FABIO SUSSUMU KOMORI

DESENVOLVIMENTO DE UM SIMULADOR
COMPUTACIONAL DE DINÂMICA DE FLUIDOS
UTILIZANDO O MÉTODO DE LATTICE
BOLTZMANN

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2012

FABIO SUSSUMU KOMORI

DESENVOLVIMENTO DE UM SIMULADOR
COMPUTACIONAL DE DINÂMICA DE FLUIDOS
UTILIZANDO O MÉTODO DE LATTICE
BOLTZMANN

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:

Microeletrônica

Orientador:

Marcelo Nelson Páez Carreño

São Paulo
2012

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 9 de julho de 2012.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Komori, Fabio Sussumu

Desenvolvimento de um Simulador Computacional de Dinâmica de Fluidos Utilizando o Método de Lattice Boltzmann/ Fabio Sussumu Komori. – ed. rev. – São Paulo, 2012.

87 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos (PSI).

1. Desenvolvimento de software 2. Autômatos celulares 3. Dinâmica de fluidos computacional 4. Simulação de escoamentos 5. Método de Lattice Boltzmann I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos (PSI). II. t.

À minha mãe (in memoriam).

AGRADECIMENTOS

À minha mãe e ao meu pai, pela criação, educação e suporte durante todo o tempo em que estivemos juntos.

À minha madrinha, pela grande ajuda nos momentos difíceis, sendo como uma segunda mãe.

Ao Prof. Dr. Marcelo N. P. Carreño, à Profa. Dra. Inês Pereyra e ao Prof. Dr. Marco Alayo, pela excelente orientação e conselhos dados, que nortearam de forma maestral esta dissertação.

Aos amigos Fabio Colombo e Murilo Mielli, pelas valorosas e frutíferas discussões, que ajudaram e muito neste trabalho.

RESUMO

Este trabalho aborda a utilização do método de Lattice Boltzmann como ferramenta de simulação para a área de dinâmica de fluidos. Além disso, apresenta o programa LBSim desenvolvido durante o período de pesquisa, construído para ser flexível e extensível (através do emprego de técnicas de orientação a objetos) e com uma interface gráfica mais amigável do que outros projetos semelhantes.

Como resultado deste trabalho, o software LBSim implementa uma série de módulos diferentes que utilizam o método de Lattice Boltzmann como base, permitindo a simulação de casos monofásico, multifásico, multicomponente, com suporte à gravidade, meios porosos, difusão, transferência de calor e paredes deslizantes.

ABSTRACT

This work approaches the use of the Lattice Boltzmann method as a simulation tool for the fluid dynamics area. Beyond that, it presents the LBSim software, developed during the period of the research and built to be flexible and extensible (through the application of techniques of the object oriented paradigm) and with a graphical interface more friendly than other similar projects.

As a result of this work, the LBSim software implements a series of different modules that uses the Lattice Boltzmann method as a base, allowing the simulations of the following cases: monophasic, multiphase, multicomponent, with support of gravity, porous media, diffusion, heat transfer and moving walls.

SUMÁRIO

1	Introdução	8
1.1	Objetivo	8
1.2	Revisão Bibliográfica	9
1.2.1	Aplicações	9
1.2.2	Lattice Boltzmann no Brasil e no Mundo	9
1.3	Próximos Capítulos	11
2	Dinâmica de Fluidos e Teoria Cinética dos Gases	12
2.1	Primeiros Conceitos	12
2.2	Equações de Navier-Stokes	13
2.3	Número de Reynolds	13
2.4	Teoria Cinética dos Gases e Equação de Boltzmann	14
3	Métodos Computacionais	15
3.1	Níveis de Simulação	15
3.2	Autômatos Celulares	16
3.3	Autômatos Celulares do Tipo Gás de Rede	16
4	Método de Lattice Boltzmann	20
4.1	Definição	20
4.2	Modelo Básico	20
4.2.1	Estrutura de Simulação	21
4.2.2	Função de Distribuição de Partículas	21
4.2.3	Valores Macroscópicos	21

4.2.4	Função de Atualização	23
4.2.5	Condições Iniciais e de Contorno	23
4.2.6	Condições de Parada	28
4.2.7	Resumo do Algoritmo	29
4.2.8	Parâmetros Físicos	29
4.2.9	Número de Reynolds no Método de Lattice Boltzmann . .	31
4.2.10	Avaliação das Forças de Arrasto	32
4.3	Alterações no Modelo Básico	32
4.3.1	Inclusão de Forças Externas	33
4.3.2	Simulações Multifásicas	33
4.3.3	Simulações Multicomponente	34
4.3.4	Meios Porosos	35
4.3.5	Escalar Passivo: Difusão de Solutos	36
4.3.6	Escalar Passivo: Simulações Térmicas	37
4.3.7	Paredes Móveis	38
4.3.8	Modelos de Turbulência	38
5	Implementação	40
5.1	Tecnologias Utilizadas	40
5.2	Orientação a Objetos	41
5.3	Interface do Programa	41
5.4	Principais Funcionalidades	44
5.5	Aspectos Não Funcionais	46
5.5.1	Processamento com Placas Gráficas	48
6	Resultados Obtidos	51
6.1	Aplicações do Modelo Básico	51

6.1.1	Escoamento de Poiseuille	51
6.1.2	Esteira de von Karman	55
6.1.3	Meios Porosos	57
6.1.4	Forças de Arrasto	59
6.2	Aplicações dos Modelos Multifásico e Multicomponente	61
6.2.1	Formação de Bolhas num Ambiente Multifásico	61
6.2.2	Efeitor Capilar	64
6.2.3	Ângulos de Contato no Caso Multicomponente	65
6.3	Aplicações do Escalar Passivo	67
6.3.1	Difusão	67
6.3.2	Simulações Térmicas	70
6.3.3	Convecção de Rayleigh-Bénard	72
6.4	Aplicações em Microfluídica	74
6.5	Aplicações com Paredes Móveis	76
6.5.1	Escoamento de Couette	76
6.5.2	Cavidade com Cobertura Deslizante	78
6.6	Aplicações com Modelo de Turbulência de Smagorinsky	81
7	Conclusão	83
	Referências	85

1 INTRODUÇÃO

O método de Lattice Boltzmann tem ganhado destaque nos últimos anos como boa alternativa para simulação de problemas em dinâmica de fluidos, em contraposição à utilização dos tradicionais métodos de resolução numérica baseados nas equações de Navier-Stokes. Suas principais vantagens são um melhor suporte à simulação de escoamentos mais complexos, com condições de contorno não triviais (meios porosos) ou com várias fases e/ou componentes, e a aplicação da paralelização computacional de maneira simples, o que fornece ao algoritmo uma grande escalabilidade.

1.1 Objetivo

O objetivo deste trabalho é desenvolver um simulador computacional de dinâmica de fluidos que utilize como forma de solução o método de Lattice Boltzmann, apresentando a potencialidade do método através de suas variações e casos de exemplo. Inicialmente, o escopo desta pesquisa estava limitado a aplicações de microfluídica (Zhang, 2011), que é uma das áreas de pesquisa do Laboratório de Microeletrônica da Escola Politécnica da Universidade de São Paulo, mas expandiu-se para outras aplicações no decorrer do trabalho. Um ponto interessante desta técnica é que esta não está limitada ao escopo da dinâmica de fluidos, mas pode ser aplicada em outras áreas, como a acústica (Viggen, 2009), ao eletromagnetismo (Mendoza; Muñoz, 2010) e até mesmo à mecânica quântica (Succi, 2001).

O programa desenvolvido, intitulado LBSim, foi desenvolvido segundo técnicas de orientação a objetos, proporcionando a seus desenvolvedores uma melhor manutenibilidade de seu código, ao tempo que permite sua extensibilidade para outros modelos e variações de uma forma mais simples se comparada com pro-

gramas estruturados. Além disso, dispõe de uma interface gráfica, o que colabora para a sua usabilidade, apresentando ferramentas para carregar os mais diversos modelos estruturais, executar as simulações e extrair os resultados obtidos, seja por renderizações, gráficos ou tabelas de dados.

1.2 Revisão Bibliográfica

Esta seção trata das aplicações do método de Lattice Boltzmann, tanto no contexto teórico, citando possíveis simulações que podem ser executadas, nos mais diversos ramos do conhecimento, quanto no contexto prático, com o levantamento dos principais softwares construídos mundialmente. O método de Lattice Boltzmann possui uma gama de aplicações bem extensa, como ficará evidente após a leitura desta seção.

1.2.1 Aplicações

As aplicações do método de Lattice Boltzmann são extensas (Succi, 2001): começam em escoamentos simples, com números moderados de Reynolds, passando por meios porosos, com geometria complexa; indo para casos turbulentos, com a utilização do modelo de Smagorinsky ou o uso de limitadores por entropia. Além disso, há outras aplicações para casos de fluxos reativos (como combustão e formação de padrões presentes em biologia), fluidos multifásicos, inclusão de partículas móveis, como as presentes em colóides e polímeros, além de fenômenos como transporte de neve e deposição. Além disso, pode ser adaptado para simular transporte de calor e fenômenos de mecânica quântica, uma vez que as equações dessa área apresentam semelhança com as presentes em dinâmica de fluidos. Outras aplicações do método de Lattice Boltzmann incluem simulações de águas rasas (Zhou, 2004), fenômenos acústicos (Viggen, 2009), microfluídica (Kanti De, 2008) e medicina, analisando questões como a respiração humana (Krause, 2010) e hemodinâmica (Mazzeo, 2009).

1.2.2 Lattice Boltzmann no Brasil e no Mundo

A pesquisa sobre o método de Lattice Boltzmann tem crescido significativamente nos últimos anos, conforme se pode observar no gráfico 1 (Microsoft, 2012).

Figura 1: Gráfico do número de publicações por ano conforme o site Microsoft Academic Search (Microsoft, 2012)

Trata-se de uma técnica relativamente nova, que foi introduzida e começou a ser estudada no fim da década de 80.

No âmbito brasileiro, além do NDS (Núcleo de Desenvolvimento de Software) da Escola Politécnica da USP (Universidade de São Paulo), onde está sendo desenvolvido este trabalho, destacam-se o LNCC (Laboratório Nacional de Computação Científica), o LMPT (Laboratório de Meios Porosos e Propriedades Termofísicas) da UFSC (Universidade Federal de Santa Catarina) e o INF (Instituto de Informática) da UFRGS (Universidade Federal do Rio Grande do Sul). Os trabalhos do LNCC incluem hemodinâmica (Golbert; Blanco; Feijóo, 2009), teoria básica (Golbert, 2007) e animação de fluidos (Judice, 2009). O LMPT possui trabalhos objetivando o estudo da permeabilidade de meios porosos. Já o INF desenvolve trabalhos na área de computação paralela (Schepke, 2007).

No contexto mundial, existem muito mais trabalhos a respeito, como é de se esperar. Um dos principais pesquisadores desta área é o Dr. Sauro Succi, diretor de pesquisa do Instituto de Aplicações Computacionais em Roma, que escreveu um livro bastante abrangente sobre a matéria em questão (Succi, 2001). Um outro livro de destaque é o do Dr. Michael C. Sukop, que trata do método de Lattice Boltzmann numa visão mais pragmática, sem se aprofundar muito nos aspectos teóricos (Sukop; Thorne Jr., 2006).

Dentre os projetos de software, pode-se destacar três projetos: OpenLB (Heuveline; Latt, 2007), Palabos (FlowKit, 2011) (os dois de código aberto) e PowerFLOW (Exa, 2011) (comercial). O projeto OpenLB é liderado pelo Dr. Mathias J. Krause, do Instituto de Tecnologia Karlsruhe. O projeto Palabos tem como coordenados o Dr. Bastien Chopard, que escreveu um livro sobre autômatos celu-

lares e o método de Lattice Boltzmann (Chopard; Droz, 1998). Ambos os projetos são escritos em C++, constituindo-se em bibliotecas e, desse modo, o usuário que deseja simular um determinado cenário necessita conhecer a estrutura desenvolvida para, então, criar um código que a utilize e realize a simulação desejada. PowerFLOW e seu conjunto de aplicativos de ajuda, desenvolvidos pela Exa Corporation, apresenta uma solução completa para o trabalho envolvendo simulações de problemas envolvendo dinâmica de fluidos, indo do pré-processamento, passando pela simulação propriamente dita e chegando no pós-processamento.

1.3 Próximos Capítulos

O restante deste texto está estruturado da seguinte forma: no capítulo 2, alguns conceitos básicos de dinâmica de fluidos e teoria cinética dos gases, que é a base teórica para o método de Lattice Boltzmann, são apresentados. No capítulo 3, são apresentados alguns métodos computacionais que foram os predecessores do método de Lattice Boltzmann (MLB). No capítulo seguinte 4, o MLB é detalhado, com comentários a respeito de seu modelo básico e algumas variações presentes na literatura. No capítulo 5, é apresentado o programa desenvolvido, LBSim, destacando suas características. Finalmente, no capítulo 6, são apresentados os resultados obtidos, definindo as simulações executadas e seus resultados gráficos e numéricos.

2 DINÂMICA DE FLUIDOS E TEORIA CINÉTICA DOS GASES

Dinâmica de fluidos é a área da Mecânica de Fluidos que estuda o escoamento dos fluidos, trabalhando com variáveis como velocidade e densidade e verificando as forças envolvidas e seus efeitos sobre os elementos fluidos.

2.1 Primeiros Conceitos

Uma primeira distinção a ser realizada para esclarecimentos se refere aos termos sólido e fluido. Um sólido pode resistir a forças de cisalhamento, dentro um certo limite, enquanto um fluido, por menor que seja a força, apresentará um movimento devido à presença desta força (White, 1998). Um fluido, ainda, na maioria dos casos, pode ser um líquido ou um gás. Um líquido, devido às suas forças de coesão, mantém seu volume constante a temperatura constante, enquanto o gás se expande no ambiente no qual está contido.

Um outro conceito importante é o do elemento fluido. Elemento fluido é um termo que define uma porção do espaço no qual variáveis como a densidade são determinadas sem perturbação dos detalhes microscópicos do fluido. Quanto menor a porção do espaço tomada como referência, os aspectos microscópicos (dinâmica molecular) passam a ter maior importância na observação. As equações definidas na área de dinâmica de fluidos não consideram os aspectos microscópicos, preocupando-se em estudar os fluidos em seu contexto macroscópico.

Outro ponto importante a destacar é o conceito da viscosidade, que está relacionada com a dificuldade do fluido escoar. Quanto maior a viscosidade, maior a resistência que o fluido fornecerá para atingir uma mesma velocidade na seção perpendicular à direção de escoamento. A equação (2.1) apresenta este conceito, definindo o coeficiente de viscosidade μ com a tensão de cisalhamento τ

e o gradiente de velocidade $\frac{du}{dy}$.

$$\tau = \mu \frac{du}{dy} \quad (2.1)$$

2.2 Equações de Navier-Stokes

A equação de Navier-Stokes se constitui na base da dinâmica de fluidos na medida em que relaciona o movimento dos elementos fluidos com as forças atuantes sobre estes, uma vez que é baseada na segunda lei de Newton. A equação é apresentada na fórmula (2.2), em sua forma indicial, onde ρ é a densidade; $\vec{u}$, a velocidade; $\vec{g}$, o vetor gravidade; p , a pressão; μ , a viscosidade dinâmica; e, finalmente, λ , o segundo coeficiente de viscosidade.

$$\rho \frac{D\vec{u}}{Dt} = \rho \vec{g} - \nabla p + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \text{div} \vec{u} \right] \quad (2.2)$$

Na equação acima, o termo à esquerda está relacionado com a variação temporal do campo de velocidades. À direita, tem-se os termos referentes às forças envolvidas, considerando o efeito da gravidade, o gradiente de pressão e a ação da viscosidade.

2.3 Número de Reynolds

O número de Reynolds (White, 1998) é um dos mais importantes parâmetros adimensionais presentes numa descrição de um escoamento de um fluido. Sua fórmula é apresentada em (2.3), onde ρ é a densidade do fluido; V , a velocidade característica do escoamento; L , o comprimento característico; μ , a viscosidade dinâmica do fluido. A viscosidade cinemática ν é um atributo derivado da relação entre a viscosidade dinâmica e a densidade do elemento.

$$Re = \frac{\rho V L}{\mu} = \frac{V L}{\nu} \quad (2.3)$$

Para escoamentos com número de Reynolds bem menores do que um, o fluido se movimenta bem lentamente, preponderando as forças inerciais. Em escoamen-

tos com números de Reynolds próximos da ordem de um, um escoamento laminar é observado, no qual há uma maior dinâmica observada. Para escoamentos com altos números de Reynolds (bem maiores do que um), observa-se o fenômeno da turbulência.

2.4 Teoria Cinética dos Gases e Equação de Boltzmann

Num gás, as moléculas viajam a diferentes velocidades e, a todo momento, colidem umas com as outras. Considerando que a distância média entre as partículas é muito maior do que o tamanho destas, pode-se considerar as moléculas como estruturas pontuais. Dado isto, pode-se definir uma função de distribuição de probabilidade $f(\vec{x}, \vec{p}, t)$, que é a probabilidade de encontrar uma partícula ao redor da posição $\vec{x}$, com momento $\vec{p}$, no instante t .

Definida a função de distribuição de probabilidades, a equação de Boltzmann (2.4) apresenta a evolução temporal de seus valores, onde K é a força externa agindo sobre as partículas e m é a massa de cada uma das partículas.

$$\partial_t f + u \partial_x f + \frac{K}{m} \partial_u f = Q(f, f) \quad (2.4)$$

A parte esquerda está relacionada com a propagação das partículas, enquanto que a direita representa o efeito de colisão entre estas. A partir da expansão de Chapman-Enskog (Chapman; Cowling, 1970), demonstra-se que a equação de Boltzmann pode ser derivada para a equação de Navier-Stokes. Na equação apresentada, observa-se que o termo de colisão é bem complexo e uma simplificação desta fórmula é dada por BGK (Bhatnagar; Gross; Krook, 1954) (2.5), que possibilitou a implementação LBGK (Lattice BGK) e a popularização do método de Lattice Boltzmann.

$$C_{BGK}(f) = -\frac{f - f^e}{\tau} \quad (2.5)$$

O termo de colisão BGK é dado pela diferença entre o valor de f em relação a um valor de equilíbrio, com um fator do inverso de τ , chamado parâmetro de relaxação.

3 MÉTODOS COMPUTACIONAIS

Este capítulo trata, de forma resumida, dos métodos computacionais que podem ser empregados na solução de problemas que envolvam a dinâmica de fluidos. É apenas uma contextualização para que o leitor possua uma breve noção das abordagens utilizadas para esse fim, não sendo o objetivo apresentar os detalhes de cada um dos métodos. Esta dissertação se foca no método de Lattice Boltzmann.

3.1 Níveis de Simulação

Existe uma clara distinção e, de certa forma, uma hierarquia que agrupa os diferentes algoritmos empregados na solução de problemas de dinâmica de fluidos, os quais podem ser classificados utilizando os seguintes níveis: microscópico, mesoscópico e macroscópico. Expliquemos, pois, estes três tipos. O nível macroscópico é aquele visível ao olho nu, que considera o fluido como um meio contínuo, contendo as variáveis que são facilmente percebidas ao ser humano: densidade e velocidade que podem ser medidas através de instrumentos triviais. O nível microscópico, como o próprio nome sugere, aborda o nível mais detalhado de fluidos, com uma observação e tratamento de suas moléculas. Por último, tem-se o nível mesoscópico, que se situa no meio desses outros dois níveis apresentados.

A tabela 1 apresenta estes três níveis de simulação juntamente com as abor-

Tabela 1: Tabela relacionando os diversos níveis de simulação para problemas em dinâmica de fluidos e exemplos de métodos de simulação

Nível de Simulação	Abordagem para Simulação
macroscópico	meios contínuos (elementos finitos)
mesoscópico	mecânica estatística (Lattice Boltzmann)
microscópico	dinâmica molecular

dagens de simulação mais comuns. Para a escala microscópica, é utilizada a dinâmica molecular, considerando-se as moléculas de um fluido como elementos rígidos e pontuais, que a todo momento se colidem uns com os outros. Na escala macroscópica, um exemplo de técnica utilizada é a de elementos finitos, que basicamente divide o domínio de simulação em pequenas regiões para as quais os valores de velocidade e pressão são calculados numericamente, utilizando como base as equações de Navier-Stokes. Por fim, tem-se a escala mesoscópica, que faz uso do conhecimento de mecânica estatística e a equação de Boltzmann para obter valores macroscópicos de densidade e velocidade a partir de funções de distribuição de partículas.

3.2 Autômatos Celulares

Como o método de Lattice Boltzmann é implementado através da utilização de autômatos celulares, uma breve explicação a respeito deste assunto é realizada. Um autômato celular consiste num conjunto organizado de células que possuem estados e são atualizadas simultânea e iterativamente através de uma regra geral que leva em consideração a vizinhança das células. As células podem ser dispostas em uma, duas ou três dimensões.

Os mais simples autômatos celulares, unidimensionais, com dois estados, foram estudados e catalogados exhaustivamente um a um (Wolfram, 2002). O resultado obtido foi a constatação de que, mesmo se tratando de construções baseadas em regras bem simples, através da execução de alguns desses autômatos, é possível obter, como saída, padrões complexos, alguns repetitivos e outros até aparentemente aleatórios.

3.3 Autômatos Celulares do Tipo Gás de Rede

Os autômatos celulares do tipo gás de rede são autômatos celulares cujo objetivo é simular o comportamento de um gás, através de regras impostas sobre suas células que abrigam moléculas que se propagam e colidem umas com as outras. O primeiro autômato celular do tipo gás de rede foi apresentado em 1973 e foi chamado de HPP (Hardy; Pomeau; de Pazzis, 1973). Trata-se de um autômato celular simples, com uma rede quadrada e vizinhança de quatro células (nordeste,

Figura 2: Regras de colisão para o autômato celular HPP (Wolf-Gladrow, 2000)

sudeste, noroeste e sudoeste). Cada célula contém 4 bits de informação, representando a existência ou não de uma partícula (princípio da exclusão) viajando em cada uma das direções cardeais. O ponto principal do autômato, assim como no método de Lattice Boltzmann, é o conceito de colisão e propagação, que constituem os principais fenômenos quando se analisa o movimento de partículas num gás.

O tratamento de colisão no modelo HPP contempla somente um caso, representado na figura 2. O tratamento da propagação também é imediato: o fato de existir uma partícula viajando para nordeste, por exemplo, faz com que, no passo seguinte, essa partícula continue viajando para nordeste, só que na célula vizinha ao nordeste, desde que não tenha sofrido colisão. Essa lógica se aplica para os outros três sentidos possíveis do modelo. Um exemplo de colisão e propagação, que correspondem a um passo, é mostrado na figura 3.

Devido à sua anisotropia (comparando-se as direções cardeais com as colaterais), o método HPP não representa de forma precisa o comportamento de um gás, como sua difusão. Em 1986, foi proposto um novo método, denominado FHP (novamente, iniciais dos criadores: Frisch, Hasslacher e Pomeau), com um tratamento similar, só que utilizando uma rede hexagonal e, portanto, seis vizinhos diretos. As regras de colisão não são tão simples quanto as presentes no HPP e são apresentadas na figura 4.

De certa forma, o método de Lattice Boltzmann pode ser considerado como uma evolução dos autômatos do tipo gás de rede. Os conceitos de propagação e colisão existem em ambos os modelos. Uma grande diferenciação é que, enquanto os autômatos celulares do tipo gás de rede apresentam o princípio da exclusão,

Figura 3: Ilustração de um passo do autômato celular HPP: colisão e propagação (Wolf-Gladrow, 2000)

Figura 4: Regras de colisão para o autômato celular FHP (Wolf-Gladrow, 2000)

isto é, dada uma posição e sentido, só pode existir uma única partícula (0 ou 1, representando a presença ou ausência de uma molécula, ou seja, um tratamento microscópico), no método de Lattice Boltzmann, esta variável é contínua, podendo assumir qualquer valor real (representando uma densidade de partículas, ou seja, um tratamento mesoscópico). Desta forma, o método de Lattice Boltzmann apresenta uma melhor precisão nos resultados, ao mesmo tempo que dá suporte a mais tipos de cenário, em troca de uma necessidade maior de processamento computacional. Além disso, o método de Lattice Boltzmann é baseado em teorias físicas e suas fórmulas possuem fundamentação teórica, diferentemente dos autômatos celulares de gás de rede, que apresentam regras bem simples e práticas para suas simulações.

4 MÉTODO DE LATTICE BOLTZMANN

O objetivo deste capítulo é introduzir o método de Lattice Boltzmann, apresentando seus conceitos básicos e definindo seu modelo. Após a leitura desta parte da dissertação, o leitor entenderá o processamento realizado em simulações que utilizem o método de Lattice Boltzmann como forma de solução.

4.1 Definição

O método de Lattice Boltzmann pode ser definido como uma técnica de simulação de fenômenos físicos que faz uso de um autômato celular cujas células representam uma porção do espaço, para o qual são atribuídos valores macroscópicos de densidade e velocidade, estes sendo baseados numa função de distribuição de partículas, o qual é atualizada iterativa e conjuntamente com base nas células vizinhas e considerando-se uma função de atualização que depende de uma função de equilíbrio (baseada num gás em equilíbrio).

A definição acima resume, de maneira simples, o método tema desta dissertação. Porém, para uma compreensão melhor do tema, as próximas seções irão tratar do assunto com um maior detalhamento.

4.2 Modelo Básico

Esta seção apresenta o modelo básico do método de Lattice Boltzmann, com a aproximação BGK. Como será visto, as fórmulas apresentadas nesta seção podem ser implementadas computacionalmente sem grandes problemas, evidenciando uma das características deste método: a facilidade de implementação.

4.2.1 Estrutura de Simulação

Um problema da área de dinâmica de fluidos envolve, necessariamente, as variáveis de espaço e tempo. No método de Lattice Boltzmann, cada intervalo entre dois passos de simulação corresponde a um intervalo de tempo real, assim como cada célula está relacionada com um pequeno elemento do espaço. Pode-se executar simulações utilizando o método de Lattice Boltzmann tanto em duas dimensões como em três dimensões.

4.2.2 Função de Distribuição de Partículas

A cada célula de simulação é associado um conjunto de valores que correspondem a uma função de distribuição estatística de partículas com velocidades determinadas previamente de acordo com o modelo adotado. Nesse ponto, é feita uma discretização também em relação à velocidade das partículas. Existem vários modelos que suportam diferentes dimensões. Para o caso 2D, adota-se mais usualmente o D2Q9 (ilustrado na figura 5), onde o último número simboliza a quantidade de velocidades no modelo. Em casos 3D, os mais utilizados são o D3Q15, o D3Q19 e o D3Q27.

Usando o modelo D2Q9 como exemplo, a função de distribuição de partículas é um conjunto de 9 escalares f_i que representam densidades probabilísticas de partículas para 9 vetores velocidade diferentes c_i , cujos módulos são os seguintes: 0 para as partículas em repouso, 1 para as partículas nas direções cardinais (norte, sul, leste e oeste) e raiz de 2 para as orientações colaterais (nordeste, noroeste, sudeste e sudoeste). As fórmulas apresentadas daqui em diante são voltadas para o modelo D2Q9, embora sejam bem semelhantes quando comparadas com suas equivalentes para os modelos 3D.

4.2.3 Valores Macroscópicos

A partir de uma certa distribuição de partículas, conforme a definição apresentada na seção anterior, é possível calcular a densidade e a velocidade macroscópicas de cada célula. Este é o grande diferencial do método em relação aos outros tradicionais: ao invés de resolver as equações de Navier-Stokes através de métodos numéricos, utilizando como variáveis a densidade e a velocidade macros-

Figura 5: Esquema demonstrando os vetores de velocidade do modelo D2Q9 e as densidades probabilísticas de partículas associadas a estes vetores.

cópicas, o modelo de Lattice Boltzmann se baseia numa descrição num nível mais abaixo (mesoscópico) para então chegar ao nível macroscópico.

Exposto isso, define-se a densidade macroscópica ρ como a soma dos valores da distribuição de partículas da célula (4.1). A velocidade macroscópica $\vec{u}$, por sua vez, é a soma vetorial do produto dos valores escalares da distribuição de partículas pelos vetores de velocidade microscópica do modelo ($\vec{c}$, representados na figura 5 para o caso D2Q9), dividida pela densidade macroscópica definida anteriormente (4.2).

$$\rho = \sum_{i=0}^{Q-1} f_i \quad (4.1)$$

$$\vec{u} = \frac{\sum_{i=0}^{Q-1} f_i \cdot \vec{c}_i}{\rho} \quad (4.2)$$

Aqui está uma das características mais importantes do método de Lattice Boltzmann: a utilização de um modelo mesoscópico do qual é possível calcu-

lar os valores macroscópicos e realizar, de certa forma, o caminho inverso (do macroscópico para o mesoscópico). A partir da equação de estado de gás ideal, pode-se definir a pressão no modelo com base na densidade local. A fórmula (4.3) apresenta este equacionamento para o caso D2Q9.

$$P = c_s^2 \rho = \frac{1}{3} \rho \quad (4.3)$$

4.2.4 Função de Atualização

O método de Lattice Boltzmann, como todo autômato celular, possui uma função de atualização que, a princípio, se aplica a todas as células. Posteriormente, outras formas de atualização serão apresentadas, de acordo com o tipo da célula em questão e dependendo do problema que se deseje resolver. A fórmula básica de atualização é apresentada em (4.4).

$$f_i(\vec{x} + \vec{c}_i t, t + \Delta t) - f_i(\vec{x}, t) = \frac{1}{\tau} (f_i^{eq}(\vec{x}, t) - f_i(\vec{x}, t)) \quad (4.4)$$

Na fórmula (4.4), τ é o parâmetro de relaxação da aproximação BGK e f_i^{eq} é a função de equilíbrio, dada pela fórmula (4.5).

$$f_i^{eq}(\rho, \vec{u}) = w_i \rho \left(1 + 3 \vec{c}_i \cdot \vec{u} + \frac{9}{2} (\vec{c}_i \cdot \vec{u})^2 - \frac{3}{2} \vec{u}^2 \right) \quad (4.5)$$

Na fórmula acima, w_i é um peso associado a cada uma das direções do modelo utilizado. No caso D2Q9, os valores de w_i são os seguintes: $4/9$ para $i = 0$, $1/9$ para $i = 1, 2, 3$ e 4 e $1/36$ para $i = 5, 6, 7$ e 8 .

4.2.5 Condições Iniciais e de Contorno

Para possibilitar uma simulação utilizando o método de Lattice Boltzmann, que faz uso de atualizações sucessivas, é necessário estabelecer uma condição inicial. Para tanto, usualmente os valores das funções de distribuição de partículas são configurados para os valores da função de equilíbrio utilizando velocidade macroscópica igual a zero e densidade macroscópica para um valor fixo (no caso mais simples, adimensional, igual a um).

Figura 6: Exemplo de domínio de simulação utilizando condições periódicas para os lados direito e esquerdo do domínio.

Figura 7: Ilustração da condição de contorno do tipo bounce-back.

Em relação às condições de contorno, vários métodos podem ser aplicados, dependendo das condições que se deseja aplicar na simulação. Uma das condições de contorno mais simples é a periódica, ilustrada na figura 6, que considera uma ligação virtual entre os elementos extremos do domínio da simulação. Dessa forma, considera-se conectados os elementos laterais (borda esquerda com borda direita), assim como a borda superior com a inferior. Este tipo de condição de contorno é apropriado para domínios de simulação repetitivos.

Outro tipo de condição de contorno bastante utilizada é a bounce-back. A condição bounce-back, como o próprio nome sugere, faz com que os valores de f_i sejam refletidos de volta, na mesma direção, só que invertendo o sentido de deslocamento. Tal condição implementa o que é chamado de condição de não escorregamento na parede, ou seja, o fluido imediatamente próximo da parede tem velocidade zero. A figura 7 ilustra esse tipo de condição de contorno.

Para os casos de condições de contorno abertas, nas quais há um fluxo de

Figura 8: Ilustração dos valores de densidades probabilísticas que devem ser calculadas de uma forma diferente, respeitando as condições de contorno especificadas.

entrada ou saída do fluido, pode-se utilizar diferentes abordagens. O objetivo das técnicas utilizadas é basicamente calcular os valores das distribuições de partículas (f_i) que ficam indefinidas após o passo de propagação. A figura 8 ilustra estes valores para o caso D2Q9. Como as células de contorno ficam situadas na borda do domínio da simulação e estas não possuem todos os vizinhos, alguns valores de distribuição não são calculados a partir da fórmula de propagação (4.4).

No programa desenvolvido, foram implementados três métodos diferentes. O primeiro deles (Estallo, 2008) se baseia nos valores das funções de equilíbrio para calcular os valores que faltam. Pode-se fixar um valor para a velocidade, u_0 , para as células de contorno, x_{in} , e, nesse caso, a densidade utilizada como parâmetro para a função de equilíbrio é a da célula vizinha, $x_{in} + \Delta x$, (caso mais comum para condições de entrada, equacionada em (4.6)). Também é possível configurar um valor de densidade, ρ_0 , e, nesse outro caso, o valor da velocidade utilizada para calcular os valores de equilíbrio é tomado da célula vizinha, $x_{out} - \Delta x$, (caso mais comum em condições de saída, equacionado em (4.7)). A figura 9 mostra um esquema demonstrando uma aplicação deste tipo de condição de contorno.

$$f_i(x_{in}) = f_i^{eq}(\rho(x_{in} + \Delta x), u_0) \quad (4.6)$$

$$f_i(x_{out}) = f_i^{eq}(\rho_0, u(x_{out} - \Delta x)) \quad (4.7)$$

O segundo método que pode ser utilizado é fixar um fluxo de entrada (através dos valores de velocidade e densidade) e aplicar uma condição de zero gradiente na saída (Yu; Mei; Shyy, 2005). Nesse esquema, para a entrada, é fixado um valor de velocidade e de densidade e calculado o valor de equilíbrio, refletindo-se, ainda,

Figura 9: Esquema demonstrando uma aplicação da condição de contorno do tipo equilíbrio.

os valores fora do equilíbrio (4.8). Na fórmula 4.8, i' representa o sentido contrário de i . Na saída, os valores da célula imediatamente anteriores à célula de contorno são copiados para esta (4.9). A figura 10 apresenta uma ilustração desse tipo de condição de contorno.

$$f_i(x_{in}) = f_i^{eq}(\rho_0, u_0) + (f_{i'} - f_{i'}^{eq}) \quad (4.8)$$

$$f_i(x_{out}) = f_i(x_{out} - \vec{c}_i \Delta t) \quad (4.9)$$

O terceiro e último método que foi implementado fixa velocidade ou pressão (densidade) no contorno desejado e faz o bounce-back das partes de não equilíbrio das distribuições f_i (Zou; He, 1997). As fórmulas de atualização devem considerar a orientação da parede. No caso da velocidade fixa, numa condição de contorno horizontal, com o fluido acima, as distribuições de partículas faltantes podem ser calculadas conforme as equações representadas de (4.10) a (4.13).

Figura 10: Diagrama demonstrando as condições de contorno de fluxo fixado na entrada e zero gradiente na saída.

$$\rho = \frac{1}{1 - u_y} [f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)] \quad (4.10)$$

$$f_2 = f_4 + \frac{2}{3}\rho u_y \quad (4.11)$$

$$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y \quad (4.12)$$

$$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y \quad (4.13)$$

Para o caso de densidade fixa, e uma condição de contorno vertical, com o fluido à direita, utilizam-se as equações descritas de (4.14) a (4.17). Por simetria, pode-se aplicar estas mesmas equações para condições de contorno em outras posições, mudando-se os índices dos f_i .

$$u_x = 1 - \frac{[f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)]}{\rho_{in}} \quad (4.14)$$

$$f_1 = f_3 + \frac{2}{3}\rho_{in}u_x \quad (4.15)$$

$$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x \quad (4.16)$$

$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x \quad (4.17)$$

Figura 11: Diagrama apresentando a condição de contorno de Zou e He.

A figura 11 apresenta um desenho esquemático ilustrando a aplicação deste tipo de condição de contorno, bem como todas as fórmulas envolvidas.

4.2.6 Condições de Parada

Por se tratar de uma técnica iterativa, partindo do estado inicial, é necessária uma certa quantidade de iterações para chegar na solução desejada. Para problemas estacionários, cuja solução não apresenta variação nas variáveis de saída em relação ao tempo, pode-se estabelecer um critério numérico como condição de parada. Em problemas não estacionários, a definição de um critério automatizado de parada não é trivial e, muitas vezes, cabe ao usuário inspecionar manualmente os resultados obtidos.

Nos casos estacionários, um critério de parada pode ser estabelecido com base na densidade macroscópica das células. Pode-se dizer que uma simulação que utilize o método de Lattice Boltzmann consiste basicamente na atualização dos valores de distribuição de partículas a cada iteração, o que acarreta uma variação na densidade macroscópica a cada momento. Pode-se definir uma variação média como sendo a média das variações absolutas de densidade macroscópica das células em relação à iteração anterior. A condição de parada adotada neste trabalho consiste em parar a simulação quando esta variação média dividida pela média das densidades for menor do que um certo valor, determinado previamente pelo usuário.

4.2.7 Resumo do Algoritmo

Uma visão geral sobre como o algoritmo utilizado no método de Lattice Boltzmann funciona, pode ser obtida no diagrama apresentado na figura 12. Observa-se a presença de duas fases distintas: colisão e propagação, que, em códigos que utilizem técnicas de paralelismo, por simplicidade, devem estar separadas por uma barreira, ou seja, a propagação de uma célula só pode ser computada após o processamento de colisão de todas as células, assim como a colisão de uma célula só pode ser executada após o cômputo da propagação de todas as células.

O fluxo normal apresentado no diagrama se aplica somente às células que representam o fluido. Para as células referentes à parede e às condições de contorno, um tratamento especial deve ser aplicado, conforme já comentado nas seções prévias. Uma outra observação importante é referente à equação (4.4): parte é utilizada na fase de colisão e parte na fase de propagação.

4.2.8 Parâmetros Físicos

Até aqui, as fórmulas apresentadas foram expostas na forma adimensional, considerando as distâncias entre células e o tempo entre os passos iguais a um. Em (Sukop; Thorne Jr., 2006), estas variáveis são chamadas, respectivamente, de unidade de lattice (lattice unit, lu) e passo de tempo (time step, ts). Porém, para resolver problemas do mundo real, é necessário definir os parâmetros físicos para que se possa relacionar os resultados da simulação com o mundo real.

Esse procedimento, utilizando-se o método de Lattice Boltzmann mais básico (LBGK), consiste no estabelecimento do Δx e do Δt . Explicuemos, então estes dois parâmetros. O valor de Δx corresponde ao espaçamento entre duas células consecutivas, enquanto Δt é o intervalo entre duas iterações consecutivas. Estes parâmetros devem ser ajustados de forma compatível com a fórmula (4.18), tomando-se o cuidado para que o valor de τ não seja próximo de 0,5, uma vez que isso causa erros numéricos na simulação.

$$\nu = \frac{1}{3} \left(\tau - \frac{1}{2} \right) \frac{(\Delta x)^2}{\Delta t} \quad (4.18)$$

A fórmula (4.18) relaciona o parâmetro de relaxamento do termo de colisão

Figura 12: Diagrama de atividades mostrando alguns detalhes do algoritmo utilizado no método de Lattice Boltzmann.

Tabela 2: Tabela apresentando a aplicação de parâmetros físicos no método de Lattice Boltzmann

Substância	$\nu(m^2/s)$	$\Delta x(m/s)$	τ	$\Delta t(s)$	$V_{max}(m/s)$
água a 20°C	1×10^{-6}	1×10^{-6}	1,0	167×10^{-9}	600×10^{-3}
água a 20°C	1×10^{-6}	1×10^{-3}	1,0	167×10^{-3}	600×10^{-6}
água a 20°C	1×10^{-6}	1×10^{-3}	0,6	$33,3 \times 10^{-3}$	3×10^{-3}
água a 20°C	1×10^{-6}	1×10^{-3}	0,51	$3,33 \times 10^{-3}$	30×10^{-3}
ar a 0°C	$13,5 \times 10^{-6}$	1×10^{-6}	1,0	123×10^{-6}	8,1
ar a 0°C	$13,5 \times 10^{-6}$	1×10^{-3}	1,0	$12,3 \times 10^{-3}$	$8,1 \times 10^{-3}$
óleo SAE 30 a 20°C	250×10^{-6}	1×10^{-6}	1,0	667×10^{-12}	150
óleo SAE 30 a 20°C	250×10^{-6}	1×10^{-3}	1,0	667×10^{-12}	150×10^{-3}

(τ , presente no lado direito da equação (4.4)) com a viscosidade cinemática do fluido a simular e os parâmetros físicos Δx e Δt . No ajuste dos parâmetros Δx e Δt , deve-se atentar para os valores da velocidade macroscópica no fluido, que devem ser mantidos pequenos. Recomenda-se que os valores se mantenham dentro do limite de 0,1 (adimensional), para que seja respeitada a desconsideração dos efeitos da compressibilidade no modelo de Lattice Boltzmann (Sukop; Thorne Jr., 2006).

A tabela 2 apresenta alguns exemplos práticos de valores físicos aplicados de acordo com as fórmulas apresentadas. Pode-se observar que substâncias mais viscosas, como o óleo nesse caso, apresentam uma menor restrição quanto às velocidades da simulação. Além disso, quanto maior a escala espacial (Δx), menor é a velocidade limite do modelo. Para uma maior estabilidade na simulação, recomenda-se o uso de τ igual a um. Porém, isso nem sempre é possível, considerando-se a simulação que se deseja simular. Portanto, este parâmetro deve ser ajustado, de forma a atender aos requisitos desejados, lembrando de evitar valores muito próximos de 0,5.

O mesmo raciocínio pode ser aplicado para a massa. Define-se um Δm , que corresponde à massa de uma célula que apresenta $\rho = 1$. Após definidas estas unidades de espaço (Δx), tempo (Δt) e massa (Δm), pode-se calcular os valores físicos com base nestas unidades de referência, através de uma análise dimensional.

4.2.9 Número de Reynolds no Método de Lattice Boltzmann

No âmbito do método de Lattice Boltzmann, pode-se calcular o número de Reynolds utilizando-se os valores adimensionais da própria simulação ao invés de

se utilizar valores reais. Dessa forma, a velocidade característica é dada em $lu.ts^{-1}$, o comprimento característico, em lu 's e a viscosidade cinemática, utilizando-se a equação (4.18), é dada por $\frac{1}{3}\left(\tau - \frac{1}{2}\right)\frac{lu^2}{ts}$. Assim, a fórmula do número de Reynolds utilizando-se os valores adimensionais presentes numa simulação que utilize o método de Lattice Boltzmann pode ser escrita conforme a equação 4.19, onde v_{lb} é dado em lu 's por ts e L_{lb} , em número de células do comprimento característico do problema.

$$Re = \frac{v_{lb}L_{lb}}{\frac{1}{3}\left(\tau - \frac{1}{2}\right)} \quad (4.19)$$

4.2.10 Avaliação das Forças de Arrasto

Para o cálculo das forças de arrasto, existem dois principais métodos que podem ser utilizados conjuntamente com o método de Lattice Boltzmann: integração de tensões e troca de momento (Yu et al., 2003). O primeiro método é utilizado em simuladores baseados nas equações de Navier-Stokes e encontra algumas dificuldades quando aplicado no método de Lattice Boltzmann. O segundo, mais apropriado para o MLB, é facilmente implementado através da fórmula (4.20).

$$\vec{F} = \sum_{x_b} \sum_{\alpha \neq 0} \vec{c}_\alpha [f_\alpha(x_f) + f_\alpha(x_b)] \quad (4.20)$$

A ideia deste método considera a função de distribuição de partículas, aplicando a conservação de quantidade de momento. Dessa forma, em cada iteração, existem, estatisticamente, f_α partículas com uma certa massa colidindo com a parede e f_i partículas com a mesma massa retornando desta. Aplica-se a conservação de quantidade de momento e conclui-se que a parede sofre a força dada pela fórmula (4.20).

4.3 Alterações no Modelo Básico

Esta seção apresenta diversas possíveis alterações no modelo de Lattice Boltzmann básico exposto na seção anterior, demonstrando a ampla aplicabilidade do método. Ressalta-se que os itens aqui apresentados não compõem uma lista

exaustiva, tratando-se apenas dos temas estudados durante a pesquisa realizada. Como o método de Lattice Boltzmann tem ganhado popularidade nos últimos anos, vários artigos científicos têm sido publicados até então, alguns deles apresentando novas abordagens e técnicas que permitem a aplicação em outras áreas do conhecimento.

4.3.1 Inclusão de Forças Externas

O método de Lattice Boltzmann permite a adição da ação de forças externas que atuam sobre os elementos fluidos. Dessa forma, efeitos como a gravidade, a separação de fases e a imiscibilidade de dois fluidos podem ser simulados de forma relativamente simples. Uma das formas possíveis de incluir o efeito dessas forças externas é através da variação da velocidade macroscópica utilizada como parâmetro de cálculo da função de equilíbrio. Nesse sentido, a mudança da velocidade é dada pela fórmula (4.21).

$$\vec{u}^* = \vec{u} + \frac{\tau \vec{F}}{\rho} \quad (4.21)$$

O efeito da gravidade é um dos exemplos mais simples de força externa, que pode ser implementada a partir da expressão (4.22), onde o vetor $\vec{c}$ aponta para o sentido da aceleração da gravidade. Observe-se que existe o coeficiente g , que pode ser ajustado para refletir a aceleração da gravidade ou até mesmo diminuir ou aumentar sua intensidade.

$$\vec{F} = \rho g \vec{c} \quad (4.22)$$

4.3.2 Simulações Multifásicas

A separação de fases ocorre pela ação de forças de atração entre as partículas de uma mesma fase. Através da inclusão de uma nova força na simulação, conforme comentado na seção anterior, é possível adicionar o efeito de separação de fases. Neste caso, a força a ser utilizada é dada por (4.23).

$$\vec{F}(\vec{x}, t) = -G\psi(\vec{x}, t) \sum_i w_i \psi(\vec{x} + \vec{c}_i \Delta t, t) \vec{c}_i \quad (4.23)$$

Na fórmula acima, G é um parâmetro relacionado com a intensidade da força entre regiões com densidades diferentes e w_a é o peso associado a cada uma das direções das velocidades microscópicas do modelo. Para a função $\psi(\vec{x}, t)$, diferentes abordagens podem ser utilizadas. A implementação adotada (Sukop; Thorne Jr., 2006) é apresentada na fórmula (4.24).

$$\psi(\rho) = \psi_0 e^{\left(\frac{-\rho_0}{\rho}\right)} \quad (4.24)$$

No caso de células que envolvam paredes, a expressão acima pode causar inconsistências na simulação. Dessa forma, um tratamento especial deve ser definido para resolver este problema. Para tanto, é definida uma nova força, de adesão, que lida com o contato entre fluido e superfície, apresentada como a fórmula (4.25). Neste caso, $s(\vec{x})$ é uma função que assume valor 0 quando $\vec{x}$ é uma célula de fluido e 1 quando se trata de uma superfície.

$$\vec{F}_{ads}(\vec{x}, t) = -G_{ads}\psi(\vec{x}, t) \sum_i w_i s(\vec{x} + \vec{c}_i \Delta t) \vec{c}_i \quad (4.25)$$

Conforme (Sukop; Thorne Jr., 2006), o parâmetro G_{ads} pode ser configurado de modo a simular diferentes ângulos de contato, entre um fluido e uma superfície. No capítulo de resultados, algumas simulações serão apresentadas mostrando o fenômeno anteriormente descrito.

4.3.3 Simulações Multicomponente

Para a simulação de problemas envolvendo vários componentes (diferentes fluidos), diversas abordagens podem ser utilizadas (Zhang, 2011). As mais comuns são a do pseudopotencial (Shan; Chen, 1993) e a de energia livre (Swift, 1995). A técnica implementada neste trabalho foi a de Shan e Chen e, por isso, este texto se foca nessa abordagem.

O método proposto por Shan e Chen consiste basicamente na utilização de várias distribuições de partículas, uma para cada componente presente na simulação. Dessa forma, é preciso definir a densidade e a velocidade macroscópica considerando mais de um componente. A densidade total é simplesmente a soma das densidades dos componentes e a velocidade é dada pela fórmula (4.26). Além

disso, existe um parâmetro de relaxação para cada componente, τ_σ , relacionado com a viscosidade dos fluidos.

$$\vec{u} = \frac{\sum_\sigma \frac{1}{\tau_\sigma} \sum_i f_i^\sigma \vec{c}_i}{\sum_\sigma \frac{1}{\tau_\sigma} \rho_\sigma} \quad (4.26)$$

Para adicionar o efeito da imiscibilidade e controlar sua intensidade, utiliza-se um componente de força adicional que atua na repulsão entre os fluidos. A fórmula desta força é apresentada em (4.27).

$$\vec{F}_\sigma(\vec{x}) = -G\psi_\sigma(\vec{x}, t) \sum_i w_i \psi_{\sigma'}(\vec{x} + \vec{c}_i \Delta t, t) \vec{c}_i \quad (4.27)$$

Assim como no caso multifásico, pode-se adicionar uma força adicional que está relacionada com a interação do fluido em relação à parede. Tal força é equacionada de acordo com a fórmula 4.28.

$$\vec{F}_{ads}^\sigma(\vec{x}, t) = -G_{ads}^\sigma \rho(\vec{x}, t) \sum_i w_i s(\vec{x} + \vec{c}_i \Delta t) \vec{c}_i \quad (4.28)$$

Observe que há um coeficiente G_{ads} para cada fluido. Note que estes coeficientes permitem simular diferentes ângulos de contato dos fluidos em relação à parede.

4.3.4 Meios Porosos

Para a simulação de meios porosos, nos quais há uma presença maciça de componentes sólidos dispostos de forma irregular, pode-se tratar o problema de duas formas diferentes. A primeira solução consiste em criar uma malha de simulação na qual existem células com tratamento de parede (bounce-back) e fluido, dispostas da mesma forma irregular como ocorre num meio poroso. Nesse caso, não há nenhum tratamento especial a ser realizado, somente o modelo LBGK com a implementação de parede e demais condições de contorno já são suficientes para a simulação.

Uma segunda abordagem (Sukop; Thorne Jr., 2006) considera de forma macroscópica o meio poroso. Dessa forma, a célula como um todo apresenta uma certa permeabilidade, que pode ser ajustada, de forma a dificultar a passagem

do fluido. Nesse caso, não se trata de um elemento de fluido propriamente dito, nem de simples parede (bounce-back). Para implementar esse tratamento, deve-se acrescentar mais um passo após o cálculo dos novos valores da distribuição de partículas. Considerando o resultado da equação 4.4 como $f_i^*(\vec{x} + \vec{c}_i t, t + \Delta t)$, utiliza-se a equação 4.29 para obter os novos valores de atualização.

$$f_i(\vec{x}, t + \Delta t) = f_i^*(\vec{x}, t + \Delta t) + n_s (f_{i'}^*(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) - f_i^*(\vec{x}, t + \Delta t)) \quad (4.29)$$

Nesta equação, n_s está relacionado com a permeabilidade do elemento, variando de 0 (totalmente fluido) para 1 (totalmente sólido). Além disso, i' representa a direção oposta a i .

4.3.5 Escalar Passivo: Difusão de Solutos

Um problema envolvendo um soluto imerso num solvente é tipicamente de dois componentes. Para tratar desse caso, pode-se utilizar duas abordagens diferentes (Sukop; Thorne Jr., 2006). Uma delas é chamada de componente ativo e a outra, componente passivo. A técnica de componente ativo requer a utilização do modelo multicomponente, em cujas simulações as densidades do soluto e do solvente são complementares, ou seja, a densidade somada do soluto e do solvente é sempre a mesma para todos os pontos do sistema. Neste caso, nenhum tipo de tratamento especial é necessário.

No caso do componente passivo, o soluto também é representado com uma segunda distribuição de partículas. A diferença é que esta distribuição não é atualizada com base na função de equilíbrio já apresentada mas sim levando-se em consideração uma fórmula de equilíbrio que depende da velocidade macroscópica da outra distribuição (relativa ao solvente). Portanto, neste cenário, o comportamento do soluto é ditado pelo solvente, daí o nome componente passivo. Esta nova fórmula de equilíbrio é apresentada em (4.30) (Sukop; Thorne Jr., 2006). Nesse caso, os componentes de segunda ordem são desprezados e, portanto, este modelo é apropriado para baixas velocidades.

$$f_{\sigma,i}^{eq} = w_i \rho_\sigma (1 + 3\vec{c}_i \cdot \vec{u}) \quad (4.30)$$

Como existe uma segunda distribuição de partículas, há também um segundo parâmetro de relaxação, τ_σ , associado à função de atualização dessa distribuição. Este parâmetro de relaxação está associado ao coeficiente de difusão através da expressão dada pela fórmula (4.31). Observe que esta equação está baseada em unidades de lattice e, ainda, é análoga à expressão da viscosidade cinemática do fluido, dada pela expressão (4.18).

$$D = \frac{1}{3} \left(\tau_\sigma - \frac{1}{2} \right) \quad (4.31)$$

No modelo escalar passivo, pode-se adicionar o efeito da gravidade, considerando-se a presença do soluto também. Neste caso, uma maior concentração de soluto num local terá uma força gravitacional maior se comparada com outro local com menor concentração do mesmo soluto (neste caso, levando em consideração de que a densidade do fluido é a mesma nos dois locais). Este fenômeno pode ser simulado através da inclusão da força externa formulada em (4.32).

$$\vec{F} = (\rho + \rho_\sigma) \vec{g} \quad (4.32)$$

4.3.6 Escalar Passivo: Simulações Térmicas

A segunda distribuição de probabilidades presente no modelo escalar passivo pode ser utilizada para representar a temperatura. Desse modo, tem-se uma distribuição de partículas da qual pode-se extrair os dados de densidade e velocidade do fluido e uma segunda distribuição cujo valor de densidade (dado pela fórmula (4.1)) representa o valor da temperatura naquela célula.

Em problemas termodinâmicos, existe um adimensional (o número de Rayleigh) que afeta diretamente o resultado das simulações. O número de Rayleigh é dado pela fórmula (4.33), onde g é a aceleração da gravidade; β , o coeficiente de expansão térmica; ΔT , a diferença de temperatura; L , o comprimento característico; ν , a viscosidade cinemática e, finalmente, χ , a difusividade térmica. Abaixo do valor crítico, que depende do problema estudado, prepondera os efeitos da

condução, enquanto que acima deste limiar, a convecção torna-se dominante.

$$Ra = \frac{g\beta\Delta TL^3}{\nu\chi} \quad (4.33)$$

Além do número de Rayleigh, um outro adimensional é importante em simulações térmicas: trata-se do número de Prandtl (Pr), definido como a relação entre a viscosidade cinemática do fluido (ν) e sua difusividade térmica (α): $Pr = \frac{\nu}{\alpha}$, onde $\alpha = \tau_\sigma - 0,5$, sendo que τ_σ é o parâmetro de relaxação associado à segunda distribuição de probabilidades (Sidik; Rosdzimin, 2008).

4.3.7 Paredes Móveis

Através de modificações no tratamento das paredes do domínio de simulação, é possível simular também paredes móveis, ou seja, paredes que transferem momento ao fluido (Yu et al., 2003). Tal tratamento é realizado através da inclusão de um novo termo no cálculo dos novos valores de f_i nas células vizinhas à parede, como apresentado na equação (4.34), onde i indica o sentido de uma célula de fluido para uma de parede no domínio da simulação, i' , o sentido oposto, o índice w é relacionado com a célula de parede e $\vec{u}_w$ é a velocidade da parede. O valor de ρ_w seria a densidade na parede, mas como esta não pode ser obtida, utiliza-se ρ_f , que é a densidade do elemento fluido no sentido i' , assumindo baixa compressibilidade (Yu et al., 2003)..

$$f_{i'}(\vec{x}_w, t + \Delta t) = f_i(\vec{x}_w, t + \Delta t) + 6w_i\rho_w\vec{c}_i \cdot \vec{u}_w \quad (4.34)$$

4.3.8 Modelos de Turbulência

Um dos modelos de turbulência que podem ser acoplados ao método de Lattice Boltzmann é o modelo LES (Large Eddy Simulation) com o equacionamento de Smagorinsky (Dupuis, 2002). Este modelo trata o efeito dos vórtices de menor escala como uma viscosidade adicional, a viscosidade turbulenta. Desse modo, em cada elemento fluido, a viscosidade efetiva é calculada como a soma da viscosidade

do fluido com esta viscosidade turbulenta, dada pela fórmula (4.35) (Dupuis, 2002).

$$\nu_t = \left(C_{smago}\right)^2 |S| \quad (4.35)$$

Na equação (4.35), C_{smago} é o coeficiente de Smagorinsky, que varia conforme o problema estudado e $|S| = \sqrt{2S_{\alpha\beta}S_{\alpha\beta}}$ é o módulo do tensor de taxa de deformação, dado por $S_{\alpha\beta} = 1/2(\partial_\beta u_\alpha - \partial_\alpha u_\beta)$. Para calcular este tensor no modelo de Lattice Boltzmann, utiliza-se a equação (4.36) (Dupuis, 2002).

$$S_{\alpha\beta} = -\frac{3}{2\rho\tau_{tot}}\Pi_{\alpha\beta}^{(1)} = -\frac{3}{2\rho\tau_{tot}} \sum_i c_{i\alpha}c_{i\beta} (f_i - f_i^{eq}) \quad (4.36)$$

Na fórmula (4.36), τ_{tot} é o fator de relaxação associado à viscosidade total. Com as expressões acima dadas, pode-se reescrever este fator de acordo com a fórmula apresentada em (4.37). Nesta fórmula, $\mathcal{Q} = \Pi_{\alpha\beta}^{(1)}\Pi_{\alpha\beta}^{(1)}$ (Dupuis, 2002).

$$\tau_{tot} = \frac{1}{2} \left(\sqrt{\tau^2 + 9(C_{smago})^2 \frac{\sqrt{8\mathcal{Q}}}{\rho}} + \tau \right) \quad (4.37)$$

5 IMPLEMENTAÇÃO

O objetivo deste capítulo é apresentar o software desenvolvido neste trabalho: o LBSim (Lattice Boltzmann Simulator). Aspectos referentes à implementação são apontados com maiores detalhes a seguir, fornecendo ao leitor uma boa noção da solução adotada.

Durante o período de pesquisa, foi desenvolvido um site para divulgar o trabalho, que foi ao ar em julho de 2011. Já conta com mais de 2000 visitas até o momento, com acessos provindos de mais de 50 países e mais de 300 cidades espalhadas pelo mundo. O link do projeto é: <http://aquamarina.lme.usp.br/lbsim>. Lá o projeto é apresentado de uma forma sucinta, sendo possível baixar o programa e exemplos de simulação, além dos que são apresentados neste texto.

5.1 Tecnologias Utilizadas

Para o desenvolvimento do programa em questão, foi utilizado o framework Qt (Blanchette; Summerfield, 2008), da Nokia, e, conseqüentemente, a linguagem de programação C++. A adoção de tal framework possibilita a criação de softwares multiplataforma, ou seja, com o mesmo código fonte, é possível compilar executáveis para os principais sistemas operacionais em vigência atualmente: Windows, Mac OS X e Linux.

A utilização da linguagem C++ proporciona uma boa relação entre desempenho e facilidade de codificação. Nesse sentido, programas escritos em C++ apresentam um bom desempenho se comparados com outras plataformas de desenvolvimento (por exemplo, Java e .Net) ao mesmo tempo que permitem a utilização de modelagem orientada a objetos, o que facilita a manutenção do código, além de propiciar uma melhor organização deste.

5.2 Orientação a Objetos

O programa de simulação foi construído utilizando os conceitos de programação orientada a objetos. Um diagrama de classes simplificado é apresentado na figura 13. Através do emprego das técnicas de herança e polimorfismo, é possível a criação de códigos com uma maior flexibilidade em sua manutenção, além de racionalizar a codificação e fomentar o reaproveitamento de dados e comportamento.

O mecanismo de herança consiste na definição de classes que herdam da sua classe mãe os atributos e métodos a esta pertencentes, podendo adicionar mais atributos e outros métodos, além de poder sobrescrever os métodos herdados, permitindo a criação de uma nova estrutura que reaproveita uma já existente, porém modificando-a para atender às necessidades do negócio. O conceito de polimorfismo está relacionado com a capacidade de se lidar com diferentes tipos de objetos de uma mesma maneira.

Para exemplificar estes conceitos, citemos os casos existentes na modelagem do programa. A abstração chave do LBSim é a classe BaseCell (que poderia ser uma interface em outras linguagens de programação que oferecem este recurso, como Java). Esta classe representa, de forma abstrata, toda e qualquer célula passível de simulação no programa. Várias outras classes herdam dessa classe mãe e são utilizadas para modelar cada um dos tipos diferentes de células necessárias para o domínio da simulação. No entanto, embora existam essas diferenças, cada uma das células passa pelo mesmo tratamento básico, o que evidencia a vantagem do emprego do polimorfismo.

5.3 Interface do Programa

A interface do programa desenvolvido (ver figura 14) possui uma tela principal, onde o resultado gráfico da simulação é apresentado. Além disso, possui dois painéis auxiliares, à direita e à esquerda. O painel lateral esquerdo possibilita ao usuário a configuração de vários elementos de visualização: tipo de renderização (os principais são o módulo da velocidade e da densidade e o campo de velocidade), habilitação e desabilitação de renderização dos elementos (eixos x, y e z; parede, fluido e partículas), configurações para a escala de cores (detalhada a

Figura 13: Diagrama de classes simplificado apresentando as principais classes de implementação do simulador desenvolvido

Figura 14: Interface do programa desenvolvido, com a tela principal, os painéis laterais e uma caixa de diálogo com um gráfico plotado

seguir) e configurações de câmera (zoom e rotação em ângulos de Euler). Por sua vez, o painel lateral direito fornece informações a respeito da simulação, tais como iteração atual, tamanho da rede, arquivo carregado, dados sobre a estabilização da solução e outros dados de desempenho, como tempo decorrido e milhões de células atualizadas por segundo (MLUPS, em inglês, millions of lattice updates per second).

A barra de menus, localizada na parte superior do programa, contém os menus para carregar, recarregar e salvar os modelos de simulação, além de salvar a renderização atual, comandos para iniciar, parar e reiniciar a simulação, bem como ir para o próximo passo e atalhos para as diversas caixas de diálogo existentes no software.

5.4 Principais Funcionalidades

O programa desenvolvido apresenta uma série de funcionalidades de forma a possibilitar ao usuário realizar as simulações, bem como realizar um pré e um pós processamentos básicos. Os parâmetros da simulação são configurados através da interface do programa, permitindo alterações inclusive no meio da simulação. Existem controles para iniciar e parar a simulação, bem como salvar o estado atual da simulação para que seja posteriormente restaurada. Como resultados da simulação, o programa fornece uma renderização gráfica baseada numa escala de cores, a geração de arquivos texto contendo os valores obtidos e uma visualização de gráfico para os casos mais simples. Isto é apenas uma visão geral das funcionalidades do programa desenvolvido. Nos próximos parágrafos, mais detalhes serão apresentados.

Para as simulações em 2D, o domínio de simulação consiste numa matriz de x elementos por y elementos, cada um destes correspondendo a uma célula do método de Lattice Boltzmann. Com o programa desenvolvido, pode-se carregar um modelo estrutural a partir de uma imagem PNG cujos pixels apresentam cores específicas, que estão relacionadas com o tipo de célula a simular, conforme a tabela 3. Este método possibilita ao usuário uma grande flexibilidade na definição das geometrias a ser simuladas, sendo necessário apenas desenhar as imagens PNG que posteriormente serão carregadas pelo programa.

Tabela 3: Tabela com a lista de cores possíveis para o arquivo de entrada do programa

Nome da cor	Código RGB	Tipo de célula
branco	FFFFFF	fluido
vermelho	FF0000	entrada de fluido
preto	000000	saída de fluido
cinza	404040	parede porosa
cinza	808080	parede
cinza	C0C0C0	parede térmica
azul	0000FF	parede deslizante
verde	00FF00	célula vazia

Através de caixas de diálogo, é possível configurar os mais diversos parâmetros da simulação, como τ (parâmetro de relaxação), velocidade de entrada, modelo a utilizar e assim por diante. Carregada a geometria e definidos os parâmetros de simulação, pode-se iniciá-la através de comandos disponíveis no programa, bem como pausar e reiniciar. Existem diferentes tipos de visualização e, enquanto esta é executada, o usuário já pode ir observando os resultados obtidos. Para a maioria das visualizações, é utilizada uma escala de cores, indo das cores mais frias até as mais quentes, conforme a teoria das cores ou, então, considerando-se o espectro de onda visível, do comprimento de onda menor até o maior.

Para auxiliar o usuário na simulação de diferentes casos com variação de parâmetros, o LBSim permite a programação de comandos através de uma interface gráfica. Desse modo, o usuário pode configurar o programa para executar sequencialmente diferentes simulações, variando os parâmetros de cada uma delas de acordo com as suas necessidades. Isso poupa tempo e trabalho de quem utiliza o programa, uma vez que não é mais preciso carregar uma simulação, configurar os parâmetros, executar a simulação, esperar terminá-la para então repetir esses mesmos passos n vezes para os parâmetros desejados. Configura-se uma vez e pode-se deixar o programa executando as n simulações.

Além disso, o programa pode executar em modo servidor. Neste modo, o LBSim fica monitorando uma pasta previamente configurada. Assim que um arquivo é gravado nessa pasta, o programa executa a simulação gravada nesse arquivo, armazenando os resultados da simulação na mesma pasta. Fazendo-se uso de uma pasta compartilhada, torna-se possível realizar simulações remotamente, ou seja, disparar uma simulação num computador diferente do que está se utilizando.

Tabela 4: Tabela comparando o desempenho do programa em duas máquinas diferentes, habilitando e desabilitando o suporte a múltiplos núcleos

Rede	Seq.1(s)	Par.1(s)	Speedup1	Seq.2(s)	Par.2(s)	Speedup2
200 x 50	421	316	133%	358	187	191%
400 x 50	683	484	141%	650	294	221%
400 x 100	1198	797	150%	1246	513	243%
800 x 100	2222	1383	161%	2405	932	258%
800 x 200	4431	2644	168%	4757	1895	251%

5.5 Aspectos Não Funcionais

Em relação aos aspectos não funcionais, três pontos merecem destaque: as questões de desempenho, portabilidade e extensibilidade. A extensibilidade, que está relacionada com a facilidade de acrescentar novas funcionalidades, é garantida através da implementação de um modelo orientado a objetos com alto grau de polimorfismo. Desse modo, novos algoritmos podem ser acoplados de maneira relativamente fácil através da inclusão de novas classes que implementem as interfaces já determinadas, sem que causem efeitos colaterais no código atual, o que coopera também para a manutenibilidade do sistema.

A portabilidade, que é a característica de um sistema poder ser implantado em várias plataformas, é alcançada através da utilização do framework Qt, da Nokia, que permite o desenvolvimento de aplicações para Windows, Linux e Mac OS X. O mesmo código fonte pode ser compilado, utilizando os compiladores adequados, para as diferentes plataformas existentes no mercado.

O aspecto mais importante é a questão do desempenho do programa. O código atual foi construído de modo a utilizar os diferentes núcleos do processador da máquina na qual é executado. Através do uso do framework QtConcurrent (Summerfield, 2010), o processamento das células é automaticamente dividido entre os núcleos do processador (caso exista mais de um). Na tabela 4, são apresentados alguns testes de desempenho ativando-se e desativando-se o uso de concorrência no programa desenvolvido.

Para todos os casos, foi utilizado uma geometria semelhante: paredes na parte superior e inferior, entrada do fluido à esquerda e saída à direita. As simulações foram executadas com 20 mil iterações em dois computadores: um com dois núcleos (caso 1: processador Intel Core 2 Duo 2,26 Ghz, com 3 Gb de memória

RAM) e outro com quatro núcleos (caso 2: processador Intel Core 2 Quad 2,4 Ghz, com 4 Gb de memória RAM). As figuras 15 e 16 apresentam os resultados obtidos, mostrando uma melhoria no desempenho para domínios com tamanhos maiores.

Figura 15: Desempenho medido em MLUPS para dois computadores, ativando-se e desativando-se o suporte a vários núcleos

Figura 16: Speedup aferido no computador com dois núcleos (caso 1) e no computador com quatro núcleos (caso 2)

5.5.1 Processamento com Placas Gráficas

A computação utilizando placas gráficas tem ganhado destaque nos últimos tempos. As GPUs (Graphics Processing Units) atuais possuem centenas de unidades de processamento, que podem ser utilizadas em paralelo para obter ganhos de desempenho consideráveis. Nesse mercado, dois fabricantes se destacam: NVIDIA e AMD/ATI. Para possibilitar o desenvolvimento de aplicações utilizando o poder de processamento dessas placas, cada fabricante disponibiliza uma plataforma de desenvolvimento.

Desse modo, hoje em dia duas plataformas disputam o mercado de desenvolvimento em placas gráficas. São elas: CUDA, plataforma desenvolvida pela NVIDIA, para ser utilizada em suas placas, não suportando placas de outros fornecedores e OpenCL, uma especificação cujo objetivo é padronizar a computação paralela, aplicando-se para os atuais processadores multinúcleo bem como as placas gráficas, tanto da NVIDIA como da AMD/ATI. Visando uma maior compatibilidade com diferentes placas gráficas e para utilizar um padrão aberto, a implementação do código do LBSim voltado para as GPUs fez uso da especificação OpenCL.

Detalhando-se a implementação realizada, foram adicionadas duas classes para lidar com essa situação. A primeira delas é uma classe do tipo *façade* que abstrai a complexidade da especificação OpenCL, fornecendo uma API mais fácil ao desenvolvedor, com métodos para carregar um programa, configurar os parâmetros e executá-lo. A segunda classe, que faz uso da primeira citada, é a que executa o código Lattice Boltzmann propriamente dito. Foram implementados 3 kernels: um para a fase de colisão, outro para propagação e o último para atualização das condições contorno.

Para testar o novo código OpenCL desenvolvido, foi utilizado um computador com processador Intel Core 2 Duo 2,4 Ghz e placa de vídeo NVIDIA GeForce GTS250 (com 128 núcleos CUDA). A mesma simulação citada foi usada, com 20 mil iterações para cada caso. Para efeito de comparação, cada simulação foi executada usando o código original do programa e, logo após, utilizando o novo código OpenCL. Como a especificação OpenCL também suporta processadores com vários núcleos, um outro teste, utilizando um computador com processador Intel Core 2 Duo 2,26 Ghz, foi realizado. Neste caso, foi comparado o desempe-

Tabela 5: Tabela comparando o desempenho do programa (medido em MLUPS) em duas máquinas diferentes, habilitando e desabilitando a implementação OpenCL voltada para CPUs e GPUs com vários núcleos

Rede	Original	GPU	Speedup	Original	CPU	Speedup
200 x 50	0,55613	1,8430	331,40%	0,37192	2,0226	543,81%
400 x 50	0,54973	3,3513	609,63%	0,41471	2,4615	593,55%
400 x 100	0,60668	5,1102	842,32%	0,43364	2,5479	587,57%
800 x 100	0,63675	6,2359	987,17%	0,43899	2,6442	602,33%
800 x 200	0,64048	6,9610	1086,8%	0,43309	2,7358	631,70%

Figura 17: Ganho de desempenho considerando o código em OpenCL, tanto para execução em CPU como GPU.

nho do código anterior do programa em relação desempenho do código OpenCL utilizando a arquitetura de vários núcleos do próprio microprocessador da máquina. Deve-se destacar que o código OpenCL desenvolvido utiliza computação com ponto flutuante de precisão simples, uma vez que a especificação OpenCL não garante suporte de precisão dupla para todas as placas gráficas. O código original do LBSim usa cálculos com ponto flutuante de precisão dupla.

Os resultados obtidos desses dois testes estão expostos na tabela 5, bem como os valores de ganho de desempenho estão plotados no gráfico 17. Pode-se observar significativas melhoras no desempenho, tanto para os casos rodando na CPU como na GPU. Além disso, percebe-se uma alta dependência do tamanho do domínio para o caso executado na GPU, uma vez que, quanto maior o domínio utilizado, há um aumento na velocidade de simulação. Isso se deve ao fato de que,

quanto maior o domínio de simulação, há uma menor proporção de momentos de sincronização do algoritmo paralelizado. Um comentário a respeito desses testes é a queda de desempenho em relação aos testes anteriores, referentes aos núcleos do processador. Observa-se que, para o mesmo processador (Intel Core 2 Duo 2,26 Ghz), os primeiros testes acusaram um desempenho entre 0,6 e 1,2 MLUPS, aproximadamente. Já no teste mais recente, os valores obtidos foram de 0,37 a 0,43 MLUPS. Isso se deve ao fato dos testes utilizando o código OpenCL forem executados com a ferramenta de processamento em lote comentada na seção 5.4, que se revelou como um fator significativo de redução de desempenho. Isso acontece porque, no código implementado, há um constante monitoramento que verifica se a simulação corrente já foi terminada, para o próximo comando ser executado. Esse ponto deve ser melhorado em futuras versões, buscando-se uma solução para esse comportamento.

6 RESULTADOS OBTIDOS

Este capítulo apresenta os resultados obtidos a partir da implementação do software desenvolvido. Como veremos, o programa codificado apresenta uma grande variedade de possibilidades de execução, enaltecendo a potencialidade do método de Lattice Boltzmann para a simulação de problemas na área de dinâmica de fluidos, além de consolidar a modelagem orientada a objetos como ferramenta para a construção de programas flexíveis que permitem um alto grau de extensibilidade.

6.1 Aplicações do Modelo Básico

Esta seção apresenta aplicações do modelo básico de Lattice Boltzmann, com o modelo D2Q9 e aproximação BGK.

6.1.1 Escoamento de Poiseuille

O escoamento de Poiseuille trata de um escoamento laminar de um fluido contido numa tubulação. Após a estabilização do escoamento, o perfil de velocidades desenvolvido numa seção transversal do tubo possui um formato parabólico, indo de zero nas paredes até $\frac{3}{2}$ da velocidade média de entrada. Além disso, é possível observar o efeito de entrada: o fluido entra com um perfil constante de velocidades e há uma região próxima da entrada, na qual existe uma transição para o perfil parabólico.

Para executar uma simulação com o escoamento de Poiseuille, foi utilizado um domínio de 200 x 50 células. Na parte superior e inferior (laterais do tubo), foi configurada uma parede (condição bounce-back), no lado esquerdo, a entrada e, no lado direito, a saída. Na entrada, a velocidade de entrada foi configurada

para 0,02 e, na saída, a densidade foi fixada em 1, com a implementação dessas condições de contorno dada por Zou e He. A simulação foi executada com 20 mil iterações, até obter uma variação de densidade menor do que 10^{-7} . O parâmetro de relaxação utilizado é 1, o que resulta em um número de Reynolds igual a 6, considerando a largura do canal como comprimento característico.

Figura 18: Gráficos gerados pelo programa LBSim para o escoamento de um canal simples. (a) Variação do logaritmo negativo da variação média da densidade. (b) Perfil do módulo da velocidade numa seção transversal do canal, localizada na parte média deste. (c) Perfil da densidade (relacionada com a pressão) ao longo do canal.

Os gráficos gerados pelo LBSim são apresentados na figura 18. No primeiro gráfico (a), que mostra a variação de densidade ao longo do canal, é possível observar o efeito reflexivo das condições de contorno de entrada e saída afetando

Figura 19: Diferentes renderizações para o escoamento de um canal simples. (a) Módulo da velocidade. (b) Densidade (relacionada com a pressão) ao longo do canal, com corte configurado para a parte central. (c) Vetores velocidade aumentados, com corte na seção transversal do canal.

a convergência do método numérico (Estallo, 2008). No segundo gráfico (b), pode-se visualizar o perfil parabólico de velocidades, com a velocidade máxima sendo $\frac{3}{2}$ (0,03) da velocidade de entrada (0,02). Finalmente, no último gráfico (c), pode-se observar a queda linear de pressão ao longo do canal.

Na figura 19, diferentes renderizações da simulação efetuada são apresentadas. Na primeira delas (a), a escala de cores foi utilizada para representar o módulo das velocidades em cada ponto da simulação. É possível observar a faixa de transição do perfil linear de velocidades para o parabólico logo na entrada do canal. Na segunda figura (b), é apresentada uma renderização, com um corte horizontal na parte central do domínio de simulação (outra funcionalidade disponibilizada pelo programa desenvolvido), utilizando a mesma escala de cores, mas representando,

Figura 20: Comparação de resultados entre a teoria e o obtido pelo programa desenvolvido, para o escoamento de Poiseuille.

agora, o módulo da densidade em cada ponto. Finalmente, no último gráfico (c), tem-se uma renderização dos vetores velocidade de uma seção transversal do canal localizada no meio do domínio de simulação.

A figura 20 apresenta uma comparação de resultados entre o previsto pela teoria e o obtido pelo programa desenvolvido LBSim. Nota-se uma grande aderência nos valores numéricos, comprovando a aplicabilidade do método de Lattice Boltzmann para escoamentos desse tipo.

Para números de Reynolds maiores, o perfil de velocidades deixa de ser parabólico, apresentando um formato mais achatado na região mediana. Modificando os parâmetros de velocidade de entrada para 0,1 e τ para 0,51, obtém-se o formato apresentado na figura 21.

Figura 21: Escoamento de Poiseuille para um número de Reynolds maior, no qual o perfil parabólico deixa de existir, apresentando um perfil de velocidades achatado no centro

6.1.2 Esteira de von Karman

A esteira de von Karman é um padrão de vórtices desenvolvidos a partir da presença de um obstáculo num escoamento de fluidos a partir de um certo número de Reynolds. Para baixos números de Reynolds (bem menores do que um), o fluxo obtido é o de Stokes, com um escoamento tipicamente laminar. Na figura 22, é apresentada uma série de simulações realizadas com o LBSim variando-se o número de Reynolds para apresentar os diferentes comportamentos de escoamento. Na figura 23, são exibidos resultados obtidos da literatura para efeitos de comparação (Yu; Yu; Liu, 2012). Como se pode observar, desprezando-se detalhes de renderização, como a escala de cores, os resultados se mostram compatíveis, com a esteira de von Karman surgindo para escoamentos com números de Reynolds acima de 80.

A geometria da simulação é composta por uma condição de contorno de entrada à esquerda, uma condição de contorno de saída à direita, paredes em cima e em baixo e um obstáculo com formato aproximadamente circular no meio do canal com 7 células de diâmetro, situado no meio do canal, a 50 células da entrada. Os números de Reynolds adotados foram: 7, 15, 26, 40, 80 e 100 (Yu; Yu; Liu, 2012), que foram obtidos de acordo com a fórmula (4.19), utilizando a equação (6.1) e variando o parâmetro de relaxação τ . As simulações foram executadas até a iteração de número 5000. Os parâmetros adimensionais utilizados foram os seguintes: 0,1 para velocidade de entrada e 7 para comprimento característico (diâmetro do obstáculo).

Figura 22: Diferentes escoamentos variando-se o número de Reynolds: (a) 7, (b) 15, (c) 26, (d) 40, (e) 80 e (f) 100

$$\tau = \frac{2,1}{Re} + 0,5 \quad (6.1)$$

Figura 23: Escoamentos através de um obstáculo cilíndrico variando-se o número de Reynolds: (a) 7, (b) 15, (c) 26, (d) 40, (e) 80 e (f) 100 (Yu; Yu; Liu, 2012)

Figura 24: Diferentes visualizações de uma simulação com paredes irregulares: (a) modelo PNG que pode ser carregado no LBSim, (b) módulos de velocidade, (c) campo de densidade (relacionado com a pressão), (d) renderização dos vetores de velocidade

6.1.3 Meios Porosos

O método de Lattice Boltzmann permite a simulação de paredes irregulares. Como o domínio é discretizado numa rede de células, basta definir o tipo de cada célula (se é parede ou fluido). Para demonstrar esta possibilidade, foi realizada uma simulação numa rede de 400×100 células, que apresenta paredes impenetráveis com contornos não triviais. A condição de entrada está à esquerda, com densidade 1 e velocidade 0,1 e, além disso, há uma condição de extrapolação na saída, à direita. Nos lados superior e inferior, existem paredes também. O domínio da simulação está apresentado na figura 24.

Nesta mesma figura, pode-se visualizar as diferentes visualizações que o programa desenvolvido fornece. Há uma visualização que considera o módulo da velocidade, outra para a densidade e outra para os vetores velocidade.

Figura 25: Renderizações das simulações utilizando o modelo proposto por (Sukop; Thorne Jr., 2006) para meios porosos permeáveis. Do lado esquerdo, estão apresentados os módulos de velocidade e, no direito, o campo de densidade (relacionado com a pressão). De cima para baixo, o coeficiente de permeabilidade configurado foi: 1; 0,8; 0,6; 0,4; 0,2 e 0.

Para demonstrar a utilização do modelo proposto para meios porosos permeáveis, foi utilizado um modelo estrutural semelhante ao da simulação anterior, retirando-se os elementos porosos próximos das paredes superior e inferior (Sukop; Thorne Jr., 2006). A estrutura utilizada é apresentada na figura 25, item (a). As simulações foram executadas com 10 mil iterações, variando-se o coeficiente da permeabilidade do modelo (n_s), indo de 0 a 1, com passos de 0,2. Os resultados

obtidos estão apresentados na figura 25.

As imagens apresentadas utilizam sempre a mesma escala de cores, com as velocidades variando de 0 a 0,12 e as densidades indo de 1 a 1,5. Os casos extremos apresentam soluções já conhecidas: para $n_s = 0$, a simulação decai para um canal livre, enquanto que, para $n_s = 1$, a simulação apresenta paredes totalmente sólidas, do tipo bounce-back. Como se pode observar, quanto maior o coeficiente de permeabilidade, a pressão de entrada aumenta e os módulos de velocidade nos espaços livres também aumentam. Este modelo proposto por (Sukop; Thorne Jr., 2006) é apropriado para a modelagem macroscópica de elementos porosos. Já o primeiro modelo básico de Lattice Boltzmann, com a utilização de paredes irregulares, é mais apropriado para simulações de meios porosos num nível mais detalhado.

6.1.4 Forças de Arrasto

Para demonstrar uma aplicação do cálculo de forças de arrasto em simulações com o método de Lattice Boltzmann, foi selecionado um escoamento simples ao redor de um obstáculo quadrado, para o qual será avaliada a força de arrasto. Para baixos números de Reynolds, a equação (6.2) pode ser utilizada para calcular o coeficiente de arrasto (Sen; Mittal; Biswas, 2011).

$$C_d = 0,7496 + 10,5767Re^{-0,66}, 2 \leq Re \leq 40 \quad (6.2)$$

O domínio de simulação utilizado foi de 900 x 500 células, com um obstáculo quadrado de 5 células de lado. A figura 26 apresenta um desenho esquemático da simulação, destacando o obstáculo quadrado. A condição de contorno utilizada foi a de Zou e He. O parâmetro de relaxação foi ajustado de acordo com o número de Reynolds desejado, utilizando-se uma velocidade de referência adimensional de 0,1. Para uma convergência apropriada, foram executadas 300 mil iterações. Para o cálculo do coeficiente de arrasto, foi utilizada a equação (6.3), onde F_d é o valor obtido da simulação para a força de arrasto, v é a velocidade de entrada e L é o lado do quadrado.

$$C_d = \frac{F_d}{\frac{1}{2}v^2L} \quad (6.3)$$

Figura 26: Desenho esquemático da simulação para avaliação das forças de arrasto sobre um obstáculo quadrado localizado no meio de um canal. Na renderização principal, estão apresentados os elementos da simulação e, em detalhe, os vetores velocidade do fluido.

Os resultados obtidos estão apresentados na figura 27. Pode-se observar que há uma boa concordância entre estes valores e os resultantes da fórmula (6.2).

Figura 27: Comparação dos coeficientes de arrasto obtidos através do programa LBSim para diferentes números de Reynolds em relação aos valores dados pela equação empírica (6.2).

6.2 Aplicações dos Modelos Multifásico e Multicomponente

Esta seção apresenta exemplos de simulação como aplicação dos modelos multifásico de multicomponente.

6.2.1 Formação de Bolhas num Ambiente Multifásico

Utilizando o modelo LBGK D2Q9, com a inclusão da força para efeitos de simulação multifásica, conforme exposto na seção 4.3.2, é possível simular a formação de bolhas num ambiente multifásico, com a observação do fenômeno de separação de fase. O domínio da simulação é de 100 x 100 células e são utilizadas condições periódicas de contorno.

Figura 28: Evolução da simulação multifásica após diferentes números de iterações: (a) 0, (b) 100, (c) 200, (d) 400, (e) 800, (f) 1600, (g) 3200, (h) 6400, (i) 12800

A simulação é inicializada com uma variação de densidade macroscópica entre 200 e 201, aleatoriamente, e velocidade macroscópica zero. Os parâmetros utilizados da força para efeitos multifásicos são os seguintes: $G = -120$, $\psi_0 = 4$ e $\rho_0 = 200$.

Analizando-se a sequência de imagens apresentada, pode-se perceber nitidamente a formação de bolhas de maior densidade (líquido) num ambiente de menor densidade (vapor). Na última iteração realizada (12800), os valores de densidade macroscópica apresentaram variação entre 86 e 530, demonstrando a limitação da diferença entre as densidades de fase, para algo em torno da ordem de 10. Estudos já foram feitos e outros modelos propostos para conseguir atingir uma razão de densidades de fase na ordem de 1000, como é o caso da água (Zheng; Shu; Chew, 2006).

Figura 29: Comparação do perfil de densidade num eixo vertical situado no meio do domínio multifásico, com diferentes valores do parâmetro G : 100, 120, 140 e 160.

Para analisar o efeito do parâmetro G nesse tipo de caso, várias simulações foram executadas alterando-se somente o valor desse parâmetro e mantendo-se os valores dos outros parâmetros utilizados na simulação anterior. Os seguintes valores para G foram considerados: 100, 120, 140 e 160. As simulações foram executadas até a iteração de número 10 mil. As condições iniciais de densidade

são as seguintes: 220 para um quadrado de lado 40 localizado no centro do domínio e 200 para o resto. Os resultados obtidos estão apresentados no gráfico 29. Os valores plotados correspondem ao perfil de densidade retirado de uma linha vertical que passa pelo meio do domínio de simulação. Pode-se observar que o parâmetro G está diretamente relacionado com a diferença de densidade entre as fases: quanto maior G , maior a razão entre as densidades das fases. Além disso, nota-se uma diminuição no tamanho das gotas, uma vez que estas se apresentam mais densas para valores maiores de G .

A mesma análise comentada no parágrafo anterior foi realizada para o parâmetro ψ_0 . Foram executadas 5 simulações, com o parâmetro citado assumindo os seguintes valores: 3,6, 3,8, 4, 4,2 e 4,4. As condições de execução são as mesmas utilizadas no exemplo anterior. Os resultados obtidos estão condensados no gráfico 30. Pode-se observar um resultado semelhante ao visto anteriormente: quanto maior o parâmetro ψ_0 , maior é a densidade da gota, assim como esta apresenta um tamanho menor.

Figura 30: Perfil de densidade para uma simulação multifásica para vários valores de ψ_0 : 3,6, 3,8, 4, 4,2 e 4,4.

Por fim, uma análise semelhante foi realizada para o parâmetro ρ_0 . Utilizando-se os valores 200, 220 e 240, chega-se ao gráfico 31. Pode-se observar que um maior valor para ρ_0 acarreta numa menor diferença de densidade

Figura 31: Perfil de densidade para uma simulação multifásica variando-se o parâmetro ρ_0 : 200, 220 e 240.

entre as fases: numericamente, há um aumento de densidade para a região menos densa e uma diminuição de densidade na região mais densa. Como pode ser visto, é possível configurar os parâmetros do modelo multifásico de modo a obter diferentes situações de separação de fase, com razões variáveis de densidade.

6.2.2 Efeitor Capilar

Com a configuração do parâmetro G_{ads} do modelo multifásico, conforme apresentado na seção 4.3.2, e a inclusão da gravidade como força externa (seção 4.3.1), é possível simular o efeito da capilaridade.

O domínio da simulação é de 100 x 100 células, com uma parede situada no lado inferior e duas paredes em forma de hastes ocupando a porção central da simulação. São utilizadas condições periódicas de contorno, para evitar efeitos de parede nas laterais e no lado superior. O fluido é inicializado com densidade constante 300. O coeficiente do efeito da gravidade é 0,001. Os parâmetros para o modelo multifásico são os seguintes: $G = -120$, $\psi_0 = 4$, $\rho_0 = 200$ e $G_{ads} = -330$, para um ângulo de contato de 0 graus. As simulações foram executadas até a iteração de número 5000.

Foram executados vários testes, variando a distância entre as hastes verticais.

Figura 32: Efeito capilar observado através da variação da distância entre as duas hastes verticais. A distância é: (a) 20, (b) 10 e (c) 5

Os resultados obtidos são apresentados nas figuras 32. Pode-se perceber que, como esperado, quanto menor a distância entre as hastes, maior é o efeito da capilaridade.

6.2.3 Ângulos de Contato no Caso Multicomponente

Através da configuração dos coeficientes de adesão dos fluidos numa simulação multicomponente, é possível simular diferentes ângulos de contato com a superfície. As simulações executadas foram feitas utilizando um domínio de 200 x 100 células, com uma região de 40 x 20 células, localizada na parte central inferior, contendo concentração de 1,94 para o primeiro componente e 0,06 para o segundo (Huang et al., 2007). Em outras regiões, a concentração é invertida: 1,94 para o segundo e 0,06 para o primeiro. Os resultados obtidos são apresentados na figura 33. A figura 34 apresenta os mesmos resultados, que foram publicados em (Huang et al., 2007). Pode-se observar grande aderência quando as duas simulações realizadas são comparadas.

Figura 33: Diferentes ângulos de contato com a superfície obtidos através da configuração dos parâmetros de adesão dos componentes

Figura 34: Resultados de simulações multicomponente alterando-se os parâmetros de adesão dos componentes, obtidos em artigo publicado (Huang et al., 2007)

6.3 Aplicações do Escalar Passivo

Esta seção apresenta simulações que podem ser executadas utilizando-se o modelo escalar passivo. Aqui, este modelo foi utilizado tanto para a simulação de concentração de solutos como para a modelagem da temperatura.

6.3.1 Difusão

A utilização de uma segunda distribuição que modela um escalar passivo permite a simulação de difusão de concentração de um certo componente. Neste caso, a distribuição principal referente ao fluido mantém sua condição inicial (velocidade zero e densidade fixa). O domínio utilizado é de 200 x 200 células, com condições periódicas de contorno.

A iteração inicial possui um solvente em repouso com densidade constante igual a um e um soluto concentrado num quadrado de 20 x 20 células com densidade também igual a um. A evolução do sistema é apresentada na figura 35. A figura 36 apresenta os perfis de concentração para as iterações 100, 200, 400, 800, 1600 e 3200.

Figura 35: Renderização da concentração num processo de difusão

Figura 36: Diferentes perfis de concentração de uma simulação de difusão

Figura 37: Perfis de concentração para simulações de difusão com diferentes coeficientes aplicados. Os parâmetros utilizados de relaxação associados aos coeficientes de difusão são os seguintes: 0,6, 0,8, 1,2 e 1,4.

Figura 38: Comparação do perfil de concentração obtido pelo LBSim e o valor teórico, para vários instantes de tempo: 100, 200, 400, 800, 1600 e 3200

A figura 37 apresenta o mesmo perfil de concentração apresentado no gráfico anterior, porém, com a variação dos parâmetros de relaxação associados aos coeficientes de difusão. Observa-se que, quanto maior o parâmetro configurado, conseqüentemente, quanto maior o coeficiente de difusão, há uma maior dispersão da concentração para um mesmo instante de tempo.

Para validar as simulações de difusão, comparou-se os resultados obtidos com a teoria, para um caso de um ponto inicial com uma concentração unitária e o resto do domínio vazio. Para este caso, pode-se utilizar a fórmula (6.4) (Crank, 1975), com $C_i = 0$ e $M_0 = 1$. Os resultados obtidos, juntamente com os valores teóricos previstos pela fórmula (6.4), estão apresentados na figura 38.

$$C = C_i + \frac{M_0}{4\pi D t} e^{-\frac{y^2}{4Dt}} \quad (6.4)$$

6.3.2 Simulações Térmicas

Nesta seção, são apresentadas diversas simulações utilizando o modelo escalar passivo para lidar com a temperatura do sistema, assim como o efeito da gravidade como força externa que varia de intensidade de acordo com a temperatura no local. A primeira série de simulações a ser apresentada trata de uma cavidade quadrada cercada por paredes sem escorregamento (ou seja, velocidade do fluido perto da parede igual a zero) (Dixit; Babu, 2006). As paredes superior e inferior estão termicamente isoladas enquanto que as laterais são mantidas com um gradiente de temperatura.

Figura 39: Apresentação gráfica das isotermas da primeira simulação variando-se o número de Rayleigh: (a) 10^3 , (b) 10^4 , (c) 10^5 e (d) 10^6

As simulações foram executadas até a iteração de número 50000, para números de Rayleigh variando de 10^3 até 10^6 , conforme feito também no artigo citado. O número de Prandtl utilizado foi de 0,7, valor característico do ar. O programa LBSim, por padrão, renderiza os resultados obtidos através de uma escala de

cores contínua, ou seja, para cada valor a ser renderizado, o programa faz uma interpolação para descobrir que cor exatamente colocar na tela. Existe a possibilidade de deixar este processo discreto, isto é, utilizar uma gama finita de cores. Esta funcionalidade foi utilizada nas simulações desta seção, para mostrar graficamente o contorno das isothermas. A figura 39 apresenta os resultados obtidos para a simulação em questão, mostrando os valores da densidade do fluido (primeira distribuição de probabilidades) somada com a densidade referente à temperatura (segunda distribuição de probabilidades). A densidade do fluido (adimensional) é um, e os valores de densidade referentes à temperatura nas paredes são 1 e 11.

Figura 40: Contorno das isothermas para uma cavidade sob o efeito da gravidade, com a temperatura da parede esquerda maior do que a a direita, variando-se os números de Rayleigh: (a) 10^3 , (b) 10^4 , (c) 10^5 e (d) 10^6 (Dixit; Babu, 2006)

A figura 40 apresenta um resultado presente na literatura para efeitos de comparação (Dixit; Babu, 2006). Como se pode perceber, os resultados obtidos estão compatíveis se comparados com os presentes no artigo citado.

6.3.3 Convecção de Rayleigh-Bénard

O problema da convecção de Rayleigh Bénard envolve os fenômenos de condução e convecção térmica em fluidos dentro de uma estrutura retangular, sob a ação da gravidade, cercada por paredes, das quais a superior possui uma temperatura menor do que a inferior. Este problema pode ser estudado com base no parâmetro adimensional chamado número de Rayleigh. Conforme este número aumenta, os efeitos de convecção se tornam mais significativos do que os de condução.

A implementação realizada considera β e χ iguais a um (vide fórmula 4.33). O coeficiente de gravidade utilizado foi 10^{-3} . Os domínios de simulação foram dimensionados de forma que a largura fosse sempre 3 vezes a altura da estrutura retangular. O parâmetro de relaxação foi ajustado para 1, o que resulta em $\nu = \frac{1}{6}$. Foram executadas 3 simulações, cada uma com 10 mil iterações, com números de Rayleigh iguais a 100, 1000 e 10000, com larguras de 25, 50 e 100, respectivamente.

A figura 41 apresenta os resultados obtidos, apresentando os mapas de temperatura e os vetores velocidade do fluido. No primeiro caso, com número de Rayleigh igual a 100, pode-se perceber que há uma variação linear da temperatura, predominando o efeito da condução de calor. No segundo caso, com número de Rayleigh igual a 1000, vê-se claramente formar as estruturas de convecção de Rayleigh-Bénard, com uma circulação de fluido ora no sentido horário, ora no sentido anti-horário. Finalmente, o terceiro caso, com um número de Rayleigh mais alto, 10000, pode-se perceber o surgimento de estruturas mais complexas, e o campo de velocidades se torna mais desordenado que o caso anterior.

Figura 41: Renderizações da convecção de Rayleigh-Bénard. Os números de Rayleigh são os seguintes: (a) e (b), 100; (c) e (d), 1000 e (e) e (f), 10000. Nos itens (a, c, e), estão apresentadas a visualização da temperatura utilizando-se uma escala de cores. Nos itens (b, d, f), estão apresentados os vetores velocidade do fluido envolvido.

Figura 42: Zoom da figura 41, item (d), mostrando a rotação dos vetores velocidade na simulação.

6.4 Aplicações em Microfluídica

Para demonstrar uma aplicação na área de microfluídica, consideremos um canal em formato de Y, utilizado, por exemplo, em misturadores. Esta estrutura é composta por dois elementos de entrada à esquerda e um canal de saída à direita. Os canais apresentam $100\ \mu\text{m}$ de largura e a velocidade na entrada é de $1,2\ \text{mm/s}$. A figura 43 apresenta o modelo estrutural da simulação.

Para validar a simulação realizada, a mesma simulação também foi realizada no software comercial COMSOL (COMSOL, 2011). A figura apresentada faz uma comparação entre os resultados obtidos, para vários pontos da simulação. Como se pode observar, os resultados apresentam-se bem semelhantes quando comparados lado a lado.

Figura 43: Renderização dos módulos de velocidade para o canal em Y, acompanhados de gráficos apresentando os módulos de velocidade em seções transversais dos canais. À esquerda, estão apresentados os resultados obtidos com o LBSim, enquanto à direita são apresentados os resultados obtidos através do programa COMSOL (Komori; Mielli; Carreno, 2011)

Figura 44: Renderização de uma estrutura mais complexa utilizando o programa LBSim, com a representação do módulo de velocidade.

Como um outro exemplo ilustrativo, a figura 44 apresenta um exemplo de simulação executada pelo LBSim numa estrutura mais complexa, apresentando o módulo de velocidade utilizando uma escala de cores. A geometria foi gerada através de um programa de desenho vetorial.

6.5 Aplicações com Paredes Móveis

Esta seção apresenta exemplos de simulação utilizando como uma das condições de contorno as paredes móveis, apresentadas na seção 4.3.7.

6.5.1 Escoamento de Couette

O escoamento de Couette é um dos tipos de escoamento mais simples que são encontrados na literatura. Trata-se de um escoamento de um fluido localizado entre duas paredes, uma delas fixa e a outra deslizando com uma velocidade constante. Dadas estas condições, o perfil de velocidades observado numa seção transversal é linear, variando de zero próximo à parede fixa e indo até a velocidade máxima perto da parede deslizando.

Para simular este cenário, foi utilizado um domínio de 200×50 células, no qual estão presentes os seguintes elementos: uma parede fixa preenchendo todo o lado inferior, uma parede deslizando de 180 células centralizada no lado superior e, ao seu lado, nos cantos esquerdo e direito, paredes fixas de 10 células. As condições de contorno são periódicas e a velocidade na parede é 0,02. Após 10 mil passos, obteve-se os resultados apresentados nas figuras 45 e 46.

Figura 45: Renderização do módulo dos vetores velocidade obtidos na simulação realizada

Figura 46: Vetores velocidade para $x = 100$ (meio do canal)

Figura 47: Perfil de velocidades para simulações de escoamentos de Couette com diferentes razões entre comprimento e largura do canal.

Nota-se que o perfil obtido de velocidades não é exatamente linear. Isso se deve ao fato de que o comprimento do canal não é infinito, e suas dimensões não são muito maiores do que a largura do canal. Para chegar a um perfil mais próximo do linear, deve-se, portanto, aumentar o comprimento do canal. Desse modo, a simulação citada foi refeita com comprimentos iguais a 400 e 800. Os resultados gerados estão apresentados na figura 47.

6.5.2 Caverna com Cobertura Deslizante

Uma das aplicações de paredes móveis mais básicas é a caverna com uma cobertura deslizante (em inglês, conhecido como lid-driven cavity). Trata-se de uma caverna, geralmente quadrada, cercada por uma parede em seus cantos laterais e inferior, com uma parede deslizante com velocidade constante na parte superior. Para validar os resultados obtidos, foi utilizado como referência o trabalho de Ghia et. al. (1982). Por esse motivo, o domínio da simulação utilizado foi de 129 x 129 células. A figura 48 apresenta um exemplo do domínio citado, assim como os eixos horizontal e vertical utilizados na validação dos resultados.

As simulações foram executadas com velocidade igual a 0,1. Através da variação do termo de relaxação, seguindo a fórmula (4.19), é possível configurar a simulação com o número de Reynolds desejado, conforme fórmula (6.5).

$$\tau = \frac{38,7}{Re} + 0,5 \quad (6.5)$$

Os números de Reynolds utilizados foram os seguintes: 100, 1000 e 5000. Foi realizado um levantamento da convergência da solução a partir da variação média da densidade conforme mostrado na figura 49. Observa-se que, quanto maior o número de Reynolds, a simulação tende a convergir mais devagar.

A figura 50 apresenta uma série de gráficos contendo uma comparação entre os resultados obtidos com o programa desenvolvido, LBSim, e os obtidos por Ghia et. al. (1982). Como se pode notar, os resultados estão bem próximos entre si, com uma pequena discrepância para números de Reynolds maiores, de acordo com o esperado, uma vez que o método de Lattice Boltzmann básico é apropriado para baixos números de Reynolds.

Figura 48: Esquema apresentando um exemplo de cavidade com cobertura deslizante, bem como os eixos utilizados para validar os resultados numéricos.

Figura 49: Variação média da densidade para as simulações de cavidade com cobertura deslizante, com número de Reynolds iguais a 100, 1000 e 5000

Figura 50: Gráficos comparando os resultados obtidos com os presentes na literatura. À esquerda (a, c, e), velocidades no eixo y (v) numa linha horizontal no centro do domínio. À direita (b, d, f), velocidades no eixo x (u) numa linha vertical no centro do domínio. Os números de Reynolds são os seguintes: (a) e (b): $Re = 100$, (c) e (d): $Re = 1000$, (e) e (f): $Re = 5000$

6.6 Aplicações com Modelo de Turbulência de Smagorinsky

Para demonstrar uma aplicação do modelo de turbulência LES com coeficiente de Smagorinsky, apresentado na seção 4.3.8, a cavidade de cobertura deslizante é revisada, para o caso com número de Reynolds igual a 5000. Para tanto, a mesma simulação apresentada na seção anterior foi refeita, agora com coeficientes de Smagorinsky diferentes de zero. As figuras 51 e 52 apresentam os resultados obtidos das simulações executadas com o modelo LES de Smagorinsky, para os coeficientes 0 (sem modelo de turbulência) e 0,15. Pode-se observar claramente que os resultados com coeficiente de Smagorinsky igual a 0,15 estão mais próximos dos tabelados no artigo de referência (Ghia et. al., 1982).

Figura 51: Perfis de velocidades em y na linha horizontal média do domínio, para coeficientes de Smagorinsky: 0 e 0,15.

Figura 52: Perfis de velocidades em x na linha vertical média do domínio, para coeficientes de Smagorinsky: 0 e 0,15.

Para corroborar a melhora nos resultados, foram calculados dois valores estatísticos. O primeiro deles é o coeficiente de correlação. Para o caso da linha horizontal, o coeficiente de correlação aumentou de 0,997345 (sem o modelo de Smagorinsky) para 0,997568 (com coeficiente de Smagorinsky igual a 0,15). No caso da linha vertical, a variação foi de 0,995962 para 0,997482. Também foram calculadas as somas dos quadrados dos erros. No caso horizontal, este valor caiu de 0,036963 para 0,014343; enquanto que no caso vertical, teve-se uma redução de 0,033413 para 0,017449.

7 CONCLUSÃO

O desenvolvimento deste mestrado teve diversas etapas. Num primeiro momento, conceitos básicos sobre o método de Lattice Boltzmann, bem como matérias relacionadas, foram estudados, ao mesmo tempo em que um esforço inicial de desenvolver o código em C++ na plataforma Qt com renderização OpenGL era empregado. Após esse estágio inicial, iniciou-se uma pesquisa mais ampla com o intuito de descobrir o potencial do MLB, desenvolvendo novos modelos de simulação, que resultaram no extenso rol de alterações no modelo básico do MLB apresentado neste texto. Numa etapa final, alguns aspectos adicionais foram implementados, como o suporte à computação utilizando placas gráficas e processamento em lote.

O método de Lattice Boltzmann é uma boa alternativa para a simulação dos mais diversos escoamentos existentes, como se pode observar nesta dissertação, oferecendo a grande vantagem de seu código ser paralelizado de forma relativamente fácil, além de dar suporte a modificações que ampliam a gama de cenários possíveis de simulação. Trata-se de um método numérico recente, mas que vem sendo adotado progressivamente, tanto pelo ambiente acadêmico (constatado através do número crescente de publicações na área), quanto pela indústria (a prova disso é a utilização de programas comerciais em projetos de grandes empresas).

O desenvolvimento do LBSim, programa de simulação que utiliza o método de Lattice Boltzmann, proporcionou aos pesquisadores uma forma prática de analisar a teoria, através da construção de modelos de simulação, sua execução e geração dos resultados obtidos. Além disso, trata-se de um programa inovador dentre os softwares existentes não comerciais, que oferece ao usuário uma interface gráfica mais amigável, com uma renderização dos resultados conforme estes são processados. Uma outra característica interessante é a utilização de técnicas

de orientação a objetos, que permitem, numa visão de futuro, extensões que possam ser acopladas ao código existente sem causar grandes dificuldades ao desenvolvedor. Para complementar os pontos positivos apresentados, não se pode deixar de observar que, até onde é de nosso conhecimento, se trata do único projeto nacional, com o intuito de desenvolver um software nessa área utilizando o método de Lattice Boltzmann.

A pesquisa realizada durante este trabalho revelou que o método de Lattice Boltzmann apresenta uma grande riqueza no que se refere aos seus conceitos teóricos e à sua aplicabilidade em problemas do mundo real. Assim, embora diversos modelos de simulação tenham sido implementados, vários outros poderão ser acrescentados à implementação realizada de forma a incrementar ainda mais o potencial do programa desenvolvido, bem como outros aspectos do programa poderão ser melhorados ou introduzidos.

O primeiro aspecto que merece destaque é a questão da paralelização. Como já comentado na seção anterior, o LBSim faz uso dos vários núcleos do computador no qual é executado. Essa característica é importante nos dias de hoje porque os computadores mais atuais têm evoluído através do acréscimo de mais núcleos, ao invés do aumento do seu clock. Além disso, o programa implementado já possui um suporte básico para processamento em placas gráficas, aproveitando o grande poder de processamento desses componentes, como pode ser visto em seções anteriores. Ainda é preciso fazer algumas modificações para otimizar o desempenho do programa, bem como adaptá-lo para lidar com várias placas gráficas ao mesmo tempo.

Um outro passo importante é a alteração do código para este suportar também modelos em 3D. Conforme comentado na seção 4.2.2, existe mais de um modelo que pode ser adotado para simulações em 3D. Além disso, a inclusão desta funcionalidade acarreta em alterações na renderização da simulação, bem como no pré e pós processamento.

REFERÊNCIAS

- Bhatnagar, P. L.; Gross, E. P.; Krook, M. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. Phys. Rev., American Physical Society, v. 94, p. 511–525, May 1954.
- Blanchette, J.; Summerfield, M. C++ GUI Programming with Qt 4. Stoughton: Prentice Hall, 2008.
- Chapman, S.; Cowling, T. G. The mathematical theory of nonuniform gases. Cambridge: Cambridge University Press, 1970.
- Chopard, B.; Droz, M. Cellular Automata Modeling of Physical Systems. Cambridge: Cambridge University Press, 1998.
- COMSOL. Multiphysics Modeling and Simulation Software. nov 2011. Disponível em: <<http://www.comsol.de>>.
- Crank, J. The Mathematics of Diffusion. 2nd. ed. Oxford: Clarendon Press, 1975.
- Dixit, H. N.; Babu, V. Simulation of high rayleigh number natural convection in a square cavity using the lattice boltzmann method. International Journal of Heat and Mass Transfer, v. 49, p. 727–739, 2006.
- Dupuis, A. From a lattice Boltzmann model to a parallel and reusable implementation of a virtual river. Tese (Doutorado) — Université de Genève - Département d'informatique, 2002.
- Estallo, S. I. Computational Gas Dynamics with the Lattice Boltzmann Method. Tese (Doutorado) — Universidad Zaragoza, 2008.
- Exa. PowerFLOW. out 2011. Disponível em: <<http://www.exa.com>>.
- FlowKit. Palabos. out 2011. Disponível em: <<http://www.palabos.org>>.
- Golbert, D. R. Modelos de Lattice-Boltzmann aplicados à simulação computacional do escoamento de fluidos incompressíveis. Dissertação (Mestrado) — Laboratório Nacional de Computação Científica, 2007.
- Golbert, D. R.; Blanco, P. J.; Feijóo, R. A. Lattice boltzmann simulations in computational hemodynamics. In: Congresso IberoLatinoAmericano de Métodos Computacionais em Engenharia CILAMCE. Buzios: [s.n.], 2009.
- Hardy, J.; Pomeau, Y.; de Pazzis, O. Time evolution of a two dimensional model system. i. invariant states and time correlation functions. Journal of Mathematical Physics, v. 14, n. 12, p. 1746–1759, 1973.

Heuveline, V.; Latt, J. The OpenLB project: an open source and object oriented implementation of lattice Boltzmann methods. Int. J. Mod. Phys. C, v. 18, p. 627–634, 2007.

Huang, H. et al. Proposed approximation for contact angles in shan-and-chen type multicomponent multiphase lattice boltzmann models. Physical Review E, v. 76, p. 066701, 2007.

Judice, S. F. P. P. Animação de Fluidos via Modelos do tipo Lattice Gas e Lattice Boltzmann. Dissertação (Mestrado) — Laboratório Nacional de Computação Científica, 2009.

Kanti De, A. Numerical Modeling of Microscale Mixing using Lattice Boltzmann Method. Tese (Doutorado) — Virginia Polytechnic Institute - State University, 2008.

Komori, F. S.; Mielli, M. Z.; Carreno, M. N. P. Simulator for microfluidics based on the lattice boltzmann method. ECS Transactions, ECS, v. 39, n. 1, p. 461–468, 2011.

Krause, M. J. Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers. Tese (Doutorado) — Karlsruhe Institute of Technology, 2010.

Mazzeo, M. D. Lattice-Boltzmann Simulations of Cerebral Blood Flow. Tese (Doutorado) — Department of Chemistry - University College London, 2009.

Mendoza, M.; Muñoz, J. D. Three-dimensional lattice boltzmann model for electrodynamics. Physical Review E, v. 82, p. 056708, 2010.

Microsoft. Microsoft Academic Search. abr 2012. Disponível em: <<http://academic.research.microsoft.com>>.

Schepke, C. Distribuição de Dados para Implementações Paralelas do Método de Lattice Boltzmann. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2007.

Sen, S.; Mittal, S.; Biswas, G. Flow past a square cylinder at low reynolds numbers. International Journal for Numerical Methods in Fluids, John Wiley Sons, Ltd., v. 67, n. 9, p. 1160–1174, 2011.

Shan, X.; Chen, H. Lattice boltzmann model for simulating flows with multiple phases and components. Phys. Rev. E, v. 47, p. 1815–1819, 1993.

Sidik, N. A. C.; Rosdzimin, A. R. M. Simulation of natural convection heat transfer in a enclosure using lattice boltzmann method. Jurnal Mekanikal, p. 42–50, December 2008.

Succi, S. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. New York: Oxford University Press, 2001.

Sukop, M. C.; Thorne Jr., D. T. Lattice Boltzmann Modeling: An introduction for geoscientists and engineers. Berlin: Springer, 2006.

Summerfield, M. Advanced Qt Programming: Creating Great Software with C++ and Qt 4. [S.l.]: Prentice Hall, 2010.

Swift, M. R. Lattice boltzmann simulation of nonideal fluids. Phys. Rev. Lett., v. 75, p. 830–833, 1995.

Viggen, E. M. The Lattice Boltzmann Method with Application in Acoustics. Dissertação (Mestrado) — Department of Physics - NTNU, 2009.

White, F. M. Fluid Mechanics. Boston: MCGraw-Hill, 1998.

Wolf-Gladrow, D. A. Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction. Berlin: Springer, 2000.

Wolfram, S. A New Kind of Science. Michigan: Wolfram Media, 2002.

Yu, D. et al. Viscous flow computations with the method of lattice boltzmann equation. Progress in Aerospace Sciences, v. 39, p. 329–367, 2003.

Yu, D.; Mei, R.; Shyy, W. Improved treatment of the open boundary in the method of lattice boltzmann equation: general description of the method. Progress in Computation Fluid Dynamics, An International Journal, v. 5, n. 1/2, p. 3 – 12, 2005.

Yu, J.; Yu, S.; Liu, X. Numerical simulation on mean flow past a circular cylinder based on the lattice boltzmann method. Applied Mechanics and Materials, v. 105-107, p. 2307–2310, 2012.

Zhang, J. Lattice boltzmann method for microfluidics: models and applications. Microfluid Nanofluid, v. 10, n. 1, p. 1–28, 2011.

Zheng, H. W.; Shu, C.; Chew, Y. T. A lattice boltzmann model for multiphase flows with large density ratio. J. Comput. Phys., Academic Press Professional, Inc., San Diego, CA, USA, v. 218, p. 353–371, October 2006. ISSN 0021-9991.

Zhou, J. G. Lattice Boltzmann Methods for Shallow Water Flows. Berlin: Springer, 2004.

Zou, Q.; He, X. On pressure and velocity boundary conditions for the lattice boltzmann bkg model. Phys. Fluids, v. 9, n. 6, p. 1591–1598, 1997.