número de partículas com velocidade entre c e $c + dc$

$$f(c) = A e^{-B c_x^2} A e^{-B c_y^2} A e^{-B c_z^2} = A^3 e^{-B c^2} \quad (2.14)$$

A distribuição de partículas no espaço de velocidade (c_x , c_y e c_z), cada partícula é representada por um ponto com coordenadas correspondentes à velocidade. Logo, supondo uma esfera centrada na origem, todos os pontos situados sobre sua superfície correspondem à mesma velocidade. Portanto, o número de partículas com velocidade entre c e $c + dc$ é imediatamente igual ao número de pontos situados entre duas cascas da esfera, com raios c e $c + dc$, conforme Fig. 2.4.

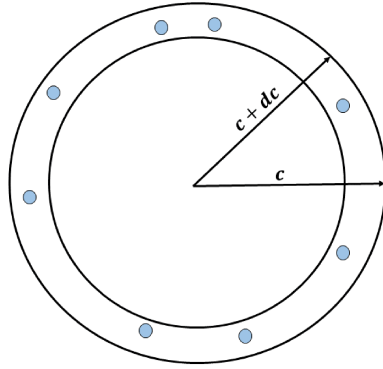


Figura 2.4 – Vista em corte de duas esferas com raios c e $c + dc$.

Da figura 2.4 tem-se,

- Área da superfície esférica de raio c : $A = 4\pi c^2$;
- Espessura entre as duas cascas de esfera: $h = dc$;

Logo, o volume (V) é $V = Ah = 4\pi c^2 dc$.

Portanto, a distribuição de probabilidade em função da velocidade é

$$f(c)dc = 4\pi c^2 A^3 e^{-Bc^2} dc \quad (2.15)$$

Integrando esta equação, obtém-se o total de 8 partículas. E integrando a função distribuição de probabilidade em todas as velocidades para encontrar o número total de partículas N e a energia total E encontra-se assim também, as constantes A e B.

Uma partícula movendo-se com velocidade c possui energia cinética,

$$K_e = \frac{1}{2}mc^2$$

Usando a função distribuição de probabilidade, a energia cinética média por partícula, é:

$$\overline{\frac{1}{2}mc^2} = \frac{\int_0^{\infty} \frac{1}{2}mc^2 f(c)dc}{\int_0^{\infty} f(c)dc} \quad (2.16)$$

O numerador da equação 2.16 representa a energia total e o denominador equivale ao número total de partículas. Nota-se que a constante A é cancelada entre o numerador e o denominador. Logo, substituindo nas integrais o valor de $f(c)$ dado pela equação 2.14, tem-se

$$\overline{\frac{1}{2}mc^2} = \frac{3m}{4B}$$

E substituindo a equação 2.10 que corresponde ao valor da energia cinética média em termos de temperatura do gás,

$$\overline{\frac{1}{2}mc^2} = \frac{3}{2}kT$$

Assim, $B = \frac{m}{2kT}$. Portanto,

$$f(c) \propto c^2 e^{\frac{-mc^2}{2kT}} \quad (2.17)$$

Em seguida, integrando sobre todas as velocidades e fazendo o resultado igual a um, determina-se a constante de proporcionalidade

$$f(c) = 4\pi \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} c^2 e^{\frac{-mc^2}{2kT}} \quad (2.18)$$

Observa-se que essa função cresce parabolicamente de zero, atinge um valor máximo e depois decresce exponencialmente. À medida que a temperatura aumenta, a posição do máximo desloca-se para a direita, um exemplo deste comportamento pode ser visto na Fig. 2.5. A área total sob a curva é sempre igual a unidade, por definição. Esta equação é denominada função distribuição de Maxwell ou Maxwell-Boltzmann.

Percebe-se que a probabilidade de encontrar uma partícula com uma velocidade específica é nula e isso se deve as colisões que alteram a velocidade das partículas continuamente. Assim, o crucial é encontrar a probabilidade das partículas estarem dentro de uma faixa de velocidades e não com uma velocidade específica. E para isto, a equação 2.18, deve ser integrada no intervalo de velocidades desejado.

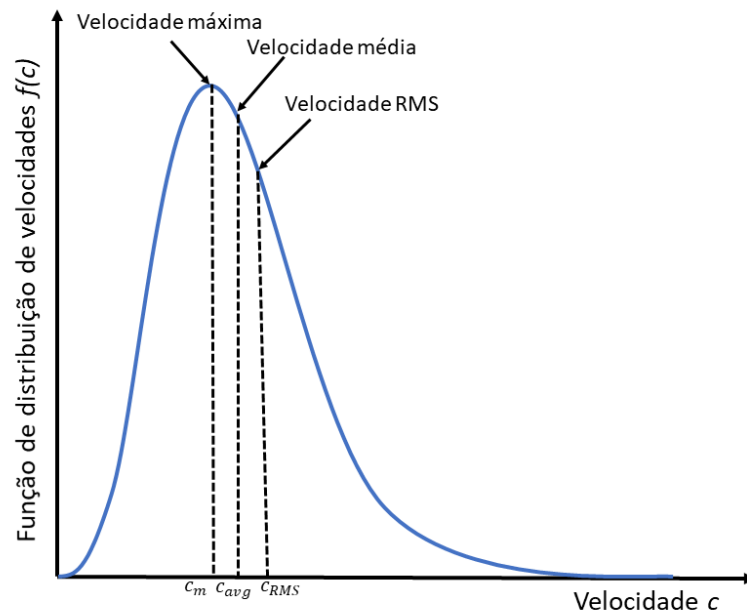


Figura 2.5 – Função distribuição de Maxwell-Boltzmann.

A figura acima mostra o comportamento da função distribuição de probabilidade em função da velocidade molecular (c). À medida que T cresce, o grau de agitação das partículas aumenta, o que leva a um crescimento no número de partículas com alta velocidade. Assim, quanto maior a temperatura mais a direita e mais achatada é a curva da distribuição de Maxwell-Boltzmann.

A velocidade mais provável é a velocidade referente ao ponto de máximo do gráfico, e para encontrá-la basta fazer $f'(c) = 0$, logo

$$c_m = \sqrt{\frac{2kT}{m}}$$

Outras velocidades importantes no estudo da distribuição de Maxwell-Boltzmann são:

- Velocidade média: $c_{avg} = \int_0^{\infty} cf(c)dc = \sqrt{\frac{8kT}{\pi m}}$;
- Velocidade média ao quadrado: $c_{avg}^2 = \int_0^{\infty} c^2 f(c)dc = \frac{3kT}{m}$;
- Raiz da velocidade quadrática média (RMS): $c_{RMS} = \sqrt{\langle c^2 \rangle} = \sqrt{\frac{3kT}{m}}$.

Essas equações podem ser aplicadas a qualquer elemento químico gasoso. Vale ressaltar que, a velocidade média das partículas do gás se encontra a direita do pico, devido à "cauda" maior no lado direito do gráfico de distribuição de Maxwell-Boltzmann. Outro ponto importante diz respeito à raiz da velocidade quadrática média que nada mais é do que a raiz quadrada da média dos quadrados das velocidades vetoriais. Logo, percebe-se que ela eleva todas as velocidades ao quadrado para depois tirar a raiz quadrada e isso se dá ao fato de que a velocidade vetorial média das partículas de gás é nula, pois há partículas de gás se movendo para a direita e outras para a esquerda. Assim, primeiro eleva-se as velocidades vetoriais ao quadrado, tornando-as positivas e isso garante que o valor médio não resulte em zero. Tal truque é bastante utilizado quando os valores médios de determinadas grandezas assumem valores positivos e negativos, como por exemplo, a tensão e a corrente em um circuito de corrente alternada.

Normalmente, os gases mais leves, ou seja, aqueles que possuem menor massa molecular, têm velocidades moleculares mais elevadas que os gases pesados (maior massa molecular) (Mohamad, 2011).

2.1.4 Distribuição de Boltzmann

Boltzmann generalizou a distribuição de Maxwell para grandes sistemas arbitrários. Ele foi o primeiro a perceber a conexão entre o conceito termodinâmico de entropia e a análise estatística de possíveis estados de um sistema grande. A distribuição de Boltzmann descreve como uma dada energia distribui-se ao longo de um grande número de partículas em um dado sistema clássico em equilíbrio térmico com volume e número de partículas determinados.

Boltzmann provou que a entropia termodinâmica S , de um sistema, em uma dada energia E está relacionada ao número de estados microscópicos W disponíveis por ele, ou seja, $S = k \log(W)$, em que k é a constante de Boltzmann. Assim, Boltzmann pôde estabelecer que, para qualquer sistema em equilíbrio térmico à temperatura T , a probabilidade de estar em um estado particular com energia E é proporcional a $e^{\frac{-E}{kT}}$, ou seja

$$f(E) = A e^{\frac{-E}{kT}} \quad (2.19)$$

A equação 2.19 é chamada de Distribuição de Boltzmann.

Considerando a energia cinética das partículas na direção x ,

$$E = \frac{1}{2}mc_x^2$$

A função de probabilidade integrada para todos os valores de velocidade deve ser igual a unidade.

Logo,

$$\int_{-\infty}^{+\infty} Ae^{\frac{-mc_x^2}{2kT}} dc = 1 \quad (2.20)$$

Assim,

$$A = \sqrt{\frac{m}{2\pi kT}} \quad (2.21)$$

A probabilidade de encontrar velocidade c é

$$f(c) = \left[\sqrt{\frac{m}{2\pi kT}} \right]^3 e^{\frac{-mc^2}{2kT}} \quad (2.22)$$

onde $c = c_x^2 + c_y^2 + c_z^2$.

Multiplicando-se um fator de $4\pi c^2$ na equação acima, obtém-se a distribuição de Maxwell-Boltzmann dada pela equação 2.18. Logo, a integração da função distribuição de Maxwell-Boltzmann sobre uma superfície esférica resulta na equação 2.22.

Um gás ideal possui uma função distribuição específica em equilíbrio, ou seja, função distribuição de Maxwell, mas, à priori não se tinha conhecimento de como esse equilíbrio seria atingido. E esta foi uma das contribuições revolucionárias de Boltzmann, que é a base do LBM (Mohamad, 2011). Faz-se necessário então, discorrer sobre a equação de transporte de Boltzmann.

2.2 A Equação de transporte de Boltzmann

O físico austríaco Ludwig Eduard Boltzmann (1844-1906) teve sua maior contribuição no ramo da termodinâmica estatística e ficou conhecido como um dos fundadores da mecânica estatística. Entre inúmeras de suas contribuições, tem-se a função distribuição de Boltzmann,

tópico já estudado anteriormente, mas vale lembrar que, a função distribuição que dá a probabilidade de encontrar partículas dentro de um certo intervalo de velocidades e posição em um determinado instante substitui o estudo de cada partícula individual, como em simulações de dinâmicas moleculares. Com isso, o LBM economiza recursos computacionais.

Assim, considerando um sistema que pode ser descrito estatisticamente por uma função de distribuição $f(r, c, t)$, que representa o número de partículas no instante t entre r e $r + dr$, as quais têm velocidades entre c e $c + dc$. Uma força externa F agindo sobre uma partícula de unidade de massa altera a sua velocidade de c para $c + Fdt$, e sua posição de r para $r + cdt$, como mostra a Fig. 2.6.

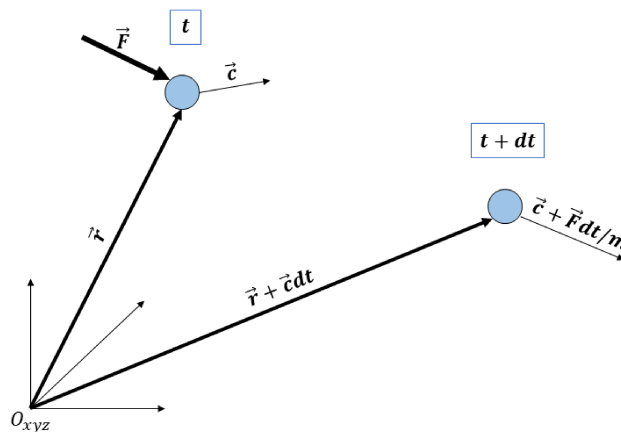


Figura 2.6 – Vetor posição e velocidade para uma partícula antes e após a aplicação de uma força F .

O número de partículas $f(r, c, t)$ antes da aplicação da força externa é igual ao número de partículas após a perturbação, $f(r + cdt, c + Fdt, t + dt)$, considerando não haver colisões entre as partículas. Entretanto, se ocorrem colisões entre as partículas há, então, uma diferença líquida entre o número de partículas no intervalo $drdc$. E a taxa de alteração entre os estados final e inicial da função distribuição é chamado de operador de colisão, Ω . Portanto, a equação que dá a evolução do número de partículas pode ser escrita como

$$f(r + cdt, c + Fdt, t + dt)drdc - f(r, c, t)drdc = \Omega(f)drdc dt \quad (2.23)$$

Dividindo a equação acima por $dt dr dc$ e fazendo $dt \rightarrow 0$, tem-se

$$\frac{df}{dt} = \Omega(f) \quad (2.24)$$

A equação acima mostra que a taxa total de alteração da função distribuição é igual a taxa de colisão. Lembrando que f é uma função de r , c e t , logo, a taxa total de alteração pode ser expandida como

$$df = \frac{\partial f}{\partial r} dr + \frac{\partial f}{\partial c} dc + \frac{\partial f}{\partial t} dt \quad (2.25)$$

E dividindo por dt , tem-se

$$\frac{df}{dt} = \frac{\partial f}{\partial r} c + \frac{\partial f}{\partial c} a + \frac{\partial f}{\partial t} \quad (2.26)$$

onde a aceleração $a = dc/dt$ e pode ser relacionada à força F pela segunda lei de Newton, $a = F/m$. O vetor r pode ser expresso em um sistema de coordenadas cartesianas 3D como $r = xi+yj+zk$, onde i , j e k são vetores unitários ao longo das direções x , y e z , respectivamente.

Portanto, a equação de transporte de Boltzmann (2.24) pode ser reescrita como

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial r} \cdot c + \frac{F}{m} \cdot \frac{\partial f}{\partial c} = \Omega \quad (2.27)$$

Para resolver a equação de Boltzmann é necessário determinar Ω que é uma função que depende de f .

Considerando um sistema sem atuação de forças externas, a equação de Boltzmann pode ser escrita como

$$\frac{\partial f}{\partial t} + c \cdot \nabla f = \Omega \quad (2.28)$$

c e ∇f são vetores.

A equação 2.24 é uma equação de advecção com um termo fonte Ω . Existem dificuldades na solução desta equação devido ao fato de que Ω é função de f .

A relação entre a equação acima e as quantidades macroscópicas tais como densidade do fluido, ρ , vetor velocidade do fluido, $\vec{u}$, e energia interna, e , são:

$$\rho(r,t) = \int M f(r,c,t) dc \quad (2.29)$$

$$\rho(r,t)u(r,t) = \int M c f(r,c,t) dc \quad (2.30)$$

$$\rho(r,t)e(r,t) = \frac{1}{2} \int M u_a^2 f(r,c,t) dc \quad (2.31)$$

onde M é a massa molecular e $\vec{u}_a$ é a velocidade da partícula relativa à velocidade do fluido, $\vec{u}_a = \vec{c} - \vec{u}$. Nota-se que as equações 3.1, 3.2 e 2.31 são a conservação de massa, quantidade de movimento e energia, respectivamente.

2.3 A Aproximação BGKW

Conforme mencionado, há uma complexidade na determinação do operador de colisão. Porém, é possível aproximar este operador com um operador simples sem introduzir um erro significativo no resultado da solução. Bhatnagar, Gross e Krook (BGK) em 1954 apresentaram um modelo simplificado para o operador de colisão. Ao mesmo tempo Welander (1954), independentemente, introduziu um operador semelhante (Mohamad, 2011). Assim, o operador de colisão é dado por

$$\Omega = \omega(f^{eq} - f) = \frac{1}{\tau}(f^{eq} - f) \quad (2.32)$$

onde $\omega = 1/\tau$. O coeficiente ω é chamado de frequência de colisão e τ é chamado de tempo de relaxamento. A distribuição de equilíbrio local f^{eq} é uma função distribuição de Maxwell-Boltzmann. Após introduzir esta aproximação, a equação 2.28, pode ser então aproximada por

$$\frac{\partial f}{\partial t} + c \cdot \nabla f = \frac{1}{\tau}(f^{eq} - f) \quad (2.33)$$

No método Lattice Boltzmann, a equação acima é discretizada supondo que seja válida ao longo de direções específicas. Logo, em uma determinada direção tem-se a seguinte equação discreta de Boltzmann

$$\frac{\partial f_i}{\partial t} + c_i \nabla f_i = \frac{1}{\tau}(f_i^{eq} - f_i) \quad (2.34)$$

É possível derivar as equações de Navier-Stokes da equação de Boltzmann acima. Nota-se que a equação 2.34 é uma equação diferencial parcial linear, com o lado esquerdo representando

a advecção, e o lado direito o processo de colisão. A equação 2.34 pode ser discretizada como

$$f_i(r + c_i \Delta t, t + \Delta t) = f_i(r, t) + \frac{\Delta t}{\tau} [f_i^{eq}(r, t) - f_i(r, t)] \quad (2.35)$$

A função de distribuição de equilíbrio local e o tempo de relaxamento determinam o tipo de problema a ser resolvido. Salienta-se a simplicidade dessa equação, podendo ser aplicada a muitos problemas físicos, bastando apenas especificar uma função distribuição de equilíbrio e um termo fonte diferentes (Mohamad, 2011).

É possível utilizar diferenças finitas ou volumes finitos para resolver a equação diferencial parcial 2.34. O crucial é resolver a equação 2.34 em duas etapas: colisão e advecção. Além disso, o domínio da solução precisa ser dividido em *lattices*. Em cada nó de *lattice*, residem as partículas que definem a função de distribuição. Algumas dessas partículas fluem para os nós vizinhos ao longo de direções especificadas. E o número de ligações que dão essas direções é determinado pelo arranjo de *lattice*. Diferentes arranjos de *lattice* são brevemente apresentados abaixo.

2.4 Arranjos de Lattice

A terminologia usual no LBM é fazer referência à dimensão do problema e o número de velocidades usando a expressão $DnQm$, onde n representa a dimensão do problema (1 para 1D, 2 para 2D e 3 para 3D) e m indica o modelo de velocidades, ou seja, o número de ligações (Mohamad, 2011). Assim, seguem alguns modelos de arranjos.

2.4.1 Unidimensional

De forma geral, dois modelos podem ser usados para representar arranjos de *lattice* para problemas em uma dimensão, são eles, D1Q3 e D1Q5, como pode ser visto na Fig. 2.7. As partículas fluem do nó central para os nós vizinhos através das ligações com uma velocidade estabelecida, chamada velocidade de *lattice*.

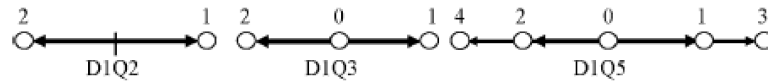


Figura 2.7 – Arranjos de *lattice* para problemas unidimensionais. Fonte: Mohamad (2011).

D1Q3 e D1Q2

Para D1Q3, existem três vetores de velocidade ($c_0 = 0$, $c_1 = \Delta x/\Delta t$ e $c_2 = -\Delta x/\Delta t$) para f_0 , f_1 e f_2 , respectivamente. Δx e Δt representam o comprimento de ligação e o intervalo de tempo, respectivamente. Assumindo $\Delta x = \Delta t$, tem-se $c_0 = 0$, $c_1 = 1$ e $c_2 = -1$. Na etapa de advecção, estas velocidades determinam a posição de cada partícula para o próximo nó. Naturalmente, a partícula central permanece no nó.

Para este arranjo, o número de partículas, em qualquer instante de tempo, não deve exceder o total de três partículas. Uma partícula estagnada, ou seja, com sua velocidade nula reside no nó central. As outras duas partículas se movem para o nó esquerdo ou direito no processo de advecção. Os fatores de ponderação, β_i , assumem valores de $4/6$, $1/6$ e $1/6$ para f_0 , f_1 e f_2 , respectivamente. A velocidade do som, c_s , em unidades de *lattice*, para D1Q3 vale $1/\sqrt{3}$ (Mohamad, 2011).

O arranjo D1Q2, possui fatores de ponderação β_i de $1/2$ e $1/2$ para f_1 e f_2 , respectivamente. A velocidade do som para este arranjo é $1/\sqrt{2}$.

D1Q5

Neste arranjo, o número total de partículas em qualquer instante de tempo não excede cinco partículas. Os fatores de ponderação para este arranjo são $\beta_0 = 6/12$, $\beta_1 = 2/12$, $\beta_2 = 2/12$, $\beta_3 = 1/12$, e $\beta_4 = 1/12$ para f_0 , f_1 , f_2 , f_3 e f_4 , respectivamente. E a velocidade do som em unidades de *lattice* é $1/\sqrt{3}$.

2.4.2 Bidimensional

D2Q5 e D2Q4

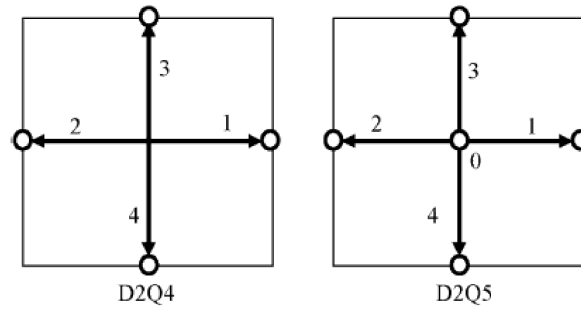


Figura 2.8 – Arranjos de *lattice* para problemas bidimensionais. Fonte: Mohamad (2011).

O modelo D2Q5 possui quatro vetores de velocidade irradiados a partir dos nós centrais, conforme Fig. 2.8. Uma das partículas reside no nó central, portanto, sua velocidade é zero, ou seja, $c(0,0)$. As funções de distribuição f_1 , f_2 , f_3 e f_4 movem-se com velocidades $c(1,0)$, $c(-1,0)$, $c(0,1)$ e $c(0,-1)$ respectivamente. Nota-se que, $\Delta x = \Delta y = \Delta t$. Os fatores de ponderação são $\beta_0 = 2/6$, $\beta_1 = 1/6$, $\beta_2 = 1/6$, $\beta_3 = 1/6$ e $\beta_4 = 1/6$.

O D2Q4 possui quatro vetores de velocidade e não há partículas residindo no nó central. O fator de ponderação para cada direção é $1/4$. A numeração de ligações de *lattice* é arbitrária, porém, é importante escolher a numeração adequada para simplificar o código da simulação.

D2Q9

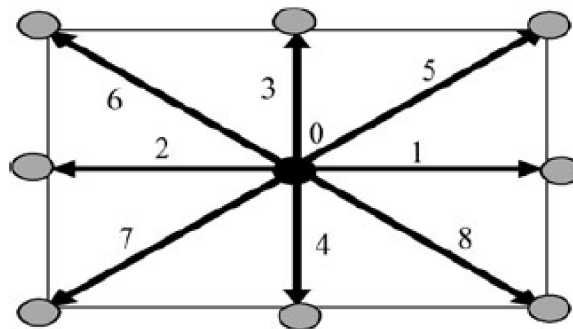


Figura 2.9 – Arranjo de Lattice D2Q9 para problemas bidimensionais. Fonte: Mohamad (2011).

Este modelo pode ser visualizado na Fig. 2.9 e possui vetores de alta velocidade. As velocidades são $c(0, 0)$, $c(1, 0)$, $c(-1, 0)$, $c(0, 1)$, $c(0, -1)$, $c(1, 1)$, $c(-1, 1)$, $c(-1, -1)$ e $c(1, -1)$ para $f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7$ e f_8 , respectivamente. Por fim, os fatores de ponderação para as funções de distribuição correspondentes são $4/9, 1/9, 1/9, 1/9, 1/9, 1/36, 1/36, 1/36$ e $1/36$.

2.4.3 Tridimensional

D3Q15

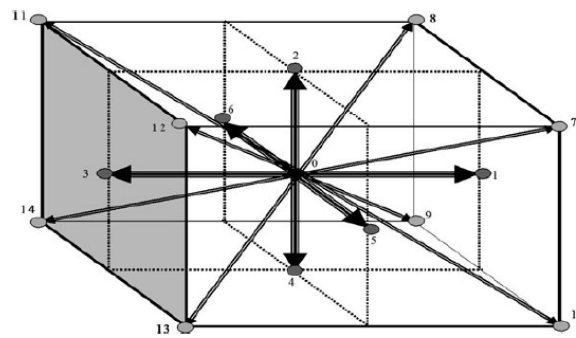


Figura 2.10 – Arranjo de *lattice* D3Q15 para problemas tridimensionais. Fonte: Mohamad (2011).

Este modelo é o mais utilizado em simulações tridimensionais, ele possui 15 vetores de velocidade, conforme Fig. 2.10. Os 15 vetores de velocidade para as funções de distribuição de $f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}$ e f_{14} são $c(0, 0, 0)$, $c(1, 0, 0)$, $c(0, 1, 0)$, $c(-1, 0, 0)$, $c(0, -1, 0)$, $c(0, 0, 1)$, $c(0, 0, -1)$, $c(1, 1, 1)$, $c(1, 1, -1)$, $c(1, -1, -1)$, $c(1, -1, 1)$, $c(-1, 1, -1)$, $c(-1, 1, 1)$, $c(-1, -1, 1)$ e $c(-1, -1, -1)$, respectivamente. Por fim, os fatores de ponderação são $\beta_0 = 16/72$, $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 8/72$ e $\beta_7 = \beta_8 = \beta_9 = \beta_{10} = \beta_{11} = \beta_{12} = \beta_{13} = \beta_{14} = 8/72$.

D3Q19

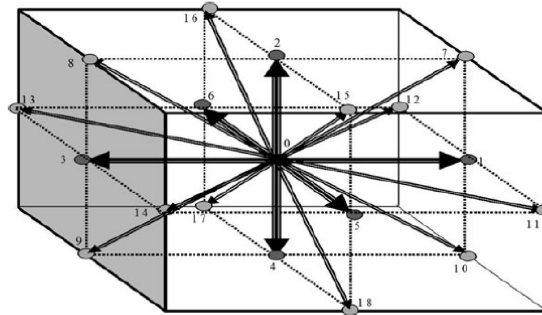


Figura 2.11 – Arranjo de *lattice* D3Q19 para problemas tridimensionais. Fonte: Mohamad (2011).

Este modelo possui 19 vetores de velocidade, com vetor central de velocidade zero, conforme Fig. 2.11. Os fatores de ponderação são, para f_0 é $12/36$, para f_1 até f_6 é $2/36$ e para f_7 até f_{18} é $1/36$. A velocidade do som em unidade de *lattice* corresponde a $c_s = c_i/\sqrt{3}$ e como $\Delta x = \Delta y = \Delta z = \Delta t = 1$, tem-se $c_i = 1$ e por fim, $c_s = 1/\sqrt{3}$.

2.5 Função de Distribuição de Equilíbrio

Conforme mencionado anteriormente, a aplicação do LBM para diversos problemas depende da determinação da função distribuição de equilíbrio f^{eq} apropriada.

Para partículas que se movem em um meio com velocidade macroscópica $\vec{u}$. A função distribuição de Maxwell, equação 2.22 pode ser reescrita como

$$f = \frac{\rho}{2\pi/3} e^{\frac{-3}{2}(\vec{c}-\vec{u})^2} \quad (2.36)$$

Ou então,

$$f = \frac{\rho}{2\pi/3} e^{\frac{-3}{2}c^2} e^{3(\vec{c}\cdot\vec{u}-u^2)/2} \quad (2.37)$$

onde $c^2 = \vec{c} \cdot \vec{c}$ e $u^2 = \vec{u} \cdot \vec{u}$.

Sabe-se que a expansão em série de Taylor para e^{-x} é,

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{3!} \dots$$

Assim, a equação 2.37 pode ser expandida como

$$f = \frac{\rho}{2\pi/3} e^{\frac{-3}{2}c^2} \left[1 + 3(\vec{c} \cdot \vec{u}) - \frac{3}{2}u^2 + \dots \right] \quad (2.38)$$

A forma geral da função distribuição de equilíbrio é,

$$f_i^{eq} = \lambda \omega_i \left[A + B \vec{c}_i \cdot \vec{u} + C (\vec{c}_i \cdot \vec{u})^2 + D u^2 \right] \quad (2.39)$$

onde $\vec{u}$ é o vetor de velocidade do fluido, A, B, C e D são constantes a serem determinadas com base no princípio de conservação de massa, momento e energia e λ é um parâmetro escalar, como densidade (ρ) e temperatura. E ela é igual à soma de todas as funções de distribuição

$$\lambda = \sum_i^m f_i^{eq} \quad (2.40)$$

onde m é o número de ligações de *lattice*. Nota-se que quando se trata de um arranjo de *lattice* que possui nó central ($c = 0$), a soma da equação 2.40 inicia com $i = 0$, do contrário, a soma inicia em $i = 1$, conforme os índices das ligações de *lattice*

Para melhor compreensão do método, a seguir é aplicado o LBM e Diferenças finitas com o propósito de comparar a solução com os dois métodos a um exemplo simples de difusão de calor em uma placa infinita submetida a uma temperatura constante com condições de contorno de Dirichlet.

2.6 Difusão de Calor em uma placa infinita submetida a uma temperatura constante

2.6.1 Equação da Difusão

A equação da difusão unidimensional é dada por

$$\frac{\partial \Psi}{\partial t} = \alpha \frac{\partial^2 \Psi}{\partial x^2} \quad (2.41)$$

onde Ψ é a variável dependente que difunde-se em ambas direções em um meio infinito. A taxa de difusão depende do parâmetro α , em que, dependendo do problema, α pode ser coeficiente de

difusão térmica, coeficiente de difusão de massa ou viscosidade cinemática, para energia, massa e difusão de momento, respectivamente. O processo de difusão se torna mais rápido à medida que o parâmetro α aumenta (Mohamad, 2011).

Fazendo uma análise de ordem de grandeza, tem-se

$$\frac{1}{\tau} \approx \alpha \frac{1}{l^2} \implies l \approx \sqrt{\alpha \tau}$$

onde τ e l são escalas de tempo e distância, respectivamente.

A equação 2.41 tem segunda derivada no espaço, portanto, a difusão ocorre em ambos os sentidos, o que requer duas condições de contorno. Além disso, a difusão em qualquer ponto depende do tempo anterior, não sendo necessário informações do tempo futuro. Para resolver essa equação deve-se impor uma condição inicial.

Lembrando que na equação 2.41, Ψ é a variável dependente que decorre do problema de interesse a ser resolvido. Assim, dois exemplos podem ser citados:

1 A equação da difusão de calor unidimensional:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (2.42)$$

onde T é a temperatura, α é uma propriedade do material chamada de difusividade térmica, dada por $\alpha = k_T / \rho C$, em que k_T , ρ e C são condutividade térmica do meio, densidade e calor específico, respectivamente.

2 A equação da difusão do momento unidimensional:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \quad (2.43)$$

onde u é velocidade e ν é a viscosidade cinemática do fluido.

2.6.2 Função Distribuição de Equilíbrio

Para o problema de difusão, é conveniente que as funções de distribuição de equilíbrio sejam consideradas constantes, onde nenhuma velocidade macroscópica está envolvida (Moha-

mad, 2011). Então,

$$f_i^{eq} = A_i$$

Utilizando a forma geral da função distribuição de equilíbrio dada pela equação 2.40 e aplicando a conservação de massa,

$$\sum_{i=1}^2 f_i^{eq} = \lambda$$

e momento,

$$\sum_{i=1}^2 f_i^{eq} c_i = 0$$

Logo,

$$\sum_{i=1}^2 A_i = A_1 + A_2 = \lambda$$

Além disso,

$$\sum_{i=1}^2 A_i c_i = A_1 c_1 + A_2 c_2 = A_1 - A_2 = 0$$

onde $c_1 = 1$ e $c_2 = -1$. Consequentemente, $A_1 = A_2 = \frac{\lambda}{2}$. De forma geral,

$$A_i = \beta_i \lambda$$

Portanto,

$$f_i^{eq} = \beta_i \lambda \quad (2.44)$$

2.6.3 Expansão Chapman-Enskog

A equação da difusão unidimensional em coordenadas cartesianas pode ser escrita como

$$\frac{\partial T(x, t)}{\partial t} = \Gamma \frac{\epsilon^2 \partial^2 T(x, t)}{\partial x^2} \quad (2.45)$$

onde $T(x, t)$ representa temperatura, Γ é o coeficiente de difusão e é um valor constante e ϵ é o número de Knudson.

Para um problema unidimensional com duas componentes de velocidade, c_i , $i = 1, 2$,

a temperatura pode ser expressa como,

$$T(x, t) = \sum_{i=1}^{i=2} f_i(x, t) = f_1(x, t) + f_2(x, t) \quad (2.46)$$

A fim de simplificar, usa-se a notação f_i no lugar de $f_i(x, t)$. A equação de difusão é linear, então

$$f_i^{eq} = \beta_i T(x, t) \quad (2.47)$$

Aplicando somatório dos dois lados da equação, tem-se

$$\sum_i f_i^{eq} = T = \beta_1 T + \beta_2 T \quad (2.48)$$

onde $\beta_1 + \beta_2 = 1$, ou seja, $\beta_1 = \beta_2 = 1/2$, então,

$$f_1^{eq} = \beta_1 T(x, t) \text{ e } f_2^{eq} = \beta_2 T(x, t) \quad (2.49)$$

Expandindo a função de distribuição em termos de um parâmetro ϵ ,

$$f_i(x, t) = f_i^0 + \epsilon f_i^1 + \epsilon^2 f_i^2 + \dots \quad (2.50)$$

onde f_i^0 é a função distribuição nas condições de equilíbrio, assim como f_i^{eq} . Somando a equação acima,

$$\sum_{i=1}^2 f_i(x, t) = \sum_{i=1}^2 [f_i^0 + \epsilon f_i^1 + \epsilon^2 f_i^2 + \dots] \quad (2.51)$$

Sabe-se que $f_1 + f_2 = T(x, t)$ e $f_1^0 + f_2^0 = T(x, t)$, isso implica que os outros termos expandidos da equação acima são nulos, isto é

$$\sum_{i=1}^2 f_i^1 = 0 \quad (2.52)$$

$$\sum_{i=1}^2 f_i^2 = 0 \quad (2.53)$$

A nova função distribuição $f_i(x + c_i \Delta x, t + \Delta t)$ é expandida utilizando a série de Taylor e assim

a equação de Lattice Boltzmann pode ser reescrita como,

$$f_i(x + c_i \Delta x, t + \Delta t) = f_i(x, t) + \frac{\Delta t}{\tau} [f_i^0(x, t) - f_i(x, t)] \quad (2.54)$$

E através de algumas manipulações matemáticas, tem-se

$$0 = \frac{\partial \Theta}{\partial t} - \left(\tau - \frac{\Delta t}{2} \right) \frac{\partial^2 \Theta}{\partial x^2} \quad (2.55)$$

Comparando as equações 2.45 e 2.55, é possível estabelecer uma relação entre macroescala e mesoescala, ou seja,

$$\Gamma = \tau - \frac{\Delta t}{2} \quad (2.56)$$

Normalizando e adimensionalizando

É sensato resolver problemas específicos normalizados, pois permitem que os parâmetros de controle possam ser agrupados em poucos parâmetros adimensionais, portanto, os resultados podem ser apresentados genericamente. Além disso, o escalonamento macroescala para a mesoescala torna-se claro (Mohamad, 2011).

Adimensionalizando a temperatura

$$\theta = \frac{\lambda - \lambda_c}{\lambda_h - \lambda_c}$$

onde λ_h e λ_c são as temperaturas quente e fria, respectivamente. O x pode ser dimensionado pelo comprimento do sistema L , ou seja, adimensionalizado $x^* = x/L$. Assim, a equação da difusão de calor adimensionalizada pode ser escrita como

$$\frac{\partial \theta}{\partial t^*} = \frac{\partial^2 \theta}{\partial x^{*2}} \quad (2.57)$$

onde $t^* = \alpha t / L^2$ é o tempo adimensional.

2.6.4 Modelagem com condição de contorno de Dirichlet

Considerando uma placa com temperatura inicial, $T = 0$. Para o tempo $t \geq 0$, a extremidade esquerda da placa possui temperatura elevada e igual à unidade, $T = 1,0$. O

comprimento da placa é de 100 unidades. A Fig. 2.12 mostra um esquema da mesma, a simulação é unidimensional, assim o comprimento vertical da figura não está em escala. A distribuição de temperatura para $t = 200$ foi encontrada utilizando LBM e FDM, conforme a seguir.



Figura 2.12 – Placa com comprimento de 100 unidades e temperatura $T = 1$ na extremidade esquerda.

Aproximação por Diferenças Finitas

Tomando a equação 2.42, a primeira etapa é discretizar o domínio e para simplificar, a discretização foi feita em segmentos iguais conforme Fig. 2.13. Os nós são classificados como $0, 1, 2, 3, \dots, k-2, k-1, k, k+1, k+2, \dots, n$.

Como a placa é de 100 unidades, o domínio espacial foi discretizado em 100 nós, logo a distância entre dois nós é $\Delta x = 1$. O domínio do tempo também foi discretizado em $\Delta t = 0,5$.

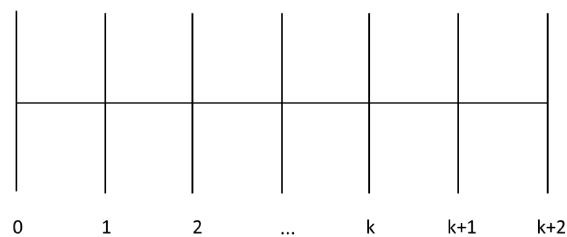


Figura 2.13 – Nós distribuídos unidimensionalmente na abordagem de Diferenças Finitas.

Utilizando diferenças centradas no espaço e avanço no tempo em um nó i , é possível obter

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = \alpha \frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{\Delta x^2} \quad (2.58)$$

onde $n + 1$ representa o tempo atual e n o tempo anterior. Assim, reforça-se que há apenas necessidade de ter informações do tempo anterior para a solução da equação. Rearranjando, tem-se

$$T_k^{n+1} = T_k^n + \frac{\alpha \Delta t}{\Delta x^2} (T_{k+1}^n - 2T_k^n + T_{k-1}^n) \quad (2.59)$$

Reorganizando a equação,

$$T_k^{n+1} = T_k^n \left(1 - \frac{2\alpha \Delta t}{\Delta x^2} \right) + \frac{2\alpha \Delta t}{\Delta x^2} \left(\frac{T_{k+1}^n + T_{k-1}^n}{2} \right) \quad (2.60)$$

E fazendo $\omega = \frac{2\alpha \Delta t}{\Delta x^2}$, a equação 2.60 fica

$$T_k^{n+1} = T_k^n (1 - \omega) + \omega (0.5T_{k+1}^n + 0.5T_{k-1}^n) \quad (2.61)$$

A fim de garantir estabilidade, os coeficientes dos termos do lado direito devem ser positivos. Assim, o termo $(1 - \omega)$ deve ser maior ou igual a zero, o que implica que $\Delta t \leq \frac{\Delta x^2}{2\alpha}$.

O último termo da equação 2.61 é uma média de temperaturas em torno de T_k . Como T_k está a meia distância de T_{k+1} e T_{k-1} , então, no estado de equilíbrio, o valor de T_k deve ser a média dos valores vizinhos. Fazendo $T_k^{eq} = 0.5[T_{k+1}^n + T_{k-1}^n]$, a equação 2.61 pode ser reescrita como

$$T_k^{n+1} = T_k^n (1 - \omega) + \omega T_k^{eq} \quad (2.62)$$

O código da simulação usando o software *MATLAB*® pode ser visualizado no Apêndice I juntamente com a implementação do mesmo problema utilizando o Método Lattice Boltzmann. Abaixo é detalhada a solução com o LBM.

Solução com o Método Lattice Boltzmann

A equação cinética para a função distribuição $f_i(x, t)$ pode ser escrita como

$$\frac{\partial f_i(x, t)}{\partial t} + c_i \frac{\partial f_i(x, t)}{\partial x} = \Omega_i \quad (2.63)$$

para um problema unidimensional D1Q2, $i = 1, 2$, onde i representa o número de ligações. Ressalta-se que como o arranjo de *lattice* utilizado não possui nó central com velocidade nula,

o índice i inicia em 1. Os termos do lado esquerdo descrevem o processo de advecção, onde a função distribuição flui ao longo das ligações de *lattice* com velocidade

$$c_i = \frac{\Delta x}{\Delta t} \quad (2.64)$$

E o termo à direita, Ω_i , caracteriza a taxa de mudança da função distribuição, f_i , no processo de colisão, ou seja, o operador de colisão para a ligação i . Pela aproximação BGK, tem-se

$$\Omega_i = -\frac{1}{\tau} [f_i(x, t) - f_i^{eq}(x, t)] \quad (2.65)$$

τ está associado ao coeficiente de difusão na escala macroscópica, representando um tempo de relaxamento para a distribuição de equilíbrio (f_i^{eq}).

A equação 2.63 com aproximação BGK pode ser discretizada como

$$\frac{f_i(x, t + \Delta t) - f_i(x, t)}{\Delta t} + c_i \frac{f_i(x + \Delta x, t + \Delta t) - f_i(x, t + \Delta t)}{\Delta x} = -\frac{1}{\tau} [f_i(x, t) - f_i^{eq}(x, t)] \quad (2.66)$$

Com 2.64 e 2.66, tem-se

$$f_i(x + \Delta x, t + \Delta t) - f_i(x, t) = \frac{\Delta t}{\tau} [f_i(x, t) - f_i^{eq}(x, t)] \quad (2.67)$$

Nota-se que a equação acima é semelhante à equação de diferenças finitas 2.62. E reorganizando-a, tem-se

$$f_i(x + \Delta x, t + \Delta t) = f_i(x, t)[1 - \omega] + \omega f_i^{eq}(x, t) \quad (2.68)$$

onde $\omega = \frac{\Delta t}{\tau}$ e representa a frequência de colisão.

A Fig. 2.14 apresenta um diagrama esquemático para três nós com ligações necessárias, onde $c_1 = c$, $c_2 = -c$. Já a Fig. 2.15 mostra a distribuição dos *lattices* para um problema de difusão unidimensional, D1Q2.

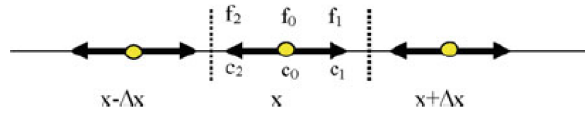


Figura 2.14 – Diagrama esquemático para arranjo de *lattice* unidimensional. Fonte: Mohamad (2011).

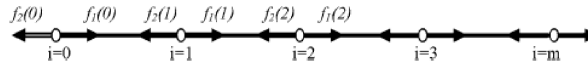


Figura 2.15 – *Lattices* para um problema de difusão unidimensional. Fonte: Mohamad (2011).

A variável dependente Ψ na equação 2.41 pode ser relacionado à função distribuição f_i , como segue

$$\Psi(x, t) = \sum_{i=1}^2 f_i(x, t) \quad (2.69)$$

A função distribuição de equilíbrio f_i^{eq} , pode ser escrita como

$$f_i^{eq} = \beta_i \Psi(x, t) \quad (2.70)$$

onde β_i representa o fator de ponderação na direção i . O fator de ponderação deve satisfazer o seguinte critério,

$$\sum_{i=1}^2 \beta_i = 1 \quad (2.71)$$

A função distribuição f_i^{eq} definida na equação 2.70 pode ser somada ao longo de todas as direções e, assim,

$$\sum_{i=1}^2 f_i^{eq} = \sum_{i=1}^2 \beta_i \Psi(x, t) = \Psi(x, t) \quad (2.72)$$

Utilizando a expansão de Chapman-Enskog para a relação entre α e ω , tem-se

$$\alpha = \frac{\Delta x^2}{\Delta t D} \left(\frac{1}{\omega} - \frac{1}{2} \right) \quad (2.73)$$

onde D representa a dimensão do problema.

No LBM, o domínio também precisa discretizado, contudo, o mais utilizado é a unidade

para Δx e para Δt , ou seja, $\Delta x = \Delta t = 1$. Logo, discretizando o domínio da integração em $\Delta x = \Delta t = 1$. Assumindo $\alpha = 0,25$ e utilizando a equação 2.73, tem-se $\omega = 1,333333$.

Da equação 2.69, tem-se

$$T(x, t) = \sum_{i=1}^2 f_i(x, t) \quad (2.74)$$

E da equação 2.70, tem-se $f_1^{eq}(x, t) = 0,5T(x, t)$ e $f_2^{eq}(x, t) = 0,5T(x, t)$.

O processo de colisão é dado por:

$$f_i(x, t + \Delta t) = f_i(x, t)[1 - \omega] + \omega f_i^{eq}(x, t)$$

onde $i = 1, 2$.

E o processo de advecção é dado por:

$$f_i(x + \Delta x, t + \Delta t) = f_i(x, t + \Delta t)$$

Sabendo-se que no LBM, as partículas em cada *lattice* possuem velocidades discretas, para este caso, $c_1 = -1$ e $c_2 = 1$. Logo, a cada passo de tempo $\Delta t = 1$, as partículas são deslocadas em uma unidade de *lattice*, para esquerda, caso a velocidade seja negativa ou para a direita, caso a velocidade seja positiva.

Deste modo, no processo de advecção, para evitar que os valores da distribuição de probabilidade, f_i sejam sobrescritos, deve-se varrer f_1 para a esquerda ($i = m, 0, -1$) e f_2 para a direita ($i = 0, m$), conforme Fig. 2.16.

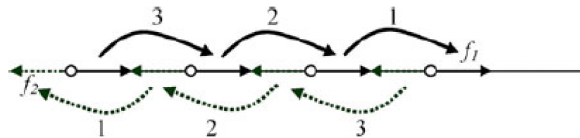


Figura 2.16 – Esquema para a etapa de advecção. Fonte: Mohamad (2011).

No contorno da esquerda, em $x = 0$, o valor de f_2 é obtido a partir do processo de advecção e f_1 é desconhecido. No contorno direito, $x = L$, f_1 é obtido do processo de advecção.

Para D1Q2 em $x = 0$, tem-se

$$f_1^{eq}(0, t) - f_1(0, t) + f_2^{eq}(0, t) - f_2(0, t) = 0 \quad (2.75)$$

e

$$f_1^{eq}(0, t) = \beta_1 T = 0,5T \text{ e } f_2^{eq}(0, t) = \beta_2 T = 0,5T$$

Portanto, em $x = 0$, $f_1(0) + f_2(0) = T$, e das etapas de advecção, $f_2(0) = f_2(1)$, então $f_1(0) = T - f_2(0)$. A Fig. 2.17 mostra a comparação dos resultados obtidos através do LBM e FDM para uma placa com condições de contorno de temperatura constante.

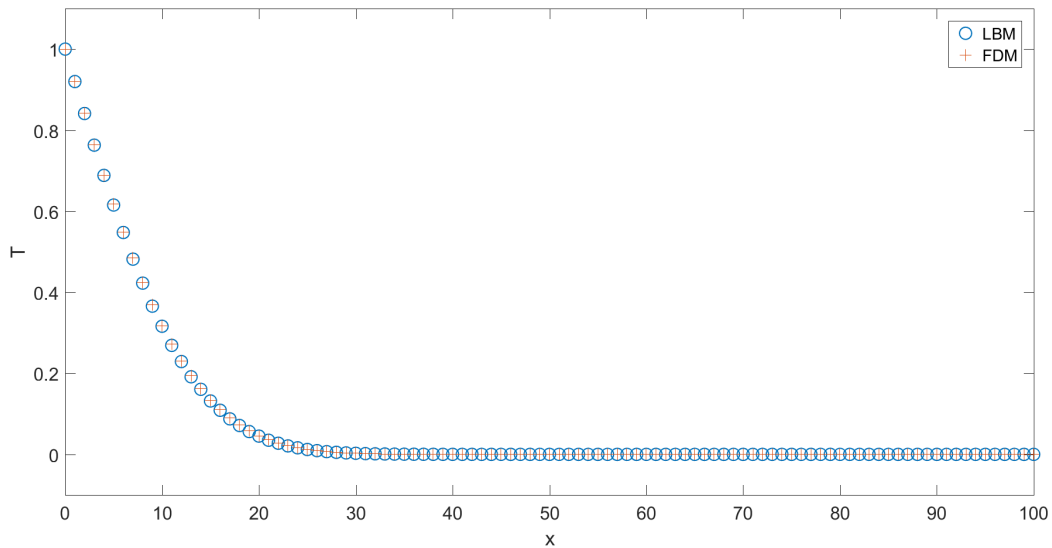


Figura 2.17 – Comparação dos resultados obtidos através do LBM e FDM para uma placa com condições de contorno de Dirichlet.

CAPÍTULO III

METODOLOGIA

Esta seção discorre a respeito da implementação do problema proposto. O fenômeno simulado é o *sloshing*, que será brevemente apresentado aqui, além das hipóteses assumidas e as particularidades da implementação utilizando o código aberto *Palabos*.

O objetivo deste trabalho consiste em simular o escoamento bifásico em um tanque de combustível, em outras palavras, analisar o movimento líquido de uma superfície livre em um reservatório parcialmente cheio. Pois, conforme mencionado, o estudo deste fenômeno é essencial para o projeto de tanques de combustível e para a implementação de sistemas de controle nos veículos espaciais, além de outras aplicações encontradas na literatura, tais como, tanques de transporte de GNL (Gás Natural Liquefeito) (Rhee, 2005) e fluxo de água no convés de um navio (Veldman, 2006).

3.1 Estudo do Sloshing

Alguns métodos analíticos para solução de problemas associados a tanques de combustíveis foram propostos ainda em meados do século XX, quando a simulação computacional estava em seu pleno início (Ibrahim, 2005). Problemas simples foram alvos destes métodos trazendo uma oportuna compreensão fundamental do *sloshing*. Resumidamente, *sloshing* pode ser descrito como um fluxo líquido de superfície livre em um reservatório submetido a um movimento forçado. Com esta descrição é possível perceber os muitos exemplos já citados até aqui, até

mesmo no cotidiano, como caminhar segurando um copo de água. Porém, este fenômeno pode ser mais intenso e até perigoso. Por isso o estudo se faz necessário, é possível assim, prever e evitar rompimentos de tanques de combustíveis em veículos espaciais, caminhões rodoviários, vagões ferroviários, grandes tanques de petróleo e reatores nucleares (Rhee, 2005). Além disso, o conhecimento do comportamento do fenômeno pode trazer soluções como projetos de sistemas de controle de trajetória em espaçonaves e controle de oscilações causadas por terremotos em edifícios altos (Celebi; Akyildiz, 2002).

Alguns incidentes envolvendo *sloshing* foram relatados. Em mares agitados, grandes massas de água podem invadir o convés do navio quando o bordo livre é ultrapassado por ondas altas. O bordo livre indica até onde a água pode alcançar com o navio carregado em cada época do ano e zona a navegar. A massa de água sólida de cor verde em contraste com a água branca composta por espuma é chamada de água verde e pode ocorrer na proa, no convés ou nas habitações do navio (Kleefsman, 2005). Principalmente no FPSO, onde muitos equipamentos sensíveis estão presentes no convés, a água verde causa muitos danos, como o incidente na plataforma de combate a incêndios da *Emerald* FSU (Unidade de Armazenamento Flutuante, do inglês (*Floating Storage Unit*)), mostrado na Fig 3.1.



Figura 3.1 – Danos causados à plataforma de combate a incêndios da *Emerald* FSU por água verde. Fonte: Kleefsman (2005)

Em janeiro de 2000, os alojamentos na proa do FPSO Varg foram atingidos por água verde,

resultando na inundação do segundo andar devido ao dano causado em uma janela, conforme Fig. 3.2.



Figura 3.2 – Danos causados na proa do FPSO Varg. Fonte: Kleefsman (2005)

Outro exemplo bem conhecido se refere ao primeiro pouso lunar da missão Apollo 11. Em seu pouso real, o escoamento do combustível restante causou um movimento oscilatório do Módulo Lunar, cerca de 2 a 3 graus, prejudicando uma manobra segura de pouso (NASA, 2019). Ainda neste contexto, a missão NEAR (Near Earth Asteroid Rendezvous) mostrada na Fig. 3.3, que teve uma espaçonave enviada para voar próximo ao asteróide Eros e durante a correção do curso final, a espaçonave experimentou movimento inesperado, o que a levou a entrar em um modo de segurança. O controle de solo foi capaz de recuperar a missão, apesar de levar um período de 13 meses de atraso do encontro planejado. Mais uma vez, o combustível foi chamado de causa provável (Veldman, 2006).



Figura 3.3 – Impressão artística da nave espacial NEAR. Fonte: Veldman (2006)

Em geral, o *sloshing* é um fenômeno de superfície livre não-linear envolvendo movimentos altamente transitórios, por esta razão, o estudo só era possível com métodos experimentais. Contudo, as ferramentas computacionais de CFD contribuem para o estudo deste fenômeno, mas requer métodos de simulação numérica que sejam capazes de lidar com este tipo de não-linearidade e conforme mencionado na Seção 2, o método Lattice Boltzmann é um bom candidato. Isso porque, este trabalho envolve a simulação de um escoamento bifásico e o LBM possui um fácil tratamento para este tipo de fluxo, não sendo necessário rastrear as interfaces entre as duas fases. E assim, para a simulação proposta foi utilizado um código aberto desenvolvido por uma equipe de cientistas baseado no método Lattice Boltzmann chamado de Palabos. Na sequência será brevemente discutido as funcionalidades da Palabos, bem como as justificativas de escolha, as equações que permitiram a obtenção dos resultados e as hipóteses assumidas para a simulação.

3.2 Utilização da Palabos

A Palabos pode ser descrito como um *framework* de uso geral para CFD, com um *kernel* baseado no método Lattice Boltzmann (LBM). Ele foi desenvolvido para ser usado como ferramenta de pesquisa e de engenharia, conhecendo o LBM é possível estender a biblioteca com modelos próprios (Palabos, 2019). Ela não necessita de instalação e está disponível gratuitamente

sob os termos de uma licença AGPLv3 de código aberto, através do *link*: <http://www.palabos.org/>.

O código é escrito em C++ utilizando o paradigma da orientação a objetos e possui poucas dependências externas, como Posix e MPI para as simulações com processamento paralelo. Neste trabalho, não foi utilizado paralelização. O código não dispõe de interface gráfica com o usuário.

A biblioteca Palabos mostra desempenhos particularmente notáveis na simulação do fluxo complexo de fluidos, basta visitar o site e encontrar diversos trabalhos que a utilizaram. É freqüente encontrar códigos LBM que implementem recursos avançados específicos, mas que são considerados rígidos para permitir outras extensões, como o acoplamento de duas redes para fluxos multifásicos. Para contornar este problema, os recursos implementados na arquitetura de software foram amplamente desenvolvidos como conceitos ortogonais. É possível, por exemplo, formular uma variante do modelo clássico de colisão BGK sem conhecimento dos componentes de software avançados da Palabos, e obter automaticamente uma versão paralela do código ou um código multifásico baseado no novo modelo (Palabos, 2019).

Ao utilizá-la, um conceito primordial são os chamados objetos dinâmicos, que estão associados a cada célula de fluido e determinam a natureza da etapa de colisão local. Este é um dos motivos da Palabos ser reconhecida pela eficiência numérica em tais modelos. Estruturas de dados específicas também estão disponíveis para algoritmos que não são estritamente locais, como por exemplo condições de contorno específicas ou forças entre partículas em modelos multifásicos.

A estrutura da grade nas simulações é baseada em uma abordagem multi-bloco. Cada bloco se comporta como uma implementação básica do LBM, enquanto ingredientes avançados de software, como paralelismo e implementações de domínio esparso, são cobertos por acoplamentos de interface específicos entre blocos.

O código C++ da Palabos é destinada a oferecer um único código aplicável a diferentes arranjos de *lattice*, em outras palavras, é possível escrever um único código genérico em 3D e executá-lo nos arranjos D3Q15, D3Q19 e D3Q27.

A versão utilizada foi a Palabos 2.0, por apresentar algumas ferramentas necessárias na solução do problema aqui abordado. Além de que, esta versão oferece um código exemplo próximo ao necessário para a simulação do escoamento bifásico em um tanque com condição de

baixo enchimento. Assim, foi necessário apenas algumas alterações e alguns acréscimos de linhas de código para obter os resultados encontrados neste trabalho.

É importante ressaltar algumas ferramentas disponíveis na Palabos e que foram utilizadas aqui. São elas:

- Fluidos multifásicos;
- Fluxos de superfície livre (abordagem volume de fluido);
- Modelo LES de Smagorinsky para turbulência de fluidos;
- Modelo BGK;
- Arranjo de *lattice* D3Q19.

O LBM implementado na Palabos consiste em um *solver* explícito. Com uma única etapa de iteração, o estado do fluido evolui de um tempo t para o próximo tempo $t + dt$. Assim, a iteração assume os seguintes passos:

1. Todas as variáveis de fluido são definidas no tempo t . As populações de partículas estão em estado de pré-colisão e são chamadas de populações de entrada;
2. O operador de colisão é aplicado a todas as partículas levando-as ao estado de pós-colisão e são chamadas de variáveis de saída;
3. O operador de advecção é aplicado ao *lattice*;
4. Todos os processadores de dados são executados de modo a representar as operações não locais ou acoplamentos entre *lattices*;
5. As populações estão novamente em estado de pré-colisão, mas no tempo $t + dt$.

O código disponibilizado pela Palabos e utilizado como base neste trabalho é conhecido por *Dam Break* e se encontra na Galeria na seção de "Fluxo multifásico / Superfície livre". O código final pode ser visto no Apêndice II.

3.3 Modelagem Numérica

Para o modelo de superfície livre, a simulação realizada descreve a evolução de um fluido inicialmente confinado em um tanque retangular. uma das paredes laterais é removida de modo que o fluido entre em um domínio confinado maior. Para isso, a Palabos utiliza um modelo de superfície livre inspirado em um método de volume de fluido e realizando algumas adaptações, foi possível estudar o comportamento do *sloshing*, observando a distribuição de velocidade e pressão de impacto ao longo do tempo.

Impactos violentos no *sloshing* pode resultar em danos estruturais. Estudos experimentais e teóricos têm sido realizados a fim de quantificar as pressões exercidas pelo impacto em paredes verticais, principalmente em regiões de maior formação de vórtices (Ferreira, 2010).

O fluxo de um fluido viscoso, homogêneo e incompressível é descrito pela conservação de massa e momento dado pelo LBM, respectivamente

$$\rho(r,t) = \int M f(r,c,t) dc \quad (3.1)$$

$$\rho(r,t)u(r,t) = \int M c f(r,c,t) dc \quad (3.2)$$

onde M é a massa molecular, ρ é a densidade do fluido, $\vec{u}$ é o vetor velocidade do fluido, $\vec{u}_a$ é a velocidade da partícula relativa à velocidade do fluido, $\vec{u}_a = \vec{c} - \vec{u}$ e $f(r,c,t)$ é a função distribuição de probabilidade que representa o número de partículas na posição r com velocidade c no instante t .

Por fim, equação que representa a colisão e advecção discretizada é

$$f_i(r + c_i \Delta t, t + \Delta t) = f_i(r, t) + \frac{\Delta t}{\tau} [f_i^{eq}(r, t) - f_i(r, t)] \quad (3.3)$$

onde Δt é o passo no tempo, τ é o tempo de relaxamento e f^{eq} é a distribuição de equilíbrio local, uma função distribuição de Maxwell-Boltzmann em cada ligação de *lattice*.

3.3.1 Condições de Contorno

Nas paredes do tanque foi utilizado condições de contorno sem deslizamento $\vec{u} = 0$. O código da Palabos utilizado como base fornece no contorno de fluxo de entrada dada por uma onda de Stokes de 5ª ordem e no fluxo de saída uma condição de Sommerfeld.

3.3.2 Superfície Livre

Na superfície livre, a tensão normal e tangencial é dada por, respectivamente

$$-p + 2\mu \frac{\partial u_n}{\partial n} = -p_0 + \sigma \kappa \quad (3.4)$$

$$\mu \left(\frac{\partial u_n}{\partial t} + \frac{\partial u_t}{\partial n} \right) = 0 \quad (3.5)$$

onde u_n e u_t são as componentes normal e tangencial da velocidade, respectivamente, p_0 , é a pressão atmosférica, σ é a tensão superficial, κ denota a curvatura total da superfície livre e μ é a viscosidade dinâmica do fluido. O ângulo entre a superfície livre e a parede é denominado de ângulo de contato, na simulação foi utilizado um ângulo de contato estático com valor de 80° .

Na Palabos, a modelagem da superfície livre é feita dentro da classe *FreeSurfaceFields3D*.

3.3.3 Monitoramento da Pressão no tanque

Utilizando o modelo BGK, sabe-se que a pressão é uma função da densidade ρ e é dada na forma

$$p = c_s^2 \rho \quad (3.6)$$

E o arranjo de *lattice* D3Q19 apresentado na Seção 1, informa que a velocidade do som em unidades de *lattice* corresponde a

$$c_s = \frac{1}{\sqrt{3}} \quad (3.7)$$

Substituindo 3.7 em 3.6, tem-se

$$p = \frac{\rho}{3} \quad (3.8)$$

Pela equação 2.40, a soma de todas as funções distribuição representa a densidade macroscópica do fluido, isto é

$$\rho = \sum_{i=0}^{18} f_i^{eq} \quad (3.9)$$

Nota-se que encontrando a densidade no ponto desejado, encontra-se a pressão no referido ponto. Logo, utilizando as ferramentas disponíveis na Palabos, há uma função *computeDensity()* que retorna uma matriz de valores de densidade média em unidades de *lattice* em cada célula do domínio. Logo, para a obtenção da pressão no sistema de unidades físicas, faz-se necessário a conversão de unidades entre o espaço mesoscópico e macroscópico.

Para o cálculo numérico, a conversão entre o espaço LBM (mesoscópico) e o espaço físico (macroscópico) é essencial, devido o comprimento e tempo discretos no modelo numérico. O cálculo do LBM começa a partir de seu problema físico no espaço macroscópico, e os resultados também devem retornar ao espaço macroscópico para compreensão da natureza física do problema (Huang et al., 2019). Assim, a equação implementada para o monitoramento da pressão foi

$$p = c_s^2 (\rho^* - 1) \frac{\Delta x^2}{\Delta t^2} \rho \quad (3.10)$$

onde c_s é a velocidade do som, ρ^* é a densidade em unidades de *lattice*, Δx e Δt representam, o incremento no espaço em metros e no tempo em segundos, respectivamente. Por fim, ρ é a densidade física do fluido.

Na equação 3.10, ρ^* é obtido através do método *computeDensity()* da classe *Cell* implementada na Palabos. A subtração $(\rho^* - 1)$ normaliza a matriz de densidades de forma que ao espaço vazio seja atribuído $p = 0$, o que é coerente, pois o fluido está confinado em um tanque fechado.

Comparando as equações 3.6 e 3.10, nota-se que a equação para obtenção da pressão macroscópica continua sendo função da densidade, acrescentando apenas termos que forneçam

a pressão em unidades físicas, ou seja:

$$\frac{\Delta x^2}{\Delta t^2} \rho = \frac{\frac{[m^2]}{[s^2]} \frac{[kg]}{[m^3]}}{[ms^2]} = \frac{[kg]}{[ms^2]} = \frac{massa}{comprimento * tempo^2} = [Pa]$$

Assim, na equação 3.10 a pressão é dada em Pascal, unidade no SI.

O objetivo aqui é monitorar a pressão de impacto na parte inferior da frente do tanque e avaliar seu comportamento com o encontrado na literatura. Abaixo pode-se observar a posição de interesse. A geometria utilizada para representar o tanque é retangular.

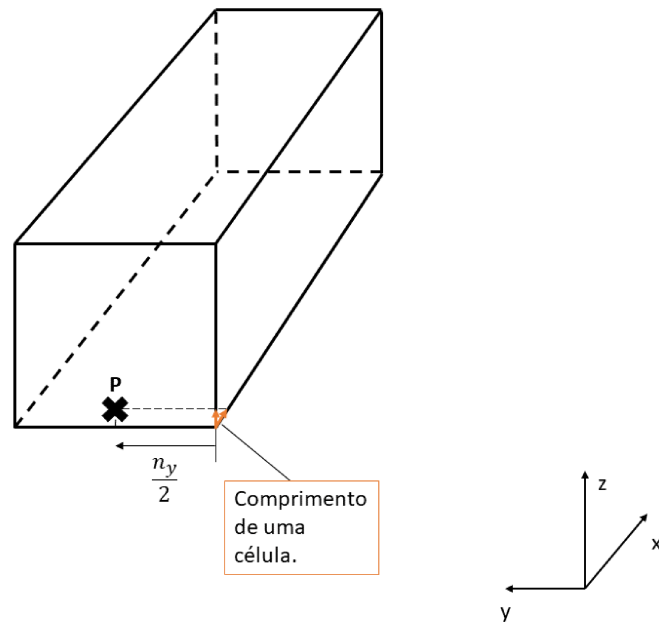


Figura 3.4 – Representação do tanque com a posição de interesse para monitoramento da pressão.

A Fig. 3.4 mostra uma representação do tanque e o ponto P onde é monitorada a pressão de impacto. Nota-se que o ponto P é próximo a parede esquerda do plano yz. Foi considerado desta forma, pois é razoável pensar no posicionamento real de um sensor neste ponto. Por esse motivo, no código foi utilizado $P(x_{cell}, n_y/2, z_{cell})$, onde x_{cell} e z_{cell} indica o deslocamento de uma célula em x e em z , respectivamente e n_y representa a quantidade de células na direção y do tanque, fazendo com que a pressão seja coletada na metade exata do comprimento em y do plano yz .

3.3.4 Hipóteses para a simulação

Para a modelagem do escoamento bifásico foi utilizada a densidade da água $\rho = 1000 \text{ kg/m}^3$ em unidades físicas ou $\rho^* = 0$ em unidades de *lattice* para representar o combustível líquido e a densidade do ar $\rho = 1,29 \text{ kg/m}^3$ em unidades físicas ou $\rho^* = 1$ para representar a mistura de gases dentro do tanque.

No cálculo da tensão superficial para modelagem da superfície livre foi definido um valor de densidade vazia $\rho_{empty} = 1$. Esse valor atribuído é o recomendável nas simulações LBM para manter a maior precisão possível nos cálculos. Usar valores mais altos pode afetar a precisão e, por fim, a estabilidade numérica do LBM.

Foi utilizado um tanque com dimensões $3,22 \times 1 \times 1 \text{ m}$ com um volume de $1,992 \times 1 \times 0,55 \text{ m}$ de água a ser liberado, conforme Fig. 3.5.

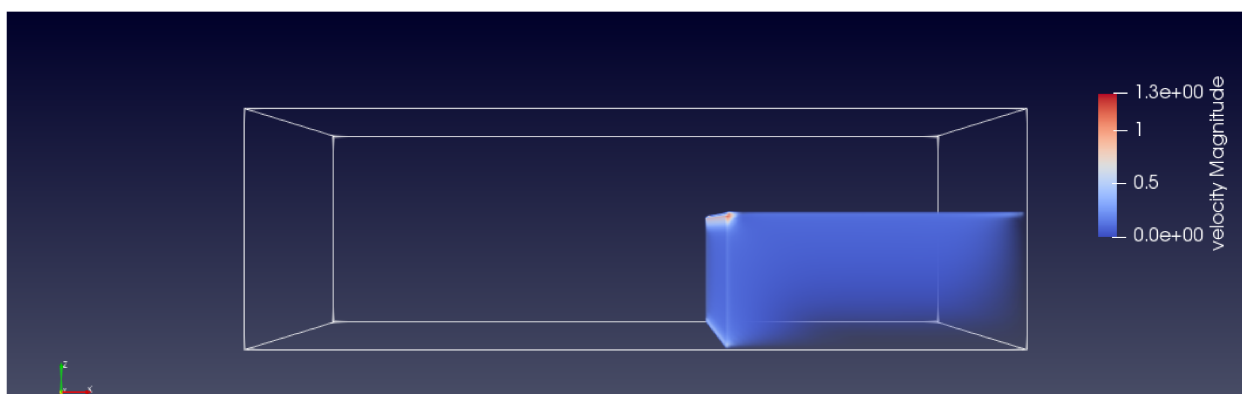


Figura 3.5 – Representação do tanque com o volume de fluido a ser liberado.

CAPÍTULO IV

RESULTADOS E DISCUSSÕES

Os parâmetros de entrada do programa foram:

- Viscosidade cinemática em unidades físicas: $1 \times 10^{-6} m^2/s$;
- Número de Bond: 100;
- Ângulo de contato: 80° ;
- Número de nós de *lattice* em *z* no tanque: 100;
- Passo no tempo: $1 \times 10^{-4} s$;
- Número de iterações: 80000.

A partir do número de iterações e do passo de tempo, sabe-se que a simulação teve um tempo total de $t = 8 s$.

Na figura 4.1 é possível observar o comportamento qualitativo da distribuição de densidade dentro do tanque para diferentes valores de tempo.

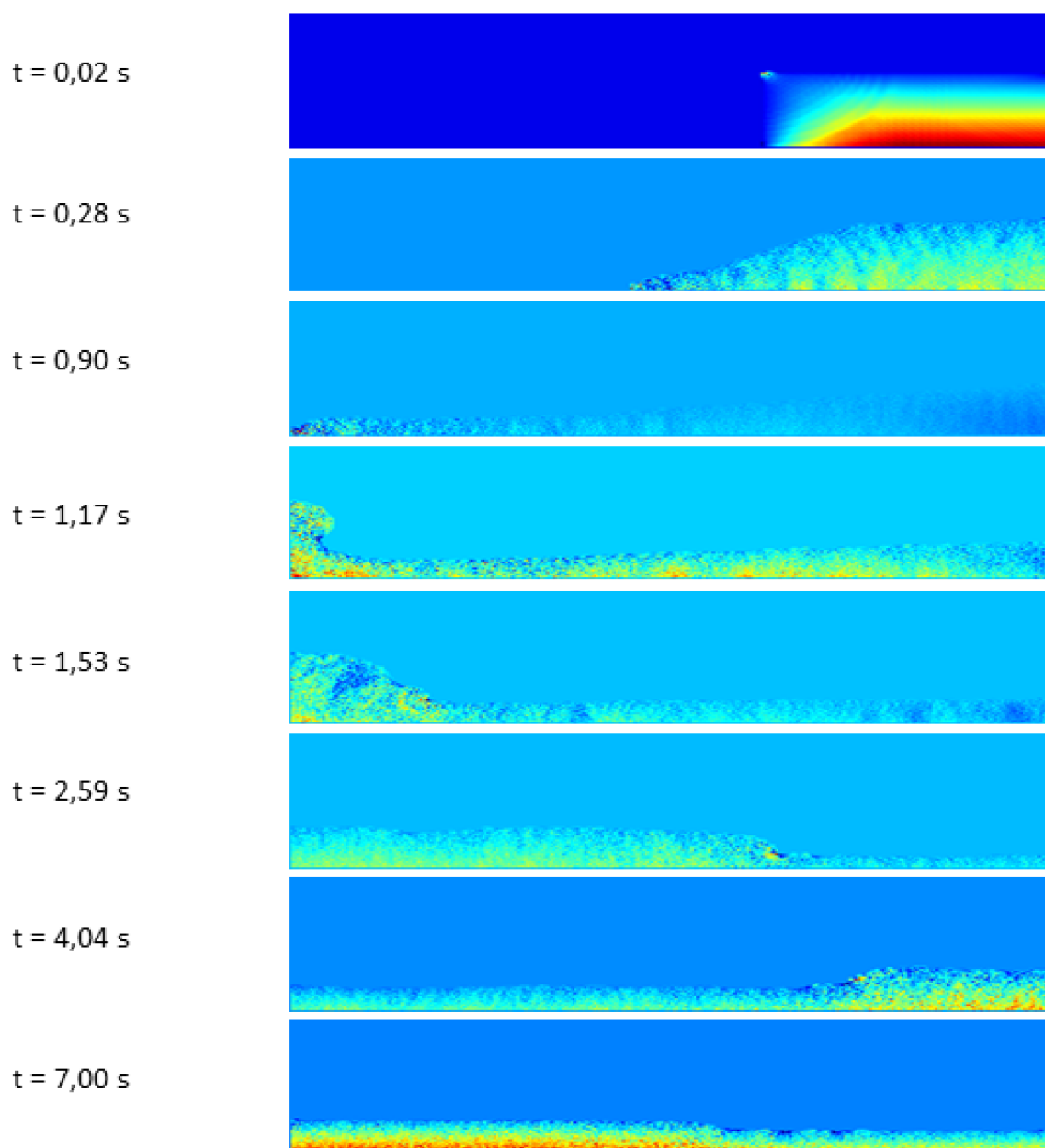


Figura 4.1 – Histórico de densidade para diferentes valores de tempo da simulação.

Nota-se que algumas instabilidades na área interfacial causada pela variação da densidade da mesma. Além disso, é possível observar que o líquido chega à parede esquerda em aproximadamente $t = 0,90 \text{ s}$, formando uma onda em torno de $t = 1,17 \text{ s}$.

Do ponto de vista da velocidade, as Figs. 4.2 à 4.5 mostram a evolução da velocidade entre os tempos $t = 0 \text{ s}$ e $t = 4,0 \text{ s}$.

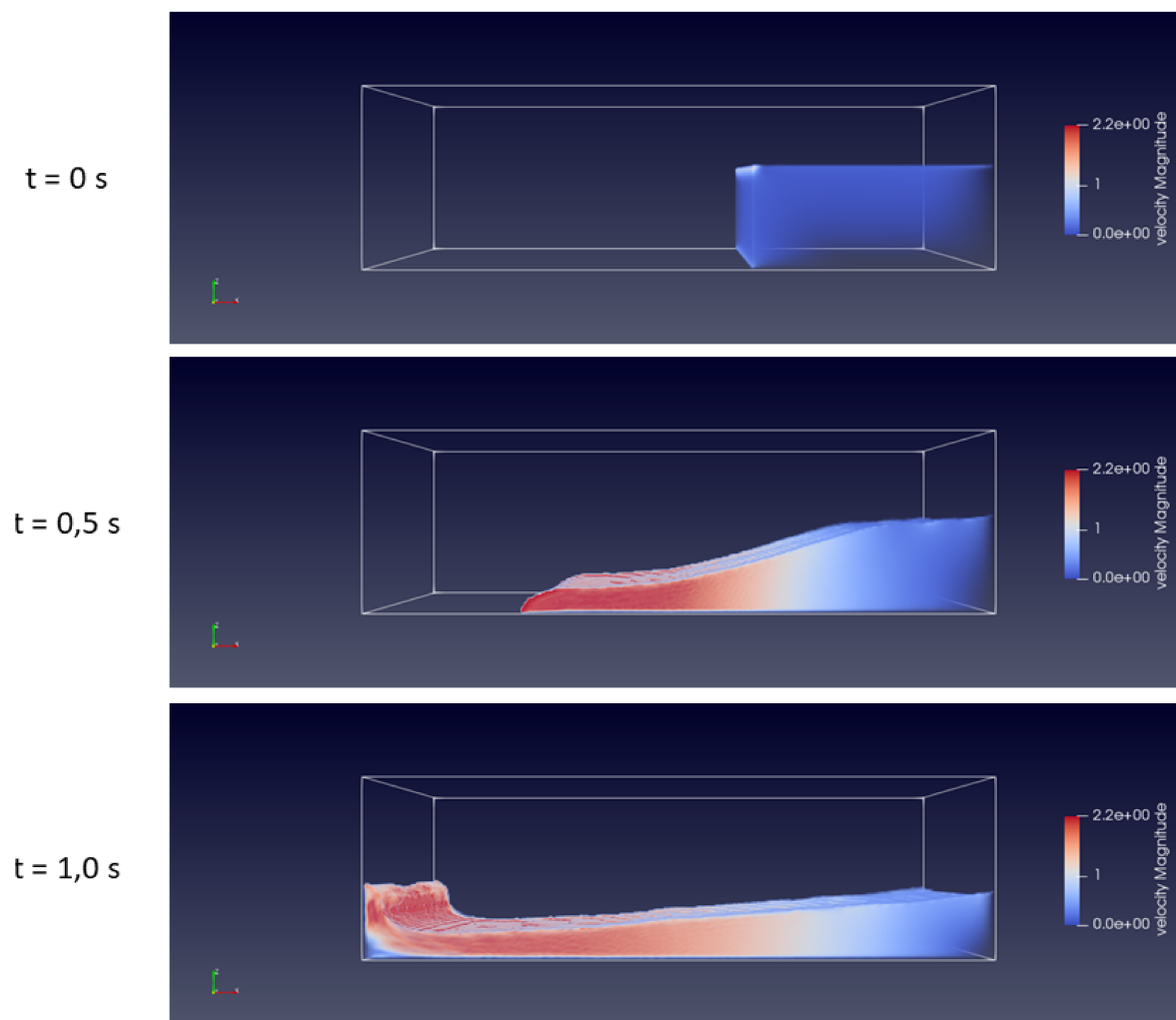


Figura 4.2 – Evolução da velocidade entre $t = 0 \text{ s}$ e $t = 1 \text{ s}$.

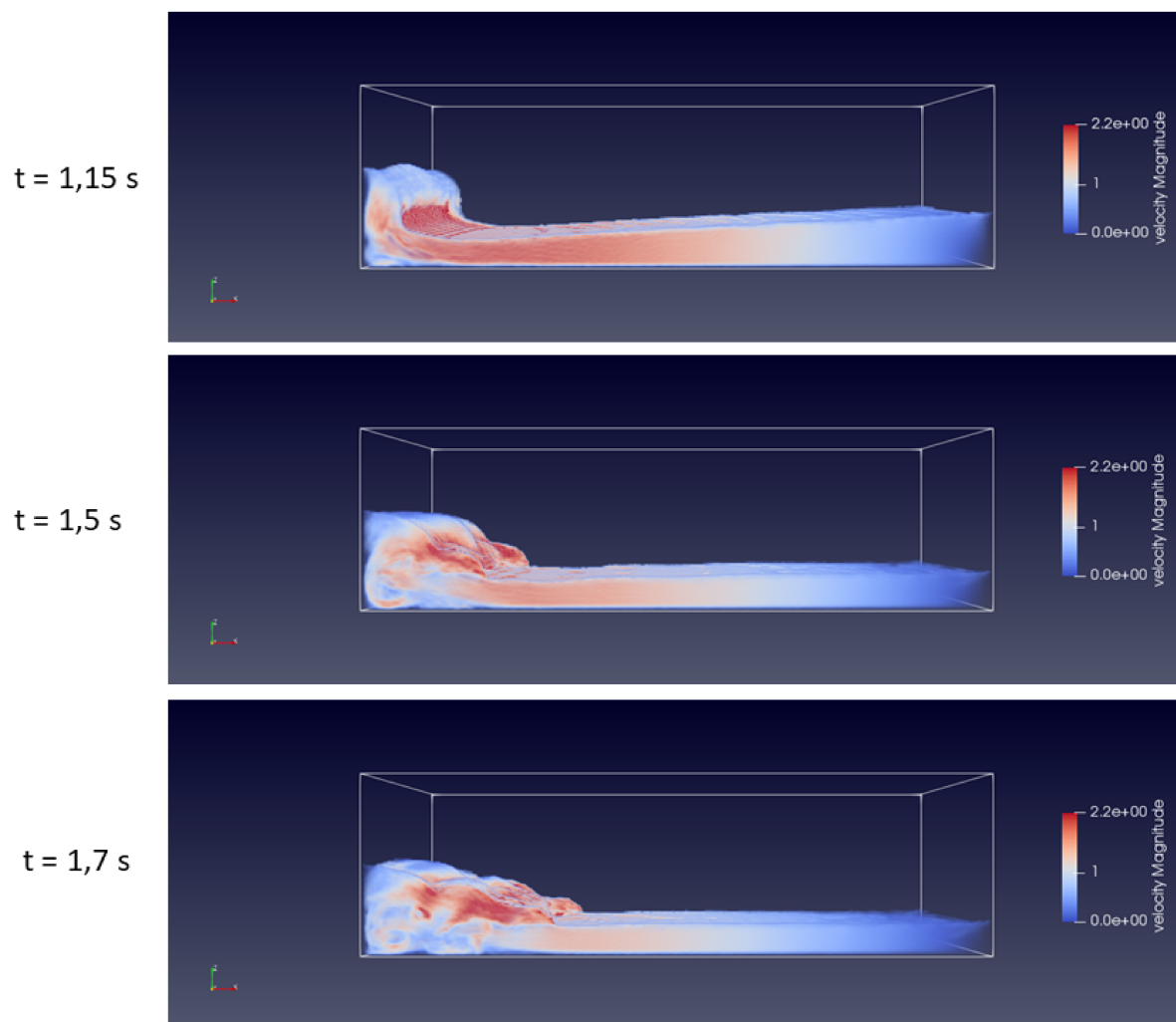


Figura 4.3 – Evolução da velocidade entre $t = 1,15 \text{ s}$ e $t = 1,7 \text{ s}$.

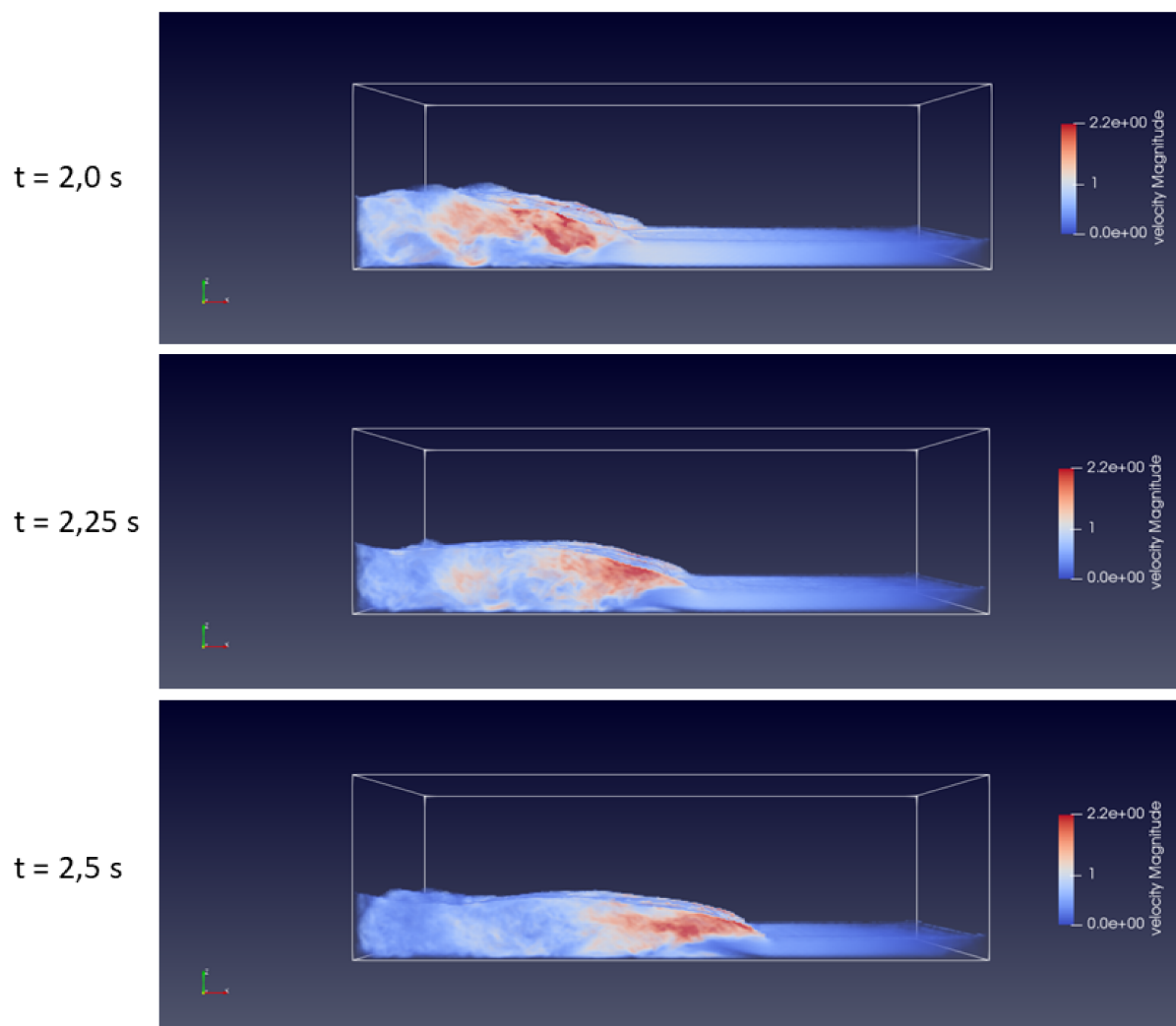


Figura 4.4 – Evolução da velocidade entre $t = 2,0 \text{ s}$ e $t = 2,5 \text{ s}$.

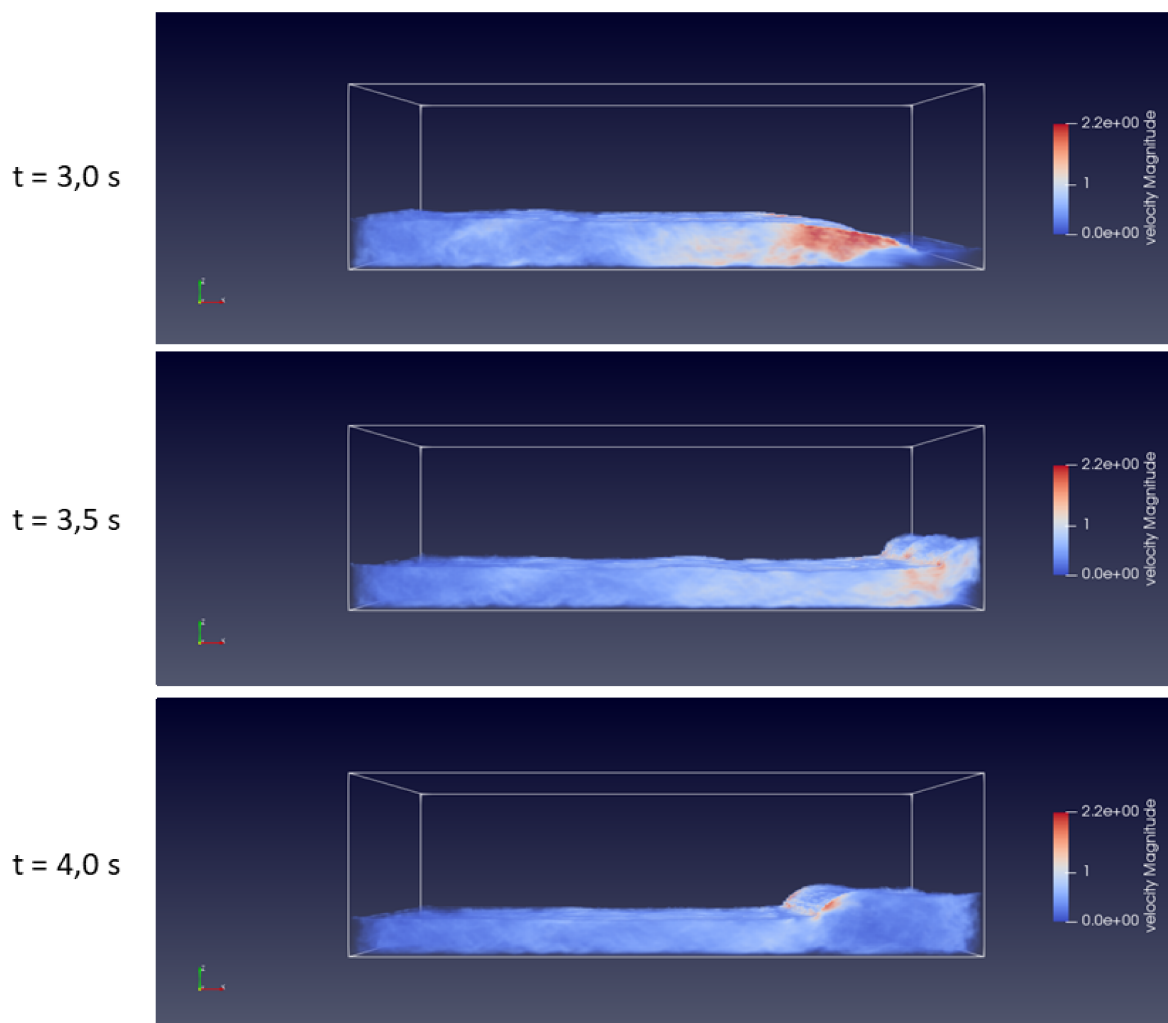


Figura 4.5 – Evolução da velocidade entre $t = 3,0 \text{ s}$ e $t = 4,0 \text{ s}$.

Nas Figs. 4.2 à 4.5 é possível observar a diminuição da quantidade de movimento logo após o impacto com a parede vertical esquerda, decrescendo ainda mais quando o escoamento tende a retornar em direção à parede vertical direita e em seguida para a esquerda novamente, repetindo este movimento oscilatório. Isso ocorre devido a condição de contorno de não-eskorregamento, que impõe que a velocidade do escoamento rente a parede inferior do tanque seja nula, e a viscosidade do fluido, que faz com que os fluido atrite com ele mesmo, aumentando assim a resistência ao escoamento, pois uma parcela do fluido tende a se mover em uma direção enquanto a outra tende a se mover na direção oposta, porém estas duas parcelas estão em contato. Além disso, o decréscimo na quantidade de movimento traz a diminuição na formação de ondas. Foi construído um vídeo da simulação com toda a evolução da velocidade e

pode ser acessado pelo *link* <https://youtu.be/mFMTyAKfTFY>.

A pressão de impacto ao longo do tempo foi obtida. Porém, estes resultados mostraram-se ruidosos, com isso, foi utilizado um filtro passa-baixa, dado pela equação

$$p_{fil}(i) = p_{fil}(i - 1) + \gamma(p(i) - p_{fil}(i - 1)) \quad (4.1)$$

onde $p_{fil}(i)$ é a pressão filtrada atual, $p_{fil}(i - 1)$ é a pressão filtrada anterior, $p(i)$ é a pressão obtida e γ é o fator de filtragem no filtro passa-baixa, que pode assumir valores entre 0 (não passa sinal) e 1 (não filtra o sinal), sendo que nestes resultados foi utilizado $\gamma = 0,1$.

Uma comparação entre o sinal de pressão obtido e o sinal após a filtragem pode ser visualizado na Fig. 4.6.

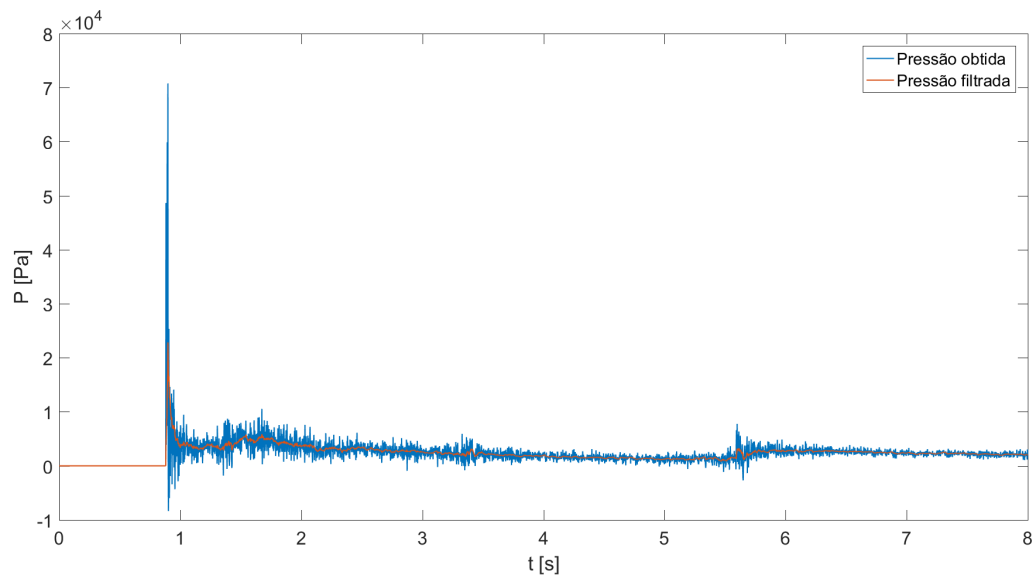


Figura 4.6 – Pressão de impacto obtida ao longo do tempo e pressão após filtragem.

A Fig. 4.7 apresenta apenas o sinal filtrado. Observa-se que o comportamento obtido é muito próximo ao encontrado na literatura, conforme Peregrine (2003) que apresenta resultados experimentais apresentados na Fig. 4.8. Tal comportamento tem como característica ressaltante o pico máximo de pressão que ocorre no momento do choque da primeira onda na parede esquerda, em seguida esta onda caminha na direção contrária e após o choque na parede direita retorna colidindo novamente na parede esquerda formando um pico secundário fraco. O restante do

perfil mostra a formação de ondas menos violentas e íngremes, devido à modificação da pressão hidrostática pela aceleração vertical da água, perdendo a magnitude de sua força neste movimento de vai e vem, por fim, o escoamento entra em equilíbrio e a pressão torna-se estável com valor quase constante até o fim da simulação. Segundo Peregrine (2003), o campo de pressão está associado a propagação e reflexão das ondas no tanque. Nota-se nas Figs. 4.8 e 4.7 que os tempos apresentados nos eixos das abcissas são diferentes e isso se dá por considerações dimensionais distintas, porém o comportamento apelidado por "*church roof*" é o mesmo.

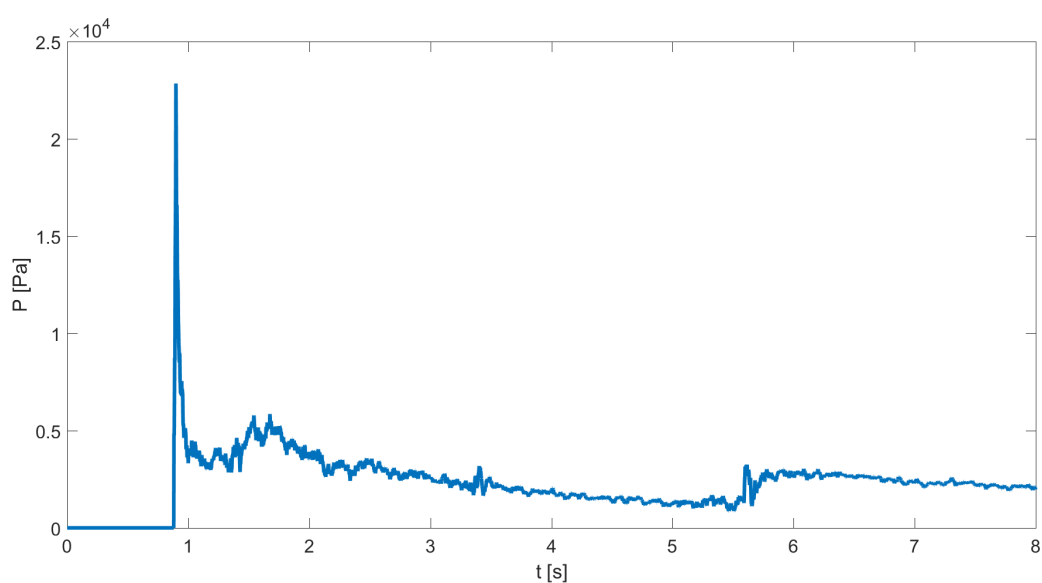


Figura 4.7 – Histórico da pressão exercida pelo impacto de ondas na parede vertical esquerda após filtragem.

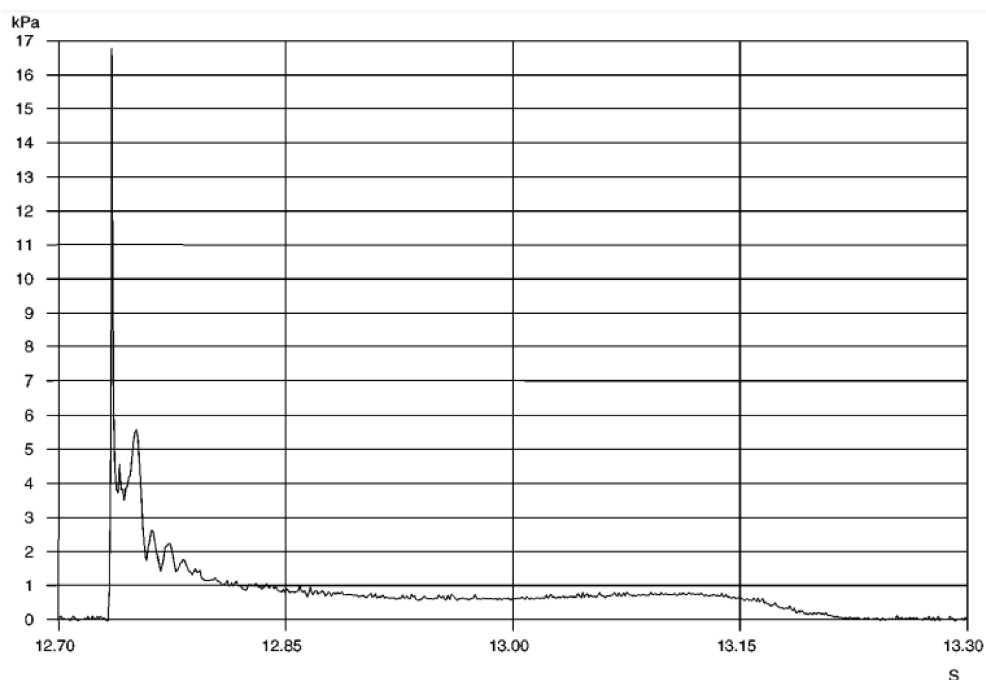


Figura 4.8 – Comportamento para a pressão causada pelo impacto de uma onda sobre uma parede vertical. Fonte: Peregrine (2003).

A Fig. 4.7 mostra o perfil típico da pressão obtida na parede vertical esquerda do tanque em questão. No gráfico, é possível notar a formação de dois picos. O primeiro e de maior amplitude acontece devido à inércia do fluido, ocorrendo em um pequeno intervalo de tempo na ordem de milissegundos. E o segundo ocorre devido à pressão hidrostática causada pelo movimento descendente da onda.

O pequeno aumento de pressão em $t = 5,62 \text{ s}$ visto na 4.7 ocorre quando há um segundo impacto do escoamento na parede esquerda. A Fig. 4.9 mostra o momento exato deste impacto.



Figura 4.9 – Momento em que há impacto de fluido pela segunda vez na parede vertical esquerda em $t = 5,62 \text{ s}$.

CAPÍTULO V

CONCLUSÕES E TRABALHOS FUTUROS

5.1 Conclusões

O Método Lattice Boltzmann foi capaz de resolver problemas complexos como, escoamento bifásico e fluxo de superfície livre, mostrando resultados coerentes e próximos aos encontrados na literatura.

A simulação realizada permitiu a obtenção do comportamento geral do *sloshing*, podendo notar, principalmente, o histórico da pressão na parede vertical esquerda do tanque devido ao impacto das ondas formadas pelo referido fenômeno. Além de resultados qualitativos como os apresentados para a velocidade.

A utilização da biblioteca *Palabos* se mostrou trabalhosa, devido a falta de documentação, não padronização dos algoritmos, não realização das boas práticas de programação e falha na comunicação entre a comunidade que a utiliza, não havendo resposta a alguns questionamentos realizados nos fóruns destinados para troca de experiências e conhecimentos. Uma alternativa seria a utilização de outras bibliotecas implementadas com o Método Lattice Boltzmann encontradas por meio de uma breve pesquisa na literatura. Cabe ressaltar que o objetivo deste trabalho não foi avaliar o desempenho da *Palabos* frente a outras bibliotecas.

5.2 Trabalhos Futuros

Primeiramente, pensando em um seguimento para a simulação mostrada neste trabalho, um melhor estudo do escoamento bifásico em um tanque de combustível pode ser obtido realizando experimentos, a fim de poder validar a simulação apresentada. Além de ser possível a realização de testes com códigos que utilizem computação paralela, como o Método Lattice Boltzmann propõe. E, pensando em termos de eficiência, é plausível pensar na utilização de algoritmos genéticos que diminuam a ociosidade dos processadores em problemas de paralelização. Tendo assim, um método de simples implementação com a praticidade da paralelização e ainda o aumento dessa eficiência de processamento com o uso de algoritmos genéticos.

REFERÊNCIAS BIBLIOGRÁFICAS

- MOHAMAD, A. A. Lattice-Boltzmann Method: Fundamentals and Engineering Applications with Computer Codes. [S.l.]: Springer, 2011.
- IBRAHIM, R. A. Liquid sloshing dynamics, theory and applications. [S.l.]: Cambridge, 2005.
- FERZIGER J.; PERIC, M. Computational Methods for Fluid Dynamics. 3. ed. [S.l.]: Springer Verlag, 2013.
- VELDMAN, A. E. P. The simulation of violent free-surface dynamics at sea and in space. European Conference on Computational Fluid Dynamics, 2006.
- RHEE, S. H. Unstructured Grid Based Reynolds-Averaged Navier-Stokes Method for Liquid Tank Sloshing. Journal of Fluids Engineering, 2005.
- HUANG, J.; BAO, C.; JIANG, Z.; ZHANG, X. A general approach of unit conversion system in lattice Boltzmann method and applications for convective heat transfer in tube banks. International Journal of Heat and Mass Transfer, 2019.
- WOLF-GLADROW, D. A. Lattice-Gas Cellular Automata and Lattice Boltzmann Models. [S.l.]: Springer, 2005.
- SUCCI, S. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond (Numerical Mathematics and Scientific Computation). [S.l.]: Oxford University Press, 2001.
- JUNIOR, D. P. C. Comportamento do sloshing em ambiente de microgravidade. Dissertação (Mestrado). Instituto Nacional de Pesquisas Espaciais, 2017.
- FERREIRA, L. R. D. Desenvolviemnto de uma bancada de testes para validação de um tanque de ondas numérico. Dissertação (Mestrado). Universidade Federal Fluminense, 2010.

- CHAPMAN S.; COWLING, T.G. The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases. [S.l.]: Cambridge University Press, 1970.
- CELEBI, M. S.; AKYILDIZ, H. Nonlinear Modeling of Liquid Sloshing in a Moving Rectangular Tanks. Ocean Eng., 2002.
- KLEEFSMAN, K. M. T. Water impact loading on offshore structures - A numerical study. University of Groningen, 2005.
- PEREGRINE, D. H. Water-Wave Impact on Walls. Annual Review of Fluid Mechanics, 2003.
- NASA. Apollo 11 Lunar Surface Journal. The first lunar landing. 2019. <http://www.hq.nasa.gov/office/pao/History/alsj/a11/a11.landing.html>. [Acessado dia 22 de Março de 2019].
- Palabos. CFD, COMPLEX PHYSICS. 2019. <http://www.palabos.org/>. [Acessado dia 22 de Março de 2019].

APÊNDICES

APÊNDICE I - CÓDIGOS UTILIZADO PARA A EQUAÇÃO DA DIFUSÃO DE CALOR EM UMA PLACA

Este apêndice traz o código utilizado nas simulação da equação da difusão unidimensional em uma placa com condições de contorno de Dirichlet utilizando os métodos Lattice Boltzmann e Diferenças Finitas. O código pode ser acessado através do link:

<https://gitlab.com/anapaulamoreira/difusaocalorlbmfdm>.

```
%codigo para solucao da equacao da difusao de calor em uma placa
%condicoes de contorno Dirichlet

clc
close all
clear all

%% Método Lattice Boltzmann

n = 100;      % número de nós de lattice
m = n + 1;   % número de nós de lattice + 1
dt=1;         % intervalo de tempo
dx=1;         % comprimento de ligacao

%Inicializando os vetores
f1 = zeros(1,m);
```

```

f2 = zeros(1,m);
T = zeros(1,m);
feq = zeros(1,m);
x = zeros(1,m);

x(1) = 0;    % comprimento inicial da placa

% comprimento total da placa
for i=2:m
    x(i) = x(i-1) + dx;
end

csq = dx*dx/(dt*dt);
alpha = 0.25;    % coeficiente de difusão térmica
omega = 1/(alpha/(dt*csq) + 0.5); % frequência de colisão

mstep=200;    % número total de etapas de tempo
twall=1;    % Temperatura em que a superficie esquerda é submetida

%% Condição inicial

for i=1:m
    T(i)=0; % Valor inicial da temperatura do domínio
    f1(i)=0.5*T(i);
    f2(i)=0.5*T(i);
end

%% main loop
for kk=1:mstep
    % processo de colisao:
    for i=1:m
        T(i)=f1(i)+f2(i);
        feq(i)=0.5*T(i);
        % feq1=feq2=feq
        f1(i)=(1 - omega)*f1(i) + omega*feq(i);
        f2(i)=(1 - omega)*f2(i) + omega*feq(i);
    end
end

```

```

end

% processo adveccao:
for i=2:m-2
f1(m-i) = f1(m-i-1); % f1
f2(i-1) = f2(i);      % f2
end

% Condição de contorno

f1(1)=twall-f2(1);    % temperatura constante, x = 0
f1(m)=f1(m-1);        % adiabática, x=L
f2(m)=f2(m-1);        % adiabática, x=L
end

%% Diferenças Finitas

fo = zeros(1,m);
f = zeros(1,m);
dxd = 1.0;
dtd = 0.500;    % passo de tempo
mstepd = 400;   % numero total de iteracoes

fo(1) = 1.0;    % condicao inicial para o valor antigo de f em x = 0.
f(1) = 1.0;     % condicao inicial para o novo valor de f em x = 0.
fo(m) = fo(m-1); % condicao inicial para o valor antigo de f em x=L
f(m) = f(m-1);  % condicao inicial para o novo valor de f em x=L

for kk=1:mstepd
for i=2:m-1
f(i)=fo(i)+dtd*alpha*(fo(i+1)-2.*fo(i)+fo(i-1))/(dxd*dxd);
end

for i=2:m-1
fo(i)=f(i); % atualizando o valor antigo
end

```

```
fo(m)=f(m-1);    % atualizando a condição de contorno em x = L
end

% plot dos resultados em um grafico
plot (x, T, 'o', x, f, '+', 'MarkerSize', 10, 'LineWidth', 1)
pause(0.1)
frame_h = get(handle(gcf), 'JavaFrame');
set(frame_h, 'Maximized', 1);
set(gca, 'FontSize', 15)
xlabel('x');
ylabel('T');
legend('LBM', 'FDM')
ylim([-0.1 1.1])
print('DirichletLBMFDM', '-dpng')
```

APÊNDICE II - CÓDIGOS UTILIZADO PARA A SIMULAÇÃO DO SLOSHING UTILIZANDO A PALABOS

Este apêndice traz o código utilizado na simulação do escoamento bifásico em um tanque de combustível com geometria retangular através do Método Lattice Boltzmann e do uso da *Palabos*. O código pode ser acessado através do link:

<https://gitlab.com/anapaulamoreira/sloshingtankfuel>.

```
/* This file is part of the Palabos library.
*
* Copyright (C) 2011-2017 FlowKit Sarl
* Route d'Oron 2
* 1010 Lausanne, Switzerland
* E-mail contact: contact@flowkit.com
*
* The most recent release of Palabos can be downloaded at
* <http://www.palabos.org/>
*
* The library Palabos is free software: you can redistribute it and/or
* modify it under the terms of the GNU Affero General Public License as
* published by the Free Software Foundation, either version 3 of the
* License, or (at your option) any later version.
*
* The library is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*/
```

```

/* sloshing adapted from a dam break problem. Este código demonstra o uso básico
* do módulo de superfície livre no Palabos.
* A tensão superficial e os ângulos de contato são opcionais.
*/

```

```

#include "palabos3D.h"
#include "palabos3D.hh"

```

```

using namespace plb;

```

```

#define DESCRIPTOR descriptors::ForcedD3Q19Descriptor

```

```

typedef double T;

```

```

// Constante Smagorinsky para o modelo LES.

```

```

const T cSmago = 0.14;

```

```

// Dimensões físicas do sistema (em metros).

```

```

const T lx = 3.22;

```

```

const T ly = 1.0;

```

```

const T lz = 1.0;

```

```

const T rhoEmpty = T(1);

```

```

plint writeImagesIter    = 20;

```

```

plint getStatisticsIter = 20;

```

```

plint maxIter;

```

```

plint N;

plint nx, ny, nz;

T delta_t, delta_x;

Array<T,3> externalForce;

T nuPhys, nuLB, tau, omega, Bo, surfaceTensionLB, contactAngle;


std::string outDir;

plint obstacleCenterXYplane, obstacleLength, obstacleWidth, obstacleHeight;

plint beginWaterReservoir, waterReservoirHeight;

plint waterLevelOne, waterLevelTwo, waterLevelThree, waterLevelFour;


// Arquivo para salvar os dados de pressão
std::ofstream out;


void setupParameters() {
    delta_x = lz / N;
    nx = util::roundToInt(lx / delta_x);
    ny = util::roundToInt(ly / delta_x);
    nz = util::roundToInt(lz / delta_x);


    T gLB = 9.8 * delta_t * delta_t/delta_x; // Gravidade em unidade de lattice
    externalForce = Array<T,3>(0., 0., -gLB);
    tau          = (nuPhys*DESCRIPTOR<T>::invCs2*delta_t)/(delta_x*delta_x) + 0.5;
    omega        = 1./tau;
    nuLB         = (tau-0.5)*DESCRIPTOR<T>::cs2; // Viscosidade em unidade de lattice


    surfaceTensionLB = rhoEmpty * gLB * N * N / Bo;


    obstacleCenterXYplane = util::roundToInt(0.744*N);
    obstacleLength         = util::roundToInt(0.403*N);

```

```

obstacleWidth      = util::roundToInt(0.161*N);
obstacleHeight     = util::roundToInt(0.161*N);
beginWaterReservoir = util::roundToInt((0.744+1.248)*N);
waterReservoirHeight = util::roundToInt(0.55*N); // 0,55 m de altura de liquido

waterLevelOne  = util::roundToInt(0.496*N);
waterLevelTwo  = util::roundToInt(2.*0.496*N);
waterLevelThree = util::roundToInt(3.*0.496*N);
waterLevelFour  = util::roundToInt((3.*0.496 + 1.150)*N);
}

// Especifica a condição inicial do fluido
// (a cada célula é atribuído o sinalizador "fluido", "vazio" ou "parede").
int initialFluidFlags(plint iX, plint iY, plint iZ) {
// Obstáculo na extremidade esquerda, que é atingido pelo fluido.
// Se insideObstacle = false => nao haveria obstaculo
bool insideObstacle =
iX >= obstacleCenterXYplane-obstacleWidth/2 &&
iX <= obstacleCenterXYplane+obstacleWidth/2 &&
iY >= ny/2-obstacleLength/2 &&
iY <= ny/2+obstacleLength/2 &&
iZ <= obstacleHeight+1;

insideObstacle = false;
if (insideObstacle) {
return freeSurfaceFlag::wall;
}
else if (iX >= beginWaterReservoir && iZ <= waterReservoirHeight) {
return freeSurfaceFlag::fluid;
}
}

```

```

else {
return freeSurfaceFlag::empty;
}
}

void writeResults(MultiBlockLattice3D<T,DESCRIPTOR>& lattice, MultiScalarField3D<T>&
volumeFraction, plint iT)
{
static const plint nx = lattice.getNx();
static const plint ny = lattice.getNy();
static const plint nz = lattice.getNz();
Box3D slice(0, nx-1, ny/2, ny/2, 0, nz-1);

// Imprimir em arquivos '.gif'

ImageWriter<T> imageWriter("leeloo");
imageWriter.writeScaledPpm(createFileName("u", iT, 6),
*computeVelocityNorm(lattice, slice));

imageWriter.writeScaledPpm(createFileName("rho", iT, 6),
*computeDensity(lattice, slice));

imageWriter.writeScaledPpm(createFileName("volumeFraction", iT, 6),
*extractSubDomain(volumeFraction, slice));

// Algoritmo para reconstruir a superfície livre e gravar um arquivo STL.
std::vector<T> isoLevels;
isoLevels.push_back((T) 0.5);
typedef TriangleSet<T>::Triangle Triangle;
std::vector<Triangle> triangles;

```

```

isoSurfaceMarchingCube(triangles, volumeFraction, isoLevels,
volumeFraction.getBoundingBox());

TriangleSet<T>(triangles).writeBinarySTL(createFileName(outDir+"/interface", iT, 6)
+".stl");

// Imprimir em arquivo '.vti'
VtkImageOutput3D<T> vtkOut(createFileName("Velocity", iT, 6), 1.);
vtkOut.writeData<float>(*computeVelocityNorm(lattice), "u", 1.);
vtkOut.writeData<3,float>(*computeVelocity(lattice), "velocity",
delta_x/delta_t);

// Monitorando a pressao de impacto a esquerda no fundo do tanque
T densityData;
float ix, iy, iz;
float pressure;
float fluidDensity = 1000;
ix = 1;
iy = ny/2;
iz = 1;
densityData = lattice.get(ix,iy,iz).computeDensity();
pressure = DESCRIPTOR<T>::(cs2*(densityData-1.)*delta_x*delta_x/
(delta_t*delta_t)*fluidDensity);
out << densityData << "\t\t" << pressure << std::endl;
std::cout << "Density = " << densityData << std::endl;
std::cout << "Pressure = " << pressure << std::endl;
}

void writeStatistics(FreeSurfaceFields3D<T,DESCRIPTOR>& fields) {
pcout << " ----- " << std::endl;
T averageMass = freeSurfaceAverageMass<T,DESCRIPTOR>(fields.freeSurfaceArgs,

```

```

fields.lattice.getBoundingBox());
pcout << "Average Mass: " << averageMass << std::endl;
T averageDensity = freeSurfaceAverageDensity<T,DESCRIPTOR>(fields.freeSurfaceArgs,
fields.lattice.getBoundingBox());
pcout << "Average Density: " << std::setprecision(12) << averageDensity << std::endl;

T averageVolumeFraction = freeSurfaceAverageVolumeFraction<T,DESCRIPTOR>(
fields.freeSurfaceArgs, fields.lattice.getBoundingBox());
pcout<<"Average Volume-Fraction: "<<std::setprecision(12)
<<averageVolumeFraction<<std::endl;

pcout << " ----- " << std::endl;
}

int main(int argc, char **argv)
{
out.open("pressureData.txt"); // Abre o arquivo para salvar os dados de pressão
out << "Densidade*" << "\t" << "Pressao (Pa)" << std::endl;
plbInit(&argc, &argv);
global::directories().setInputDir("./");

if (global::argc() != 8) {
pcout << "Error missing some input parameter\n";
}

try {
global::argv(1).read(outDir);
global::directories().setOutputDir(outDir+"/");

global::argv(2).read(nuPhys);

```

```

global::argv(3).read(Bo);
global::argv(4).read(contactAngle);
global::argv(5).read(N);
global::argv(6).read(delta_t);
global::argv(7).read(maxIter);
}

catch(PlbIOException& except) {
pcout << except.what() << std::endl;
pcout << "The parameters for this program are :\n";
pcout << "1. Output directory name.\n";
pcout << "2. kinematic viscosity in physical Units (m^2/s) .\n";
pcout << "3. Bond number (Bo = rho * g * L^2 / gamma).\n";
pcout << "4. Contact angle (in degrees).\n";
pcout << "5. number of lattice nodes for lz .\n";
pcout << "6. delta_t .\n";
pcout << "7. maxIter .\n";
pcout << "Reasonable parameters on a desktop computer are:
" << (std::string)global::argv(0) << " tmp 1.e-5 100 80.0 40 1.e-3 80000\n";
pcout << "Reasonable parameters on a parallel machine are:
" << (std::string)global::argv(0) << " tmp 1.e-6 100 80.0 100 1.e-4 80000\n";
exit (EXIT_FAILURE);
}

setupParameters();

pcout << "delta_t= " << delta_t << std::endl;
pcout << "delta_x= " << delta_x << std::endl;
pcout << "delta_t*delta_t/delta_x= " << delta_t*delta_t/delta_x << std::endl;
pcout << "externalForce= " << externalForce[2] << std::endl;
pcout << "relaxation time= " << tau << std::endl;

```

```

pcout << "omega= " << omega << std::endl;
pcout << "kinematic viscosity physical units = " << nuPhys << std::endl;
pcout << "kinematic viscosity lattice units= " << nuLB << std::endl;

global::timer("initialization").start();

SparseBlockStructure3D blockStructure(createRegularDistribution3D(nx, ny, nz));

Dynamics<T,DESCRIPTOR>* dynamics
= new SmagorinskyBGKdynamics<T,DESCRIPTOR>(omega, cSmago);

// Se surfaceTensionLB for 0, o algoritmo de tensão superficial será desativado.
// Se contactAngle for menor que 0, o algoritmo de ângulo de contato será desativado.
FreeSurfaceFields3D<T,DESCRIPTOR> fields( blockStructure, dynamics->clone(),
rhoEmpty, surfaceTensionLB, contactAngle, externalForce );

// Define todas as células da parede externa para "parede"
// (aqui, as células em massa também são definidas
// para "parede", mas não importa, porque eles são sobrescritos na próxima linha).
setToConstant(fields.flag, fields.flag.getBoundingBox(), (int)freeSurfaceFlag::wall);

setToFunction(fields.flag, fields.flag.getBoundingBox().enlarge(-1),
initialFluidFlags);

fields.defaultInitialize();

pcout << "Time spent for setting up lattices: "
<< global::timer("initialization").stop() << std::endl;
T lastIterationTime = T();

```

```

for (plint iT = 0; iT <= maxIter; ++iT) {
    global::timer("iteration").restart();

    T sum_of_mass_matrix = T();
    T lost_mass = T();
    if (iT % getStatisticsIter==0) {
        pcout << std::endl;
        pcout << "ITERATION = " << iT << std::endl;
        pcout << "Time of last iteration is " << lastIterationTime <<"seconds"<<std::endl;
        writeStatistics(fields);
        sum_of_mass_matrix = fields.lattice.getInternalStatistics().getSum(0);
        pcout << "Sum of mass matrix: " << sum_of_mass_matrix << std::endl;
        lost_mass = fields.lattice.getInternalStatistics().getSum(1);
        pcout << "Lost mass: " << lost_mass << std::endl;
        pcout << "Total mass: " << sum_of_mass_matrix + lost_mass << std::endl;
        pcout << "Interface cells: "<< fields.lattice.getInternalStatistics().getIntSum(0)
        << std::endl;
    }

    if (iT % writeImagesIter == 0) {
        global::timer("images").start();
        writeResults(fields.lattice, fields.volumeFraction, iT);
        pcout << "Total time spent for writing images: "
        << global::timer("images").stop() << std::endl;
    }

    // colisao e adveccao e todas as operacoes de superficie livre.
    fields.lattice.executeInternalProcessors();
    fields.lattice.evaluateStatistics();
    fields.lattice.incrementTime();

```

```
lastIterationTime = global::timer("iteration").stop();  
}  
out.close();  
}
```