



Universidade de Aveiro Departamento de Engenharia Mecânica
2011

Tiago Gaspar Leal

**SIMULAÇÕES LBM DE ESCOAMENTOS EM
GEOMETRIAS COMPLEXAS**



Tiago Gaspar Leal

**SIMULAÇÕES LBM DE ESCOAMENTOS EM
GEOMETRIAS COMPLEXAS**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica realizada sob a orientação científica do Doutor António C. M. Sousa, Professor Catedrático do Departamento de Engenharia Mecânica da Universidade de Aveiro.

o júri

presidente

Prof. Doutor Fernando José Neto da Silva
Professor Auxiliar da Universidade de Aveiro

vogais

Prof. Doutor António Carlos Mendes de Sousa (Orientador)
Professor Catedrático da Universidade de Aveiro

Prof. Doutor António Manuel Gameiro Lopes
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

agradecimentos

Aproveito este espaço para agradecer ao Professor Doutor António C. M. Sousa pelo apoio e orientação prestados durante a elaboração da presente dissertação. Sinto-me muito contente pelo aprendizado que recebi nas pequenas reuniões realizadas semanalmente. Um obrigado ainda pela forma como sempre me recebeu e pelo encorajamento constantemente transmitido. Foi uma verdadeira honra.

Ao Doutor Aydin Nabovati pela ajuda prestada durante a compreensão do tema sobre o qual me debrucei e por ter compartilhado comigo a sua experiência acerca do assunto.

Agradeço também ao aluno de doutoramento Gonçalo Silva, do Instituto Superior Técnico de Lisboa, que conheci através de um fórum acerca do LBM e que gentilmente sempre me tentou ajudar no que necessitei.

A todos os professores do DEM pelo conhecimento transmitido e disponibilidade para ajudar.

Aos meus colegas e amigos mais próximos por toda a partilha de conhecimento e amizade.

À minha família e namorada por todo o apoio e carinho que sempre me deram. Pai, Mãe, Catarina, Avô e Elhanna o meu obrigado por sempre estarem ao meu lado e me motivarem constantemente.

palavras-chave

método *lattice* Boltzmann, geração de malha, geometrias complexas, simulação de escoamentos.

resumo

O método *Lattice Boltzmann* (LBM) permite simular escoamentos de fluidos dos mais diversos tipos em boa concordância com a realidade. Para se retirar as devidas conclusões do cálculo executado pelo algoritmo do LBM é necessária a intervenção de uma plataforma que ilustre visualmente os resultados. Nesse sentido o presente trabalho propõe procedimentos que permitam a integração de geometrias complexas num simulador LBM. Para o efeito, adaptou-se um código LBM escrito em MATLAB disponível em literatura aberta. Foram feitas simulações de escoamentos em diversas geometrias integradas, tendo sido levado em conta soluções capazes de melhor representar resultados obtidos. Adicionalmente foram feitos pequenos testes de forma a validar o algoritmo, assim como previsões de permeabilidade de algumas estruturas porosas com soluções analíticas conhecidas. Entre as várias aplicações de engenharia que o procedimento descrito poderá ter, deu-se ênfase ao projecto de moldes a serem utilizados na produção de compósitos poliméricos. Como objectivo final e já fora do contexto da presente dissertação, pretende-se, vir a adaptar e a validar o algoritmo em escoamentos de fluido numa matriz (*preform*) mediante o processo de Moldação por Transferência de Resina (RTM), contribuindo deste modo, para a verificação de quais as condições ideais para uma total impregnação de resina na matriz e por conseguinte assegurar a integridade estrutural da peça.

keywords

lattice Boltzmann Method, mesh generation, complex geometries, fluid flow simulation.

abstract

Lattice Boltzmann Method allows the simulation of a wide range of fluid flows, with results quite consistent with reality. To facilitate the interpretation of the results produced by the LBM simulation and eventually draw the appropriate conclusions is necessary the intervention of a platform that visually illustrates the results. In this regard, this project proposes a procedure which allows the integration of arbitrary geometries in a LBM simulator. For this purpose, an LBM code was adapted; results of simulations for relevant cases are presented in this dissertation. In addition, several tests were done to validate the algorithm used, including permeability prediction in some structures, namely ordered arrangements of spheres. Among many engineering applications, the procedure described may be of particular use in the design of moulds for the production of polymer composites. In fact, the final goal, although outside the context of this dissertation, is to validate the algorithm for fluid flow prediction in preforms used in the Resin Transfer Moulding (RTM) process, thus contributing to the understanding of the impregnation of preform by the resin eventually yielding improved structural integrity.

ERRATA

Leal, T. G. Simulações LBM de escoamentos em geometrias complexas. 2012. 103. Dissertação (Mestrado em Engenharia Mecânica) - Departamento de Engenharia Mecânica Universidade de Aveiro, Aveiro, 2011.

Página	Linha	Onde se lê	Deve ler-se
iii	8	Figura 1.8	Figura 1.7
iii	8	Figura 1.7	Figura 1.8
vii	1	NOMENCLAURA	NOMENCLATURA
11	3	que serviram	que serviu
13	19	de todas	de cada
12	14;16	Em que	em que
14	3	$\vec{x} + c_i$	$\vec{x} + c_i \Delta t$
14	5	$n_i(\vec{c} + c_i, t + 1)$	$n_i(\vec{c} + c_i \Delta t, t + 1)$
15	28	do quadrado que as envolve	dos quadrados vizinhos
26	4	temos na linha	tem-se na linha
37	1	domínios	domínios
45	20	adequada	adequado
60	5;9;17	Em que	em que
77	12	Cellular Automata Cellular Automata	Cellular Automata

ÍNDICE

ÍNDICE DE FIGURAS	iii
ÍNDICE DE TABELAS	vi
NOMENCLATURA	vii
1. INTRODUÇÃO	1
1.1. Motivação	1
1.2. Objectivos.....	2
1.3. Revisão Bibliográfica	3
1.3.1. Moldação por Transferência de Resina.....	3
1.3.2. Método Lattice Boltzmann.....	5
1.3.3. Geração de Malhas	6
1.4. Sumário da Tese	10
2. MÉTODO LATTICE-BOLTZMANN	11
2.1. Fundamentação Física e Matemática	11
2.2. Evolução e Caracterização do LBM	13
2.3. Condições Fronteira	20
2.3.1. Bounce-Back.....	20
2.3.2. Condições Fronteira Periódicas.....	22
3. CÓDIGO NUMÉRICO	24
4. TESTES E VALIDAÇÃO	31
4.1. Simulação Numérica do Escoamento de <i>Poiseuille</i>	31
4.2. Verificação da Continuidade Usando o LBM.....	35
5. TRATAMENTO DOS RESULTADOS	40
5.1. Integração de Novas Geometrias.....	40
5.2. Visualização dos Vectors de Velocidade	45
5.3. Visualização 3D	50
5.4. Escoamentos em Meios Porosos	58
5.4.1. Previsão da Permeabilidade em Arranjos de Esferas.....	59
6. CONCLUSÕES E TRABALHOS FUTUROS	75
REFERÊNCIAS.....	77
ANEXO I – Código de Validação - Escoamento de <i>Poiseuille</i>	84

ANEXO II – Código de Validação - Princípio da Continuidade	85
ANEXO III - Código D2Q9 Gerador da Simulação da “Geometria 1”	86
ANEXO IV - Código D3Q19 para “Cilindros Intersectados”	89
ANEXO V - Código D3Q19 para “Estrutura CFC de $L = 51$ e $a = 16$ ”	93
ANEXO VI - Código para Representação Gráfica da Permeabilidade	98
ANEXO VII - Código para Representação Gráfica da Permeabilidade Normalizada.....	102

ÍNDICE DE FIGURAS

Figura 1.3 - Esquema das várias etapas do RTM, adaptado de [36].	4
Figura 1.1 - Velocidades <i>lattice</i> discretizadas para o modelo D2Q9 [32].	6
Figura 1.2 - Velocidades <i>lattice</i> discretizadas para o modelo D3Q19 [33].	6
Figura 1.4 - Exemplos de malhas regulares [40].	7
Figura 1.5 - Exemplo de malha não regular [40].	7
Figura 1.6 - Esquematização da discretização e geração de malha a partir de objectos reais, adaptado de [41].	8
Figura 1.8 - Linhas de fluxo de velocidade em torno de um veículo [42].	9
Figura 1.7 - Esquematização do processo de geração de malha a partir de dados CAD, adaptado de [41].	9
Figura 1.9 - Representação esquemática do processo de geração de malhas a partir de geometrias sintéticas, adaptado de [41].	9
Figura 2.1 - Rede <i>lattice</i> hexagonal FHP [46].	13
Figura 2.2 - Todas colisões FHP presentes no <i>lattice</i> hexagonal [11].	14
Figura 2.3 - Representação de nós <i>lattice</i> D2Q9 e D3Q19 [57].	16
Figura 2.4 - Função de distribuição de partículas de uma célula antes e depois da etapa de movimento, adaptado de [58].	17
Figura 2.5 - Função de distribuição de partículas de quatro células vizinhas antes e depois da etapa de movimento, adaptado de [58].	17
Figura 2.6 - Esquematização do processo dedutivo do LBM, adaptado de [38].	19
Figura 2.7 - Descrição esquemática do <i>bounce-back</i> , adaptado de [63].	21
Figura 2.8 - Descrição esquemática do <i>halfway bounce-back</i> (à esquerda) e do <i>bounce-forward</i> (à direita) [70].	21
Figura 2.9 - Ilustração esquemática da acção do <i>halfway-bounce-back</i> e do <i>bounce-forward</i> , adaptado de [70].	22
Figura 2.10 - Condições fronteira periódicas, em que se nota a interacção entre nós de saída e entrada [11].	23
Figura 3.1 - Resultado do escoamento de fluido em meio poroso.	29
Figura 3.2 - Resultado do escoamento de fluido obtido na secção $z = 5$.	30
Figura 3.3 - Resultado do escoamento de fluido obtido na secção $y = 5$.	30
Figura 4.1 - Representação à saída de escoamento bidimensional entre duas paredes.	31
Figura 4.2 - Solução numérica e solução analítica do perfil de velocidades do escoamento de <i>Poiseuille</i> para simulação de 21 nós ao longo da altura da conduta.	32
Figura 4.3 - Solução numérica e solução analítica do perfil de velocidades do escoamento de <i>Poiseuille</i> para simulação de 41 nós ao longo da altura da conduta.	32
Figura 4.4 - Solução numérica e solução analítica do perfil de velocidades do escoamento de <i>Poiseuille</i> para simulação de 61 nós ao longo da altura da conduta.	33
Figura 4.5 - Solução numérica e solução analítica do perfil de velocidades do escoamento de <i>Poiseuille</i> para simulação de 81 nós ao longo da altura da conduta.	33

Figura 4.6 - Solução numérica e solução analítica do perfil de velocidades do escoamento de <i>Poiseuille</i> para simulação de 101 nós ao longo da altura da conduta.	33
Figura 4.7 - Erro relativo máximo das simulações realizadas em função do número de nós presentes na altura da conduta.	34
Figura 4.8 - Erro relativo médio das simulações realizadas em função do número de nós presentes na altura da conduta.	35
Figura 4.9 - Representação esquemática de uma coluna de vectores de velocidade das simulações.	36
Figura 4.10 - Canal com contracção e expansão de 100x20 nós.	36
Figura 4.11 - Canal com contracção e expansão de 100x30 nós.	37
Figura 4.12 - Canal com contracção e expansão de 100x40 nós.	37
Figura 4.13 - Canal com contracção e expansão de 100x50 nós.	37
Figura 4.14 – Erro relativo máximo entre diferenças de caudais em domínios com diferentes números de nós ao longo da secção transversal ao escoamento.	39
Figura 5.1 - Simulação a um escoamento em torno da “geometria 1” com resolução de 20x20 pixéis/nós.	41
Figura 5.2 - Simulação a um escoamento em torno da “geometria 1” com resolução de 60x60 pixéis/nós.	42
Figura 5.3 - Aproximação a pormenor da simulação de escoamento em torno da “geometria 1” com resolução de 60x60 pixéis/nós.	43
Figura 5.4 - Simulação a um escoamento em torno da “geometria 2” com resolução de 63x52 pixéis/nós.	44
Figura 5.5 - Aproximação a pormenor da simulação de escoamento em torno da “geometria 2” com resolução de 63x52 pixéis/nós.	44
Figura 5.6 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 20x20 pixéis/nós.	46
Figura 5.7 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 40x40 pixéis/nós.	47
Figura 5.8 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 60x60 pixéis/nós.	47
Figura 5.9 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 120x120 pixéis/nós.	48
Figura 5.10 - Simulação do escoamento por entre a “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 200x200 pixéis/nós.	48
Figura 5.11 - Simulação do escoamento através da “geometria 3”, com diferente definição nos vectores de velocidade (120x120 pixéis/nós).	49
Figura 5.12 - Simulação do escoamento através da “geometria 3”, com diferente definição nos vectores de velocidade (200x200 pixéis/nós).	50
Figura 5.13 - Simulação de escoamento tridimensional em torno de uma esfera, com definição de 12x12x12 nós.	51

Figura 5.14 - Simulação de escoamento tridimensional em torno de uma esfera, com definição de 12x12x12 nós; cortes em $y = 9$ e $z = 9$	52
Figura 5.15 - Escoamento tridimensional através de uma geometria de composta por 5 objectos esféricos, com um domínio de 12x12x12 nós.....	52
Figura 5.16 - Simulação do escoamento tridimensional através de uma geometria composta por 5 objectos esféricos; cortes em $z = 9$ e $y = 6$	53
Figura 5.17 - Imagem ilustrativa do procedimento de geração de geometrias 3D.	53
Figura 5.18 - Geometria 3D gerada mediante o procedimento de prolongação na terceira dimensão de imagens 2D.....	54
Figura 5.19 - Escoamento em torno da Geometria 3D gerada mediante expansão na terceira dimensão da "Imagem 1".....	55
Figura 5.20 - Geometria obtida mediante expansão na terceira dimensão da "Imagem 1"; cortes em $z = 6$ e $y = 6$	55
Figura 5.21- Escoamento tridimensional através de uma geometria porosa, com definição de 14x14x14 nós.	56
Figura 5.22 - Escoamento tridimensional através de uma geometria porosa, com definição de 14x14x14 nós; cortes em $y = 7$ e $x = 3$	56
Figura 5.23 - Escoamento tridimensional através de uma geometria de objectos cilíndricos cruzados, com uma definição de 18x18x18 nós.	57
Figura 5.24 - Simulação de escoamento tridimensional através de objectos cilíndricos cruzados, com definição de 18x18x18 nós; cortes em $z = 9$ e $y = 6$	57
Figura 5.25 - Geometria cúbica simples de raio $a = 12$, num domínio de $L = 51$, com porosidade $\phi = 0,95$	61
Figura 5.26 - Geometria cúbica de corpo centrado de raio $a = 10$, num de domínio de $L = 51$, com porosidade $\phi = 0,94$	62
Figura 5.27 - Geometria cúbica de faces centradas de raio $a = 10$, num domínio de $L = 51$, com porosidade $\phi = 0,87$	62
Figura 5.28 - Simulação numérica de escoamento através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\phi = 0,74$).	64
Figura 5.29 - Simulação numérica de escoamento através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\phi = 0,48$).	65
Figura 5.30 - Simulação de escoamento tridimensional através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\phi = 0,74$); corte em $y = 1$	65
Figura 5.31 - Simulação de escoamento tridimensional através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\phi = 0,74$); corte em $y = 26$	66
Figura 5.32 - Simulação de escoamento tridimensional através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\phi = 0,48$); corte em $y = 1$	66
Figura 5.33 - Simulação de escoamento tridimensional através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\phi = 0,48$); corte em $y = 26$	67
Figura 5.34 - Geometria cúbica da faces centradas de raio $a = 40$, num domínio $L = 291$, com porosidade $\phi = 0,96$	68

Figura 5.35 - Geometria CFC de $L = 291$, $a = 80$ (porosidade $\phi = 0,65$); corte em $y = 1$	68
Figura 5.36 - Geometria CFC de $L = 291$, $a = 80$ (porosidade $\phi = 0,65$); corte em $y = 146$	69
Figura 5.37 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos, em oposição à correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 51$	70
Figura 5.38 - Permeabilidade prevista normalizada pelo raio das esferas em função dos diferentes arranjos de esferas: CS, CC e CFC.....	71
Figura 5.39 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos com a correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 31$	73
Figura 5.40 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos, com a correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 81$	73
Figura 5.41 - Permeabilidade prevista normalizada pelo raio das esferas em função dos diferentes arranjos de esferas: CS, CC e CFC para domínios de $L = 31$, $L = 51$ e $L = 81$	74

ÍNDICE DE TABELAS

Tabela 4.1 - Caudais mássicos presentes no domínio ao longo das condutas.....	38
Tabela 5.1 - Correspondência entre valores numéricos de RGB num mapa de cores.....	45
Tabela 5.2 - Raios de esferas e respectivas porosidades utilizadas nas simulações de $L = 51$	61
Tabela 5.3 - Conversão de unidades <i>lattice</i> para unidades físicas [11].	63
Tabela 5.4 - Raios de esferas com respectivas porosidades utilizados nas simulações de $L = 31$	72
Tabela 5.5 - Raios de esferas com respectivas porosidades utilizados nas simulações de $L = 81$	72

NOMENCLATURA

a	raio de esferas pertencentes a dada estrutura cúbica
A	área
c	fracção volúmica sólida
c_i	conjunto de velocidades discretas
c_s	velocidade do som
C_d	coeficiente de arrasto
$c_{máx}$	fracção volúmica sólida máxima de dada estrutura cúbica
D	dimensão espacial
f	função de distribuição
f_i	conjunto de funções de distribuição
f^e	função de distribuição de equilíbrio
f_i^{eq}	conjunto de funções de distribuição de equilíbrio
f^M	distribuição de velocidades de Maxwell-Boltzmann
K	força externa
K	permeabilidade
L	comprimento de aresta de domínio cúbico
m	massa de partícula
\dot{m}	caudal mássico
n_i	conjunto de posições binárias do LGCA
\vec{p}	vector de quantidade de movimento
P	pressão
Q	número de velocidades de um modelo LBM
Q	operador de colisão da equação de Boltzmann
R	constante de Boltzmann
R	raio
t	instante de tempo

T	temperatura
u	componente de velocidade
\vec{u}	vector velocidade
$\langle u \rangle$	velocidade média
$V_{sólido}$	volume de sólido
V_{total}	volume total
\vec{x}	vector de posição
w_i	factor de ponderação
Δx	distância entre nós (unidade <i>lattice</i>)
Δt	intervalo de tempo
∇	gradiente
∇P	gradiente de pressão
v	velocidade de uma molécula
τ	tempo de relaxação
Λ	comprimento médio livre entre partículas
ρ	massa volúmica
φ	porosidade
ϑ	viscosidade cinemática do fluido
v	velocidade de dada partícula antes de colidir
v_1	velocidade da "partícula 1" antes de colidir
v'	Velocidade de dada partícula após colidir
v'_1	velocidade da "partícula 1" após colidir
$\sigma(\Omega)$	secção de cruzamento numa colisão de partículas
$\vec{\xi}$	velocidade macroscópica do escoamento
A	dimensão no espaço de determinada rede <i>lattice</i>
α_n	coeficientes de Sangani e Acrivos
B	número de velocidades discretas de determinada rede <i>lattice</i>

β	número de esferas presentes numa estrutura cúbica
μ	viscosidade dinâmica
Φ	variável que relaciona a porosidade com a permeabilidade normalizada
χ^n	função da fracção volúmica sólida de dada estrutura cúbica
∂_t	derivada em ordem ao tempo
∂_x	derivado em ordem à posição
∂_v	derivada em ordem à velocidade
a,b,c	pontos cartesianos
x,y,z	coordenadas cartesianas

1. INTRODUÇÃO

1.1. Motivação

A Dinâmica dos Fluidos é uma temática que tem sido alvo de grande interesse científico desde o tempo de Newton [1], tendo tido avanços significativos durante o século XIX com a formulação das equações de Navier-Stokes [2]. Resumidamente, o grande objectivo desta área do conhecimento é conseguir compreender e representar os mais diferentes tipos de escoamento de fluidos.

Trata-se de uma área de grande relevância em processos industriais e naturais. Nesse sentido, muito trabalho de investigação tem vindo a ser feito ao longo do tempo para se desenvolverem métodos numéricos que permitam melhores reproduções matemáticas do comportamento de escoamentos de fluidos. A par destes, também metodologias experimentais têm sido alvo de desenvolvimentos a fim de através de visualização e medição se poder melhor compreender os fenómenos inerentes aos escoamentos de fluidos. Estes esforços acompanhados pelo progresso computacional registado a partir de meados do século passado culminaram no aparecimento de uma nova área de investigação científica: *Computational Fluid Dynamics* (CFD), em português Dinâmica dos Fluidos Computacional. Hoje em dia é inegável se dizer que este ramo científico desempenha papel preponderante na resolução de problemas de várias áreas da engenharia, por exemplo: Mecânica, Civil, Aeroespacial e Química. Relativamente aos métodos experimentais, o CFD oferece as seguintes vantagens [3]:

- Tempo de projecto e desenvolvimento é significativamente inferior;
- CFD tem a capacidade de simular escoamentos não reproduzíveis em modelos experimentais;
- CFD disponibiliza informação de forma mais detalhada e organizada;
- CFD tem custos muito inferiores aos dos testes experimentais;
- CFD consome menos energia;

Falando especificamente de escoamentos em meios porosos [4, 5], estes também se encontram em várias áreas da ciência e engenharia como é o caso da mecânica, das geociências, ciência dos materiais, biologia e biofísica. Este particular tipo de escoamentos, por ser um assunto de interesse comum, acabou por se tornar um campo de estudos autónomo.

Modelos tradicionais assentes nas equações de Navier-Stokes (ex: elementos finitos, diferenças finitas ou volumes finitos) aplicados à escala microscópica e mesoscópica

(escalas em que se resolvem estruturas porosas) apresentam limitações de ordem numérica visto se basearem em pressupostos de conservação macroscópica (considera-se o fluido como um meio contínuo e as variáveis de interesse como sendo todas diferenciáveis).

Na tentativa de ultrapassar as dificuldades enunciadas foram desenvolvidos métodos com fundamentação teórica diferente, orientados a partir de trabalhos efectuados por físicos que se dedicaram ao desenvolvimento de metodologias de base estatística que permitissem uma melhor compreensão de fenómenos dinâmicos em estados de não equilíbrio. Foi assim que o *Lattice-Gas Cellular Automata* e mais tarde o método *lattice-Boltzmann* (LBM) [6] surgiram.

Devido à sua abordagem estatística e de cariz microscópico/mesoscópico o LBM encontra-se muito bem estabelecido em trabalhos envolvendo escoamentos em meios porosos.

Um exemplo típico de escoamento em meio poroso acontece na produção de compósitos, nomeadamente no processo Moldação por Transferência de Resina (RTM), já que este processo incorpora uma etapa de injeção de resina a elevada pressão para o interior de uma matriz constituída por fibras [7]. De referir que o RTM será descrito em maior detalhe posteriormente.

Os maiores problemas associados ao RTM estão relacionados com falhas de preenchimento da matriz pela resina, originando poros (bolsas de ar); e com a ocorrência de vazamento de resina por zonas indesejáveis na matriz (entradas de ar e eventuais espaços entre o molde e a preforma por exemplo) [7-9]. Nesse sentido torna-se útil o conhecimento dos parâmetros inerentes à injeção da resina e se saber qual o comportamento da resina aquando da sua injeção na matriz, entrando aqui a utilidade do LBM nesta problemática [8,9].

Sabendo que a configuração das fibras presentes na matriz é de cariz complexo, torna-se necessário procurar meios capazes de gerar geometrias para acomodar estas configurações e desta forma servir de plataforma ao cálculo do algoritmo LBM.

1.2. Objectivos

A presente dissertação tem como principal objectivo o desenvolvimento de um sistema que permita gerar/mapear uma malha cartesiana a ser utilizada com um simulador LBM a partir de geometrias arbitrárias por Desenho Assistido por Computador - CAD.

Pretende-se assim, ter a possibilidade de gerar ou importar de forma rápida e intuitiva geometrias complexas para serem integradas num simulador LBM, para deste modo, se poder obter simulações de escoamentos numericamente definidos e condicionados pelas ditas geometrias.

Será ainda objecto de estudo a execução de algumas simulações de forma a testar o método e a comparar resultados numéricos com soluções analíticas presentes na literatura.

No futuro, e não fazendo parte do presente projecto, pretende-se ter a capacidade de obter as malhas directamente dos resultados experimentais obtidos via NMR/MRI – (ressonância magnética nuclear/imagens por ressonância magnética). Neste sistema haverá uma segmentação binária, à semelhança daquilo que é pretendido no presente projecto, que atribua a cada “voxel” (elemento de volume) a designação de “fluido” ou “sólido”. De mencionar que já existem códigos comerciais que fazem a conversão das imagens experimentais em elementos de volume (*A Software Package for Automated Analysis of Rock Pore Structure in 3-D Computed Microtomography Images*) a teoria do princípio pode ser consultada em:

http://www.ams.sunysb.edu/~lindquis/3dma/3dma_rock/3dma_rock.html#Sec_Pore. Deve ser dito que a maioria destes códigos têm várias limitações, pois funcionam como “caixas pretas”, e os seus custos de utilização/aquisição são relativamente elevados.

Um outro objectivo do presente projecto é a utilização dos ditos elementos de volume (*voxels*) em simulações tridimensionais de escoamento de fluidos utilizando o LBM.

Esta capacidade tem aplicação imediata no projecto de moldes a serem utilizados na produção de compósitos poliméricos, nomeadamente no processo RTM, já que a análise do escoamento da resina numa matriz idêntica à realidade, permite um projecto melhorado do molde de modo a assegurar uma total impregnação da resina na matriz, garantindo desta forma a integridade estrutural da peça a ser moldada.

1.3. Revisão Bibliográfica

1.3.1. Moldação por Transferência de Resina

A moldação por transferência de resina (RTM – “resin transfer moulding”) é um processo utilizado no processamento de materiais compósitos de matriz polimérica reforçados por fibras para aplicações de elevada performance nas indústrias aeroespacial, automóvel, biomédica e desportiva [34].

Trata-se de um processo não muito recente, tendo começado a ser usado na indústria no início da década de 1940. Contudo a sua utilização foi muito limitada até à década de 1970 devido, principalmente, à inadequação das resinas e equipamento existentes. A partir dos anos 80 desenvolveram-se matrizes fibrosas e reduziu-se a viscosidade das resinas, o que permitiu o fabrico de produtos de geometrias mais complexas para aplicações mais diversificadas. Combinando os desenvolvimentos enunciados com um menor capital a investir e menores libertações de compostos voláteis, a popularidade deste método aumentou fortemente [35].

O processo caracteriza-se pela injeção de resina sob pressão para o interior de uma matriz constituída por fibras. O RTM inicia-se pela colocação do reforço de fibras devidamente orientadas, depois disso o molde é fechado e a resina é injectada até impregnar a totalidade da matriz. Segue-se logo de seguida o processo de cura que é

auxiliado por ventilação externa e saídas de ar. Por fim, na última etapa a peça é retirada da matriz. O processo encontra-se esquematizado na Figura 1.3 [36].

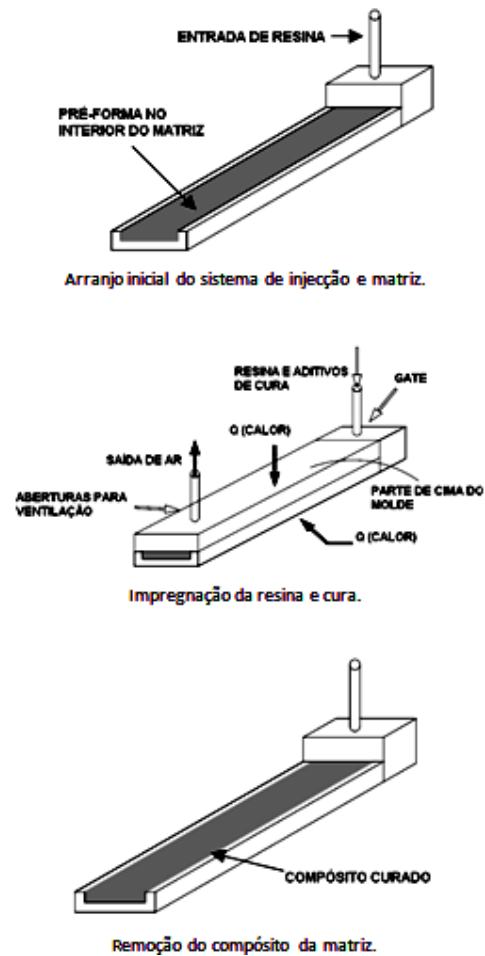


Figura 1.1 - Esquema das várias etapas do RTM, adaptado de [36].

A produção de compósitos mediante o RTM, possibilita a automatização do processo e conseqüente produção em série. Tem também a capacidade de produzir peças de grandes dimensões e complexas, sem que haja necessidade de adição de partes menores. Além disso, o tempo de ciclo produtivo é pequeno comparativamente com outros processos, garantindo bons acabamentos superficiais e boa tolerância dimensional [37].

Por outro lado, o RTM apresenta algumas desvantagens. Desde logo a dificuldade no projecto da matriz, isso acontece devido à incerteza existente na determinação das entradas de resina e saídas de ar adequadas para a prevenção de falhas de preenchimento. Outra desvantagem prende-se com o facto de nalguns casos existir a possibilidade de deslocamento da pré-forma durante a injeção [37].

1.3.2. Método Lattice Boltzmann

Método *Lattice Boltzmann* (LBM), embora ainda se encontre pouco referenciado em Portugal, a sua difusão na comunidade científica internacional é já relevante. O crescente interesse no método levou alguns autores a publicarem textos de referência que caracterizam e fundamentam o método [6, 10, 11].

O LBM tem a capacidade de simular escoamentos de fluidos em diferentes condições físicas, conseguindo retratar o comportamento de escoamentos acompanhados de mudança de fase (evaporação, condensação) [12 - 15], cavitação [16 - 18], substâncias dissolvidas [19 - 21], fenómenos de transferência de calor [22 - 25], meios porosos [8, 9, 26 - 28] e interações com superfícies [29].

Durante os últimos 20 anos tem-se vindo a assistir a um aumento exponencial de artigos científicos que utilizam o LBM como palavra-chave, sendo a física, a engenharia química e a engenharia mecânica as principais áreas do conhecimento que se têm servido desta ferramenta [10].

O trabalho do físico austríaco Ludwig Eduard Boltzmann [30] acerca da teoria cinética dos gases [31] inspirou o modelo *Lattice-Gas Cellular Automata* (LGCA) [6] que tem a capacidade de simular gases como “pacotes” de partículas em pontos discretos no espaço, representados por variáveis booleanas. O LBM tem origem no LGCA, tendo ambos características de base comuns: ambos possuem um conjunto de nós ligados entre si; algumas variáveis de estado são definidas em cada nó; e ambos os modelos fazem uso de uma regra evolutiva, indicando que o comportamento de uma partícula tem a influência das vizinhas (nomeadamente em termos de colisões e movimento) [32].

A rede *lattice*, como é tecnicamente designada, discretiza e restringe as velocidades, de modo a que as partículas possam apenas se deslocar num número finito de direcções e com um número limitado de velocidades. Em cada espaço de tempo, as partículas movem-se ao longo das suas correspondentes direcções chegando ao nó vizinho. Se mais do que uma partícula chegar simultaneamente ao mesmo nó, a regra de colisão é aplicada, redistribuindo as partículas atendendo às leis de conservação de massa e momento [10].

A grande diferença que o LBM apresenta relativamente ao LGCA reside na utilização de funções de probabilidade $f_i(\vec{x})$ (função de distribuição de partículas), substituindo a metodologia morosa de interpretação do conjunto de variáveis booleanas calculadas (será visto em maior detalhe posteriormente). Além da diferença referida, o LBM possui um operador de colisão mais simples, permitindo trabalhar-se em mais dimensões e com mais direcções de velocidade [32].

A Figura 1.1 e a Figura 1.2 representam exemplos de rede *lattice*, no caso, um modelo bidimensional com 9 velocidades e outro tridimensional com 19 velocidades, respectivamente. Estes modelos serão abordados em maior detalhe posteriormente.

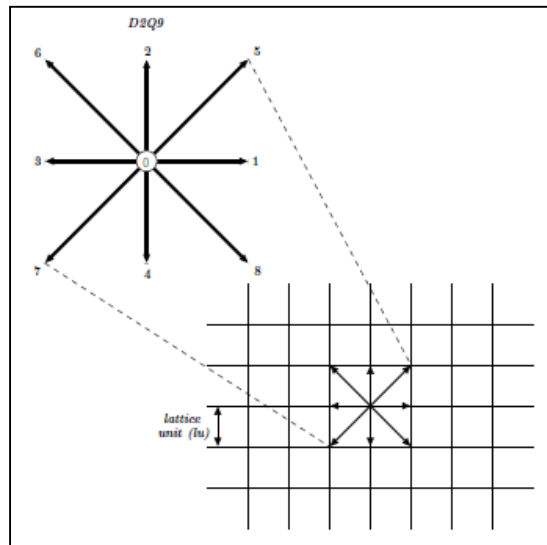


Figura 1.2 - Velocidades *lattice* discretizadas para o modelo D2Q9 [32].

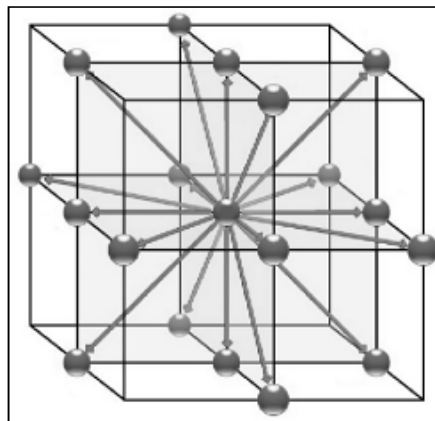


Figura 1.3 - Velocidades *lattice* discretizadas para o modelo D3Q19 [33].

1.3.3. Geração de Malhas

Aquando da implementação de um modelo LBM em geometrias complexas é de grande utilidade se conseguir obter uma malha da geometria a simular. A malha em causa discretiza a geometria, servindo de plataforma aos cálculos que vão sendo efectuados pelo algoritmo programado.

A geração de malhas é um processo importante nos códigos numéricos de LBM, pois é responsável pelo nível de precisão dos resultados da simulação [38]. Assim, torna-se necessário ter em conta determinados factores, como a necessidade da discretização e da correspondente resolução da malha numérica estarem proporcionais ao estado físico do sistema. É sabido ainda que o código é sempre mais robusto e eficiente quando usa uma malha bem construída [38].

As formas de gerar malhas tendem a ser códigos complexos e pouco acessíveis. Por norma são usados como “caixas negras”, sendo utilizados de pronto sem se saber ao certo o que está por detrás da criação das malhas. Isto acontece porque o *software* gerador de malhas, é em geral de difícil integração com outros códigos [39].

As malhas podem ser de dois tipos regulares ou não regulares. A principal característica das malhas regulares prende-se com o facto de todos os nós da malha terem um número igual de células ou elementos adjacentes. Em contraste, uma malha não regular possui ligações não padronizadas entre os diferentes elementos, permitindo uma maior flexibilidade [40]. As Figuras 1.4 e 1.5 ilustram os dois tipos de malha mencionados.

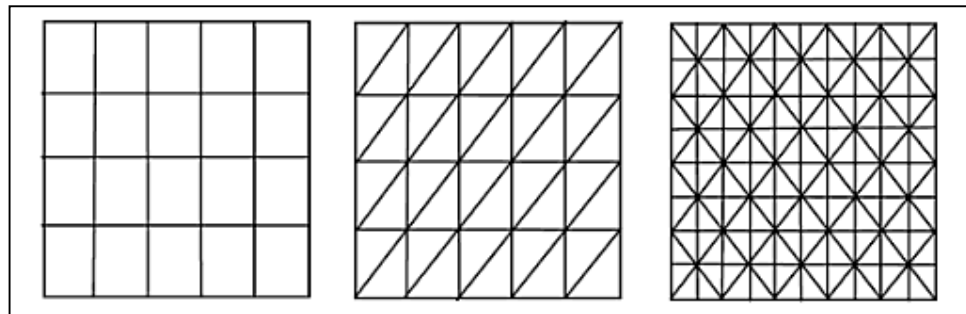


Figura 1.4 - Exemplos de malhas regulares [40].

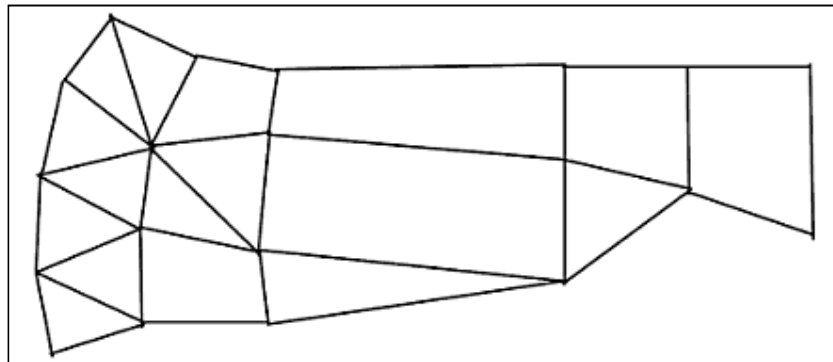


Figura 1.5 - Exemplo de malha não regular [40].

Devido à relativa simplicidade em integrar geometrias complexas, sem que estas tenham grande impacto no desempenho do algoritmo, o LBM ganhou preferência na área de escoamentos em geometrias complexas, em que os métodos tradicionais baseados nas equações de Navier-Stokes apresentam várias complicações na geração de malhas [37]. Desde cedo se tornou óbvio que a abordagem assente em pontos discretos celulares permitiria uma rápida e semi-automática geração de malhas a partir de imagens digitais, dados CAD e outras geometrias criadas [41].

Em grande parte dos casos, são usadas matrizes binárias para mapear as malhas numéricas directamente a partir de “Tomografia Computarizada” e Raios X. Estes tipos de malhas geralmente têm a mesma resolução da matriz de elementos de volume (*voxels*) original e, conseqüentemente, o código consegue calcular todo o tipo de escoamento com

alguma precisão e eficiência. Além disso, pelo menos caso as matrizes sejam compostas por valores brancos e pretos, a representação geométrica é reduzida à primeira ordem o que leva a uma certa perda na precisão caso a imagem inicial seja composta por valores cinzentos. No entanto, para a maior parte dos casos, resoluções elevadas não são necessariamente importantes [38].

Seguidamente serão apresentadas três estratégias de discretização de geometrias que têm vindo a ter sucesso em trabalhos relacionados com geometrias complexas para escoamentos LBM.

1.3.3.1. Imagens Digitais

No que diz respeito a investigação de escoamentos em meios porosos, o desenvolvimento de novos dispositivos em áreas como a engenharia química ou biomédica, exigem uma concepção da geometria do caso de estudo o mais realista possível, no sentido de se conseguir um correcto conhecimento das propriedades do escoamento em análise. Usando Tomografia Computorizada 3D (CT), geometrias complexas arbitrárias podem ser digitalizadas, podendo os dados deste processo (dados CT) posteriormente convertidos em dados de elementos de volume *Lattice*-Boltzmann. As etapas do processo mencionado encontram-se esquematizadas na Figura 1.6 [41].

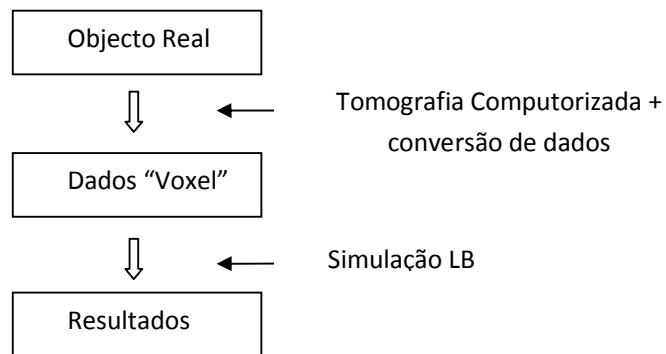


Figura 1.6 - Esquematização da discretização e geração de malha a partir de objectos reais, adaptado de [41].

1.3.3.2. Dados CAD

Em grande parte dos projectos na indústria automóvel, aeronáutica entre outras, modelos completos de veículos e produtos finais estão disponíveis em ficheiros de Desenho Assistido por Computador (CAD). Ora, utilizando software apropriado é possível fazer a discretização quase automática da geometria para modelos que já integram a malha de “voxels” para simulações LBM. A Figura 1.7 apresenta a esquematização do processo de discretização enunciado e na Figura 1.8 encontra-se ilustrado um exemplo de simulação de escoamento mediante uma geração de malhas automática a partir de um modelo em CAD [41].

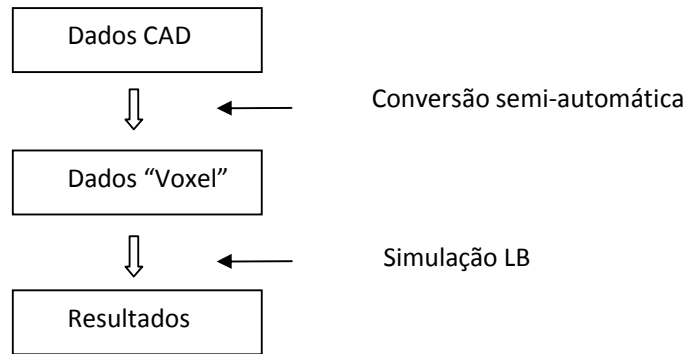


Figura 1.7 - Esquematização do processo de geração de malha a partir de dados CAD, adaptado de [41].

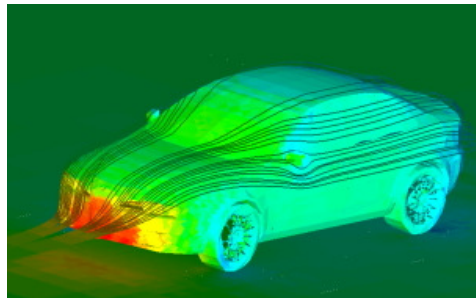


Figura 1.8 - Linhas de fluxo de velocidade em torno de um veículo [42].

1.3.3.3. Geometrias Sintéticas

Para casos em que a aplicação dos procedimentos anteriores seja inviável ou esteja indisponível, é comum se criarem geometrias artificiais de forma arbitrária, distribuindo-se em determinado domínio estruturas sólidas para criar determinada geometria ou meio poroso. Na Figura 1.9 encontra-se representada a descrição esquemática do processo de geração de malhas mediante o processo mencionado [41].

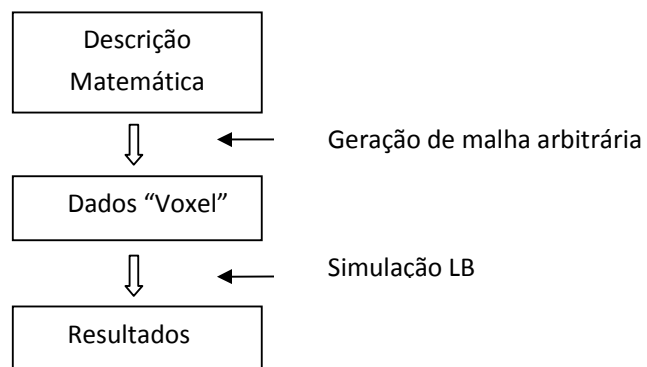


Figura 1.9 - Representação esquemática do processo de geração de malhas a partir de geometrias sintéticas, adaptado de [41].

1.4. Sumário da Tese

A presente dissertação apresenta no Capítulo 2 o método *lattice* Boltzmann (LBM), abordando a sua base teórica e demonstrando os seus fundamentos matemáticos, neste capítulo também são abordados diferentes tipos de condições fronteira que podem ser utilizados em códigos LBM.

O escopo deste trabalho é focado na temática de geração de malhas e criação de geometrias complexas, para o efeito procurou-se códigos LBM existentes que pudessem servir de ferramenta base ao estudo a executar. Assim, no Capítulo 3 é descrito o código utilizado, analisando de forma cuidada a sua metodologia de cálculo e modo de operação.

O Capítulo 4 tem como objectivo testar e validar os resultados. Um primeiro teste foi feito de modo a reproduzir o escoamento de *Poiseuille* para domínios com diferente número de nós ao longo da altura da respectiva conduta; tendo sido feito um segundo teste, ao princípio continuidade, tendo sido verificada a conservação de massa ao longo de vários domínios.

No Capítulo 5, é feita a apresentação do tratamento de dados. Para atingir os objectivos propostos procedeu-se a um conjunto de etapas que visam melhorar os resultados obtidos com o código utilizado. Assim, este Capítulo apresenta as alterações e incrementos feitos no código de modo a: permitir a integração de diferentes geometrias no simulador e melhorar a visualização dos resultados obtidos. Por fim são apresentados alguns testes de permeabilidade em estruturas cúbicas ocupadas por arranjos de esferas ordenadas.

No Capítulo 6 são expostas as principais conclusões resultantes do trabalho efectuado.

2. MÉTODO LATTICE-BOLTZMANN

Como mencionado na secção 1.3.1, o LBM é baseado na equação de Boltzmann formulada em 1872 pelo físico austríaco Ludwig Boltzmann. No presente capítulo far-se-á uma breve apresentação da fundamentação física e matemática que serviram de base ao LBM [6, 10, 11].

O LBM será depois abordado sob o ponto de vista de evolução do *Lattice Gas Cellular Automata* (LGCA). Serão apresentadas as características base comuns dos dois métodos e feita uma descrição do LBM e das suas particularidades [6, 10, 11].

O presente Capítulo termina com uma breve descrição dos principais tipos de condição fronteira que são implementados no modelo LBM tratado no presente trabalho [6, 10, 11, 32].

2.1. Fundamentação Física e Matemática

A teoria cinética dos gases tenta descrever o comportamento das propriedades macroscópicas dos gases, como por exemplo pressão e temperatura, através de uma abordagem estatística à interacção entre as moléculas que os constituem [31].

Neste âmbito, a partir de meados do século XIX, o físico escocês James Clerk Maxwell, publicou alguns artigos de investigação que tinham uma abordagem física completamente diferente da que se utilizava na altura, impulsionando a chamada mecânica estatística e criando uma nova forma de abordar fenómenos termodinâmicos.

Poucos anos depois, Ludwig Boltzmann começou a publicar trabalhos relacionados com os de Maxwell, só que com um ponto de partida completamente diferente: enquanto Maxwell baseou o seu trabalho e as suas deduções matemáticas em fenómenos empíricos, Boltzmann quis criar hipóteses teóricas para deduzir comportamentos físicos [31]. Foi no contexto descrito que em 1872 Ludwig Boltzmann apresentou uma equação estatística, mais tarde denominada de equação de Boltzmann (eq. 2.1), que pretendia dar a probabilidade de se obter a localização de uma molécula com determinada quantidade de movimento num determinado instante de tempo. Essa equação traduz-se numa distribuição de funções $f(\vec{x}, \vec{p}, t)$ e pode ser escrita como [6]:

$$\partial_t f + v \partial_x f + \frac{K}{m} \partial_v f = Q(f, f). \quad (2.1)$$

A equação de Boltzmann descreve o comportamento de duas partículas no momento em que colidem, indicando as direcções e velocidades antes e depois da colisão. O membro esquerdo da equação é conhecido como o operador de transporte, e descreve o movimento global das moléculas com velocidade v , sujeitas a um campo de forças externo K na posição x , enquanto o lado direito da equação é designado por operador de colisão, sendo dado por [6],

$$Q(f, f) = \int dv_1 \int d\Omega \sigma(\Omega) |v - v_1| [f(v')f(v'_1) - f(v)f(v_1)] \quad (2.2)$$

sendo $\sigma(\Omega)$ a secção de cruzamento da colisão diferencial de duas moléculas que transforma as velocidades de entrada $\{\mathbf{v}, \mathbf{v}_1\}$ em velocidades de saída $\{\mathbf{v}', \mathbf{v}'_1\}$.

Devido à sua estrutura complicada, o operador de colisão é normalmente substituído por expressões mais simples que não retiram o sentido físico do cálculo, já que conservam o número de partículas, quantidade de movimento e energia [43]. Esse tipo de procedimento é usualmente feito a partir da aproximação de Bhatnagar-Gross-Krook (BGK) [43] cujo operador de colisão que substitui (2.2) é dado por [6]:

$$C_{BGK}(f) = -\frac{f - f^e}{\tau} \quad (2.3)$$

em que f^e representa o equilíbrio local resultante da distribuição de Maxwell (também conhecida por distribuição de Maxwell-Boltzmann), sendo dada por [6]:

$$f^e = \rho \left(\frac{m^2}{2\pi RT} \right)^{3/2} e^{-\frac{(\xi - \bar{u})^2 m^2}{2RT}} \quad (2.4)$$

em que ξ é a velocidade macroscópica do escoamento, ρ é a densidade, T é a temperatura, m a massa duma partícula e R a constante de Boltzmann. Voltando ao operador de colisão BGK, τ representa o tempo de relaxação entre as colisões. O tempo de relaxação em termos gerais, é o tempo que um sistema demora a retomar o equilíbrio após um distúrbio.

Com a dita aproximação BGK e desprezando as forças externas, a equação de Boltzmann (2.1) toma a seguinte forma:

$$\frac{\partial f}{\partial t} + v \nabla f = -\frac{1}{\tau} (f - f^e). \quad (2.4)$$

2.2. Evolução e Caracterização do LBM

O método *lattice* Boltzmann pode ser explicado a partir da teoria cinética dos gases, acima mencionada. Contudo, na maior parte dos casos é mais conveniente discutir o método numa outra perspectiva: como sendo uma evolução do *Lattice Gas Cellular Automata* (LGCA) com quem partilha características base comuns.

Os modelos *Lattice-Gas* são metodologias que simplificam e simulam a interacção entre partículas pertencentes a um dado volume de controlo de determinado gás, tomando como entidades discretas parâmetros como velocidade, tempo e espaço de determinado movimento de partículas.

O primeiro modelo LGCA foi proposto em 1973 por Hardy, de Pazzis e Pomeau [44], tendo ficado conhecido por HPP (iniciais dos seus três autores). A importância do HPP é principalmente histórica, já que no limite macroscópico não converge com as equações de Navier-Stokes.

Uma década depois, em 1986, Frisch, Hasslacher e Pomeau [45] apresentaram um modelo assente numa rede *lattice* hexagonal simétrica (Figura 2.1) que demonstrava maior concordância com modelos tradicionais da dinâmica dos fluidos. Este modelo ficou conhecido por FHP, sigla das iniciais dos seus autores.

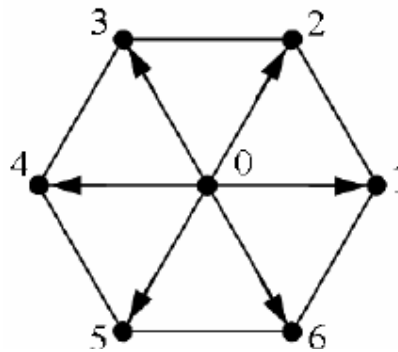


Figura 2.1 - Rede *lattice* hexagonal FHP [46].

De forma resumida podemos caracterizar o FHP pelos seus seguintes pressupostos [6, 10, 11, 46]:

- A massa de cada partícula é constante e igual a 1;
- O movimento das partículas é restringido às seis direcções numeradas (Figura 2.1);
- Em cada instante de tempo, cada partícula migra para o nó vizinho correspondente à direcção do vector de velocidade a si associado, não sendo permitidos “saltos” para qualquer outro nó;
- Em cada instante de tempo, ao longo de determinada direcção, apenas uma partícula se pode mover (princípio da exclusão);

A regra evolutiva do modelo é baseada em termos de colisões e movimento. Sendo o movimento das partículas de um nó actual para outro adjacente de acordo com um

conjunto de velocidades discretas, c_i , definidas consoante a geometria *lattice* usada. Assim, uma partícula localizada em \vec{x} no momento de tempo t com a velocidade c_i mover-se-á para a posição $\vec{x} + c_i\Delta t$ no momento de tempo $t+1$, o que matematicamente pode ser expresso por,

$$n_i(\vec{c} + c_i\Delta t, t + 1) = n_i(\vec{x}, t) \quad (2.6)$$

em que n_i é o número ocupação binária em cada nó que pode tomar o valor 0 ou 1 atendendo à presença ou ausência de partícula, por outras palavras:

$$\begin{aligned} n_i(\vec{x}, t) &= 0 \text{ ausência de partículas na posição } \vec{x} \text{ e instante } t \\ n_i(\vec{x}, t) &= 1 \text{ presença de partículas na posição } \vec{x} \text{ e instante } t \end{aligned} \quad (2.7)$$

Para o caso de rede *lattice* hexagonal, ilustrada na Figura 2.1, apenas se podem verificar seis diferentes tipos de colisão de partículas, os quais se encontram esquematizados na Figura 2.2.

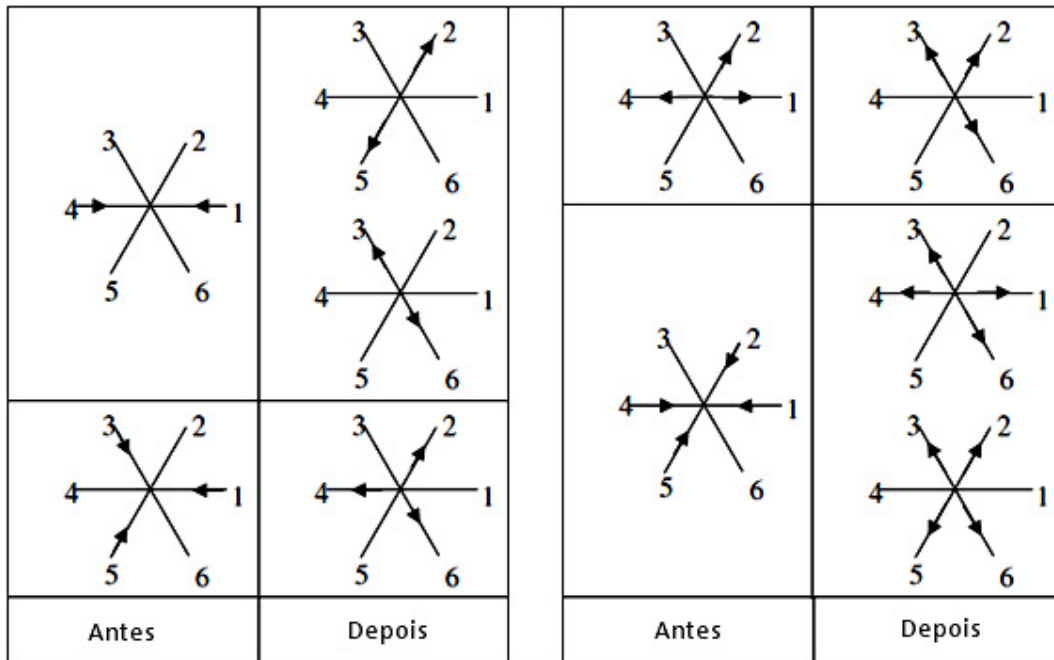


Figura 2.2 - Todas colisões FHP presentes no *lattice* hexagonal [11].

Apesar do sucesso do modelo enunciado (FHP), continuavam a existir discrepâncias para com soluções assentes nas equações de Navier-Stokes. As principais limitações apontadas ao LGCA estavam relacionadas com os seguintes problemas [11]:

- O modo de processamento das velocidades discretas na rede *lattice*;
- As regras inerentes ao operador de colisão apresentavam falhas;
- Presença de ruído;

Perante os resultados animadores do FHP, e atendendo aos desafios que as limitações enunciadas lançaram, durante o final dos anos 80 e início dos 90, muitos esforços foram feitos no sentido melhorar o comportamento dos modelos assentes no LGCA [47 - 53], entre outros.

Por entre os sucessivos modelos e trabalhos realizados, em 1988, McNamara e Zanetti publicaram um artigo [54] que introduziu o LBM como um novo método para simulações hidrodinâmicas.

O contributo dado pelos diversos autores no referido período, culminou na substituição das representações booleanas de partículas, características do LGCA, por funções de distribuição de partículas; reduziram-se os problemas de ruído; e foram propostos novos operadores de colisão, sendo estas as principais distinções que se estabelecem entre o LGCA e o LBM. O operador de colisão baseado na aproximação de Bhatnagar-Gross-Krook (BGK) (eq. 2.4) foi introduzido no LBM em 1992 por Quian et al. [55] e Chen et al. [56]. Esta implementação tornou o operador de colisão do LBM mais eficaz, garantindo a conservação de massa e de quantidade de movimento durante a colisão mediante uma expressão de cálculo mais simples.

Em termos de implementação, tal como acontece no LGCA, o LBM utiliza uma rede *lattice*. Contudo, no LBM dependendo da aplicação, diferentes redes *lattice* podem ser utilizadas. As mesmas são apresentadas pela notação $D\alpha Q\beta$, em que α é o número de dimensões e β o número de velocidades *lattice*. O LBM pode ser descrito como sendo um conjunto de nós que modelam um fluido, servindo cada nó de “plataforma” ao movimento de partículas de fluido individuais. Todos os nós são actualizados em cada instante de tempo por regras simples e tendo em conta o comportamento dos nós adjacentes, sendo assim o movimento global do fluido definido pela interacção entre o conjunto de nós.

No presente trabalho serão utilizadas as redes *lattice* D2Q9 e D3Q19. Reparando no modelo D2Q9 representado na Figura 2.3, verifica-se que o mesmo é descrito por 8 vectores de velocidade que são ilustrados e numerados; é através de cada um deles que as partículas podem migrar para nós vizinhos; assim, partindo do centro, as partículas podem-se deslocar para cada vértice ou para cada ponto médio dos quadrados vizinhos, a nona velocidade é contabilizada no centro do nó para os casos em que a partícula se mantém em equilíbrio. No que toca ao modelo D3Q19 apresentado no lado direito da mesma Figura 2.3, as velocidades *lattice* são também ilustradas e numeradas, partindo cada uma do centro para cada face o cubo envolvente, e até aos pontos médios de cada aresta do cubo.

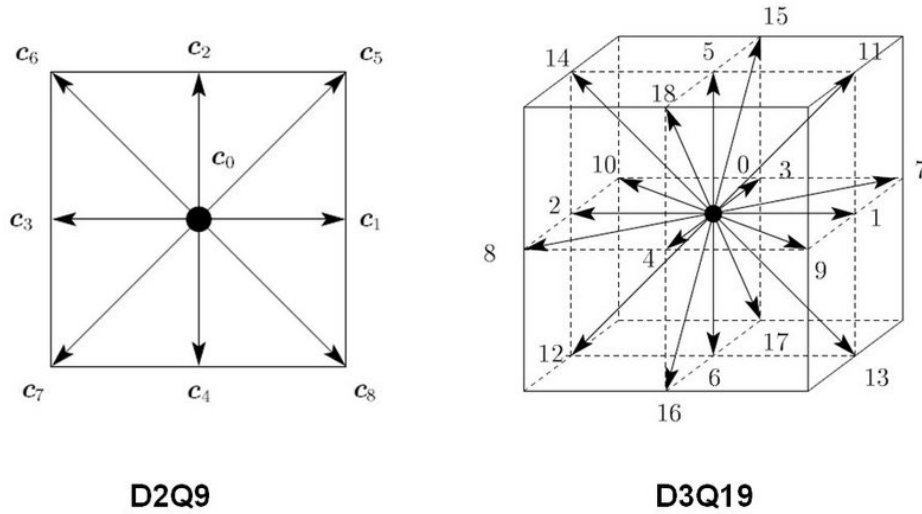


Figura 2.3 - Representação de nós *lattice* D2Q9 e D3Q19 [57].

Para simular o comportamento do fluido, todos os nós armazenam o número de partículas que se deslocam ao longo dos seus vetores *lattice*. Este processo é realizado pelas funções de distribuição de partículas que por norma são designadas por f_i , em que i é o correspondente número de vetores *lattice*. Assumindo que cada partícula tem a mesma massa 1, a partir das funções de distribuição de partículas duas propriedades físicas importantes do fluido podem ser calculadas. Como a função distribuição de partículas contém a quantidade de partículas que se deslocam numa dada direcção em cada nó, o somatório de todas essas funções de distribuição em cada nó dá a densidade (massa por volume) (eq. 2.8) [32]. A outra propriedade importante que pode ser extraída de cada nó é a velocidade e a direcção global em que as partículas presentes num nó se deslocam, este processo resulta da soma de todas as distribuições de funções de partículas multiplicadas por cada vector *lattice*, posteriormente divididos pela densidade (eq. 2.9) [32].

$$\rho = \sum_{i=0}^{\beta-1} f_i \quad (2.8)$$

$$\vec{u} = \frac{1}{\rho} \sum_{i=0}^{\beta-1} f_i \vec{c}_i \quad (2.9)$$

O algoritmo do LBM executa duas etapas em cada instante de tempo: uma de movimento e outra de colisão. A etapa de movimento é responsável pela deslocação das partículas na malha, direccionando as funções de distribuição das mesmas; a etapa de colisão dá conta dos choques que se dão entre as partículas no caso de mais que uma chegue simultaneamente ao mesmo nó. Nas Figuras 2.4 e 2.5 encontram-se representados

nós que contêm as funções de distribuição de partículas antes e depois das deslocações (modelo D2Q9) [58].

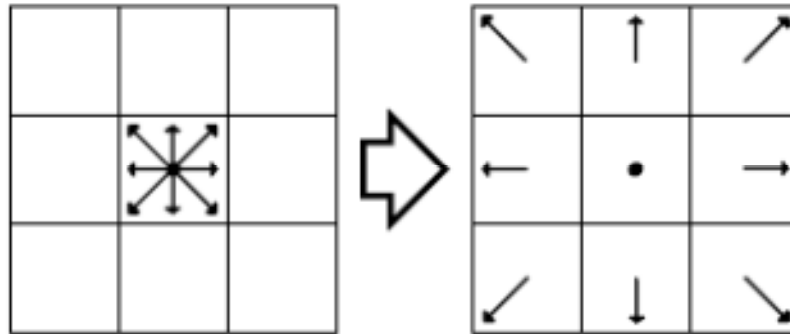


Figura 2.4 - Função de distribuição de partículas de uma célula antes e depois da etapa de movimento, adaptado de [58].

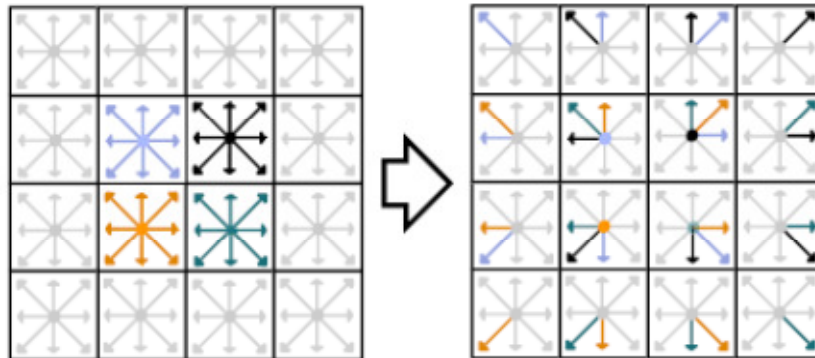


Figura 2.5 - Função de distribuição de partículas de quatro células vizinhas antes e depois da etapa de movimento, adaptado de [58].

Ambas as etapas descritas podem ser combinadas numa equação, que geralmente na literatura aparece na seguinte forma [11]:

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{\Delta t}{\tau} (f_i - f_i^{eq}) \quad (2.10)$$

A equação 2.10 é conhecida como equação *lattice* Boltzmann. Por norma, assume-se que a escala de tempo Δt é 1. O lado esquerdo da equação tem em conta a etapa de movimento. Por seu turno o membro do lado direito é responsável pelas colisões combinando as funções de distribuição de partículas antes da colisão com a função de distribuição de equilíbrio.

Por sua vez a função de distribuição de equilíbrio tem de ser calculada a partir das velocidades e densidades iniciais, sendo para o caso geral dada por [11]:

$$f_i^{eq} = \rho w_i \left[1 + \frac{3}{c_s^2} c_i \cdot u + \frac{9}{2c_s^4} (c_i \cdot u)^2 - \frac{3}{2c_s^2} u \cdot u \right] \quad (2.11)$$

Dependendo da rede *lattice* utilizada a equação 2.11 assume diferentes velocidades *lattice* c_i e factores de ponderação w_i , que são soluções aproximadas de funções de equilíbrio globais associadas à rede *lattice* em causa [6]. Assim para o modelo D2Q9 temos as seguintes velocidades *lattice* c_i e factores de ponderação w_i :

$$\begin{aligned}
 c_0 &= 0 & i &= 0 \\
 c_i &= c \left[\cos\left(\frac{(i-1)\pi}{2}\right), \sin\left(\frac{(i-1)\pi}{2}\right) \right] & i &= 1,2,3,4 \\
 c_i &= \sqrt{2}c \left[\cos\left(\frac{(2i-1)\pi}{4}\right), \sin\left(\frac{(2i-1)\pi}{4}\right) \right] & i &= 5,6,7,8
 \end{aligned} \tag{2.12}$$

$$w_i = \begin{cases} \frac{4}{9} & i = 9 \\ \frac{1}{9} & i = 1,2,3,4 \\ \frac{1}{36} & i = 5,6,7,8 \end{cases} \tag{2.13}$$

Por seu turno, no modelo D3Q19 as velocidades *lattice* c_i e os factores de ponderação w_i são:

$$\begin{aligned}
 c_1 &= (1,0,0) & c_2 &= (0,1,0) & c_3 &= (-1,0,0) \\
 c_4 &= (0,-1,0) & c_5 &= (0,0,1) & c_6 &= (0,0,-1) \\
 c_7 &= (1,1,0) & c_8 &= (1,-1,0) & c_9 &= (-1,-1,0) \\
 c_{10} &= (-1,1,0) & c_{11} &= (1,0,1) & c_{12} &= (0,1,1) \\
 c_{13} &= (-1,0,1) & c_{14} &= (0,-1,1) & c_{15} &= (1,0,-1) \\
 c_{16} &= (0,-1,-1) & c_{17} &= (-1,0,-1) & c_{18} &= (0,1,-1)
 \end{aligned} \tag{2.14}$$

$$w_i = \begin{cases} \frac{1}{3} & i = 9 \\ \frac{1}{18} & i = 1,2,3,4,5,6 \\ \frac{1}{36} & i = 5,6,7,8,9,10,11,12,13,14,15,16,17,18 \end{cases} \tag{2.15}$$

Para comprovar que a equação de Boltzmann, e consequentemente que a equação *lattice* Boltzmann (já que ambas possuem a mesma fundamentação física), podem ser utilizadas para descrever o comportamento de fluidos, as equações de Navier-Stokes podem ser deduzidas por um processo denominado de expansão de Chapman-Enskog ou análise multi-escala [6, 59 - 62]. Assim, assumindo um *número de Mach* pequeno, ou seja uma baixa relação entre a velocidade do escoamento e a velocidade local do som; e servindo-se do *número de Knudsen*, que é o rácio entre o comprimento médio livre λ (distância média entre duas colisões sucessivas) e a escala espacial característica do sistema, o método multi-escala de Chapman-Enskog mostra que a equação da continuidade

(eq. 2.16) e de Navier-Stokes para escoamentos incompressíveis (eq. 2.17) [32] são recuperadas.

$$\nabla \cdot \vec{u} = 0 \quad (2.16)$$

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{\nabla P}{\rho} + \vartheta \cdot \nabla^2 u \quad (2.17)$$

De forma a sintetizar os conceitos abordados no presente Capítulo, a Figura 2.6 ilustra de forma esquemática a evolução da equação de Boltzmann até à equação *lattice* Boltzmann, assim como o processo dedutivo que recupera as equações de Navier-Stokes da dinâmica dos fluidos tradicional.

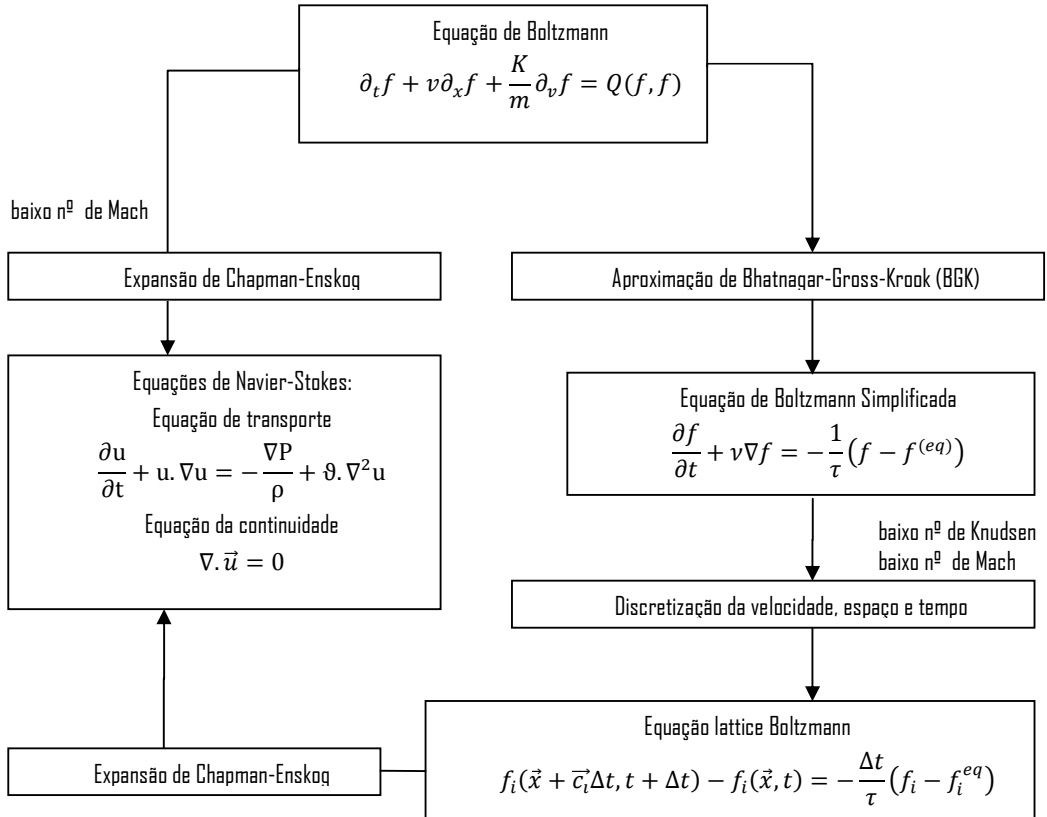


Figura 2.6 - Esquemática do processo dedutivo do LBM, adaptado de [38].

Comparando agora a equação *lattice*-Boltzmann (ELB) com as equações Navier-Stokes, com base na Figura 2.6, verificam-se as diferenças seguidamente listadas [11, 46, 49 - 51]:

- Enquanto a LBE é obtida a partir de uma equação diferencial parcial (PDE) de primeira ordem, enquanto as equações de Navier-Stokes são PDEs de segunda ordem.

- O operador convectivo das equações de Navier-Stokes é um termo não linear, por outro lado todos os termos da LBE são lineares devido ao operador de colisão simples LBGK.
- Devido à natureza mesoscópica (escala de grandeza entre a macroscópica e microscópica) do LBM, meios porosos de escala micro podem ser facilmente simulados, o que faz dele uma ferramenta poderosa em simulações na escala micro e nano.

2.3. Condições Fronteira

Os diversos problemas de escoamentos de fluidos são fortemente influenciados pelo ambiente que os envolve. Matematicamente, e no que toca a simulações CFD a influência criada pelas envolturas é definida e prescrita por condições fronteira.

A implementação das condições fronteira define o modo como as partículas de fluido vão interagir com as paredes sólidas que confinam o escoamento, com eventuais obstáculos presentes no domínio, e define também o seu comportamento à entrada e saída do domínio de estudo.

Neste âmbito muitas metodologias e procedimentos vêm sendo estudados e testados desde o surgimento do LBM [63 - 69].

No presente trabalho, por se tratar de simulações de escoamentos em meios porosos de geometrias complexas, a implementação de condições fronteira que retratem com eficácia as inúmeras colisões com meios sólidos é necessária. Uma técnica muito bem sucedida e que é recorrentemente utilizada é a conhecida regra *bounce-back*. Este procedimento é de implementação relativamente simples e consegue retratar com alguma precisão o comportamento do fluido na imediação de objectos sólidos (obstáculos e paredes), como é explicado na próxima secção.

O código considerado no presente trabalho utiliza a condição fronteira baseada no *bounce-back*, o *half-way bounce-back*, para considerar as colisões entre as partículas de fluido e os objectos sólidos/paredes que confinam o escoamento e na entrada e saída do escoamento foram implementadas condições fronteira periódicas. Estes dois tipos de condição fronteira serão tratados seguidamente.

2.3.1. Bounce-Back

As primeiras configurações do *bounce-back* surgiram em modelos LGCA [45, 48], tendo sido depois integradas nos modelos LBM.

No que toca à interacção existente nas interfaces fluido/sólido, o *bounce-back* caracteriza-se como um simples ressalto, definindo que as funções de distribuição das partículas de fluido ao serem transferidas para nós da parte sólida são reflectidas para os

nós de fluido de onde provieram, com igual grandeza e sentidos opostos, como a Figura 2.7 ilustra.

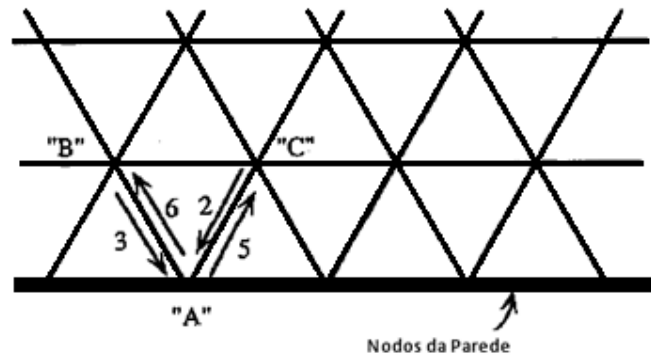


Figura 2.7 - Descrição esquemática do *bounce-back*, adaptado de [63].

Este procedimento relativamente simples, torna a implementação do *bounce-back* bastante atractiva em termos computacionais, contudo diversos trabalhos vieram a comprovar que a aplicação do *bounce-back* para a maioria das aplicações possui pouca precisão. Assim, na tentativa de obter comportamentos mais realistas junto às paredes, surgiram dois tipos particulares de condição fronteira baseados no *bounce-back* [67]: o *half-way bounce-back* (ressalto a meio caminho) e o *bounce-forward* (ressalto para a frente). O *half-way bounce-back* executa exactamente o mesmo processo enunciado acerca do *bounce-back*, mas com a particularidade de colocar a interface da parede a meio caminho entre o nó de fluido e o nó de sólido. Por seu lado o *bounce-forward* difere do anterior por não reflectir as funções de distribuição em sentido contrário mas sim para o nó de fluido adjacente (ressalto para a frente). A Figura 2.8 ajuda a perceber os dois tipos de condição fronteira enunciados.

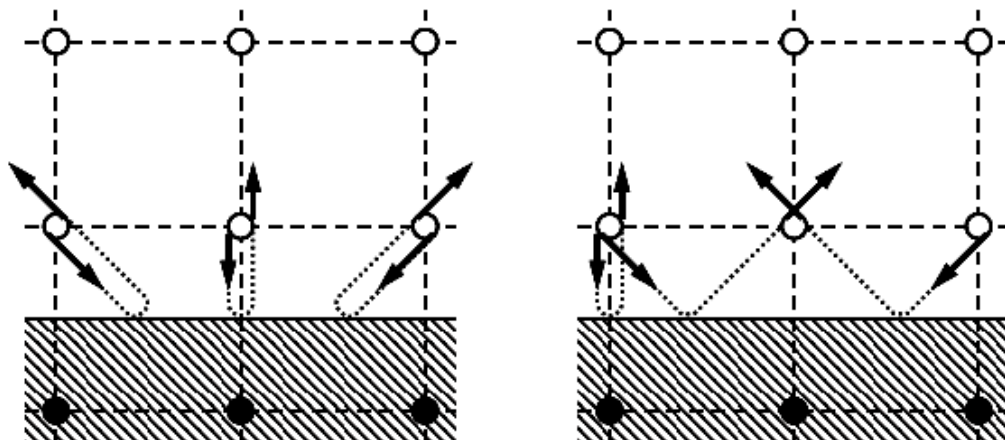


Figura 2.8 - Descrição esquemática do *halfway bounce-back* (à esquerda) e do *bounce-forward* (à direita) [70].

Referir ainda que na literatura, é comum aparecer o *bounce-back* e o *half-way bounce-back* associados às designadas condições fronteira sem escorregamento (*no-slip boundary*

condition) e o *bounce-forward* às designadas condições fronteira com escorregamento (*slip boundary conditions*). Na Figura 2.9 verifica-se na parede esquerda uma condição fronteira sem escorregamento, ou seja o ressalto efectuado pela partícula junto à parede é para trás (*bounce-back*); quanto à parede superior, em que se considera a presença de uma condição fronteira com escorregamento constata-se que o ressalto da partículas após colidir com a parede é para a frente (*bounce-back forward*).

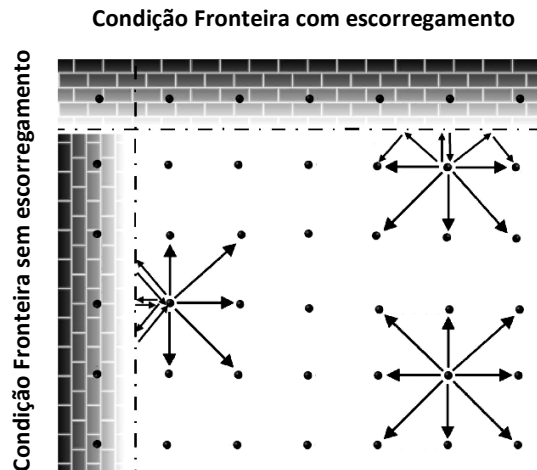


Figura 2.9 - Ilustração esquemática da acção do *half-way-bounce-back* e do *bounce-forward*, adaptado de [70].

2.3.2. Condições Fronteira Periódicas

As condições fronteira periódicas são relativamente simples de implementar [10]. Em termos de simulações LBM, são muito utilizadas para definir o comportamento do fluido à entrada e saída do domínio em estudo.

Este tipo de condição fronteira trata os nós de saída do escoamento como sendo coincidentes com os de entrada, definindo que a etapa de movimento de partículas verificada nos nós de saída do escoamento é direccionada para os nós da entrada [66]. O efeito das condições fronteira periódicas pode ser visto como um caso repetição do comportamento do escoamento. Na Figura 2.10 encontra-se representado um conjunto de partículas de fluido (modelo D2Q9), apresentando-se a forma como as partículas de fluido interagem nas fronteiras de entrada e saída [11]. Assim, constata-se que nos nós de saída, as funções de distribuição de partículas migram (por indicação das setas) para os nós de entrada (situados no lado oposto), fazendo com que o escoamento tenha um comportamento cíclico/repetido.

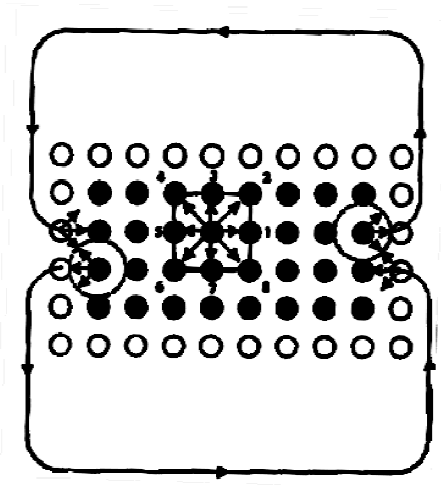


Figura 2.10 - Condições fronteira periódicas, em que se nota a interacção entre nós de saída e entrada [11].

3. CÓDIGO NUMÉRICO

Na presente dissertação foram utilizados dois códigos LBM disponíveis para consulta aberta na Internet em <http://www.exolete.com/code/lbm> [71]. Trata-se de dois códigos escritos em MATLAB, sendo a sua estrutura clara e concisa [71].

De modo a facilitar a compreensão do programa, o autor colocou na parte superior dos códigos um pequeno texto e uma figura que ilustra as direcções *lattice* assumidas, no sentido de ajudar a leitura dos mesmos.

Seguidamente, o código D2Q9 (Algoritmo 3.1) será exposto detalhadamente, em pequenos excertos por entre o texto explicativo.

Assim, o código inicia-se na linha 7 (Algoritmo 3.1.1.) com a definição de alguns parâmetros constantes para as funções de equilíbrio (eq. 2.11) como a densidade, factores de ponderação, a velocidade *lattice* e o número de nós do domínio. Seguidamente, na linha 8 são geradas as matrizes que vão servir de base às etapas de propagação das partículas (matriz “F”) e das funções de distribuição de equilíbrio (matriz “FEQ”); na mesma linha é ainda calculada a área do domínio (“msize”) e criado o vector “CI”, este vector será mais tarde útil para o cálculo do *half-way bounce back* (regra de colisões com objectos sólidos caracterizada no Capítulo 2). De seguida, na linha 11 é definida a matriz “BOUND” que é a responsável pela definição do meio poroso, ou seja, define e diferencia os nós de fluido e de sólido, tomando os nós de fluido o valor “0” e os sólidos valor “1”; no caso utiliza a função do MATLAB “rand”, que gera uma matriz de dimensão “nx x ny” de “0s” e “1s” inseridos de forma aleatória.

Algoritmo 3.1.1. – Excerto das primeiras 11 linhas do código D2Q9

```
1 %           2D Lattice Boltzmann (BGK) model of a fluid.
2 % c4 c3 c2 D2Q9 model. At each timestep, particle densities propagate
3 % \ / outwards in the directions indicated in the figure. An
4 % c5 -c9 - c1 equivalent 'equilibrium' density is found, and the densities
5 % / | \ relax towards that state, in a proportion governed by omega.
6 % c6 c7 c8 Iain Haslam, March 2006.
7 omega=1.0; density=1.0; t1=4/9; t2=1/9; t3=1/36; c_squ=1/3; nx=31; ny=31;
8 F= repmat(density/9,[nxny 9]); FEQ=F; msize=nx*ny; CI=[0:msize:msize*7];
9 %a= repmat(-15:15,[31,1]); BOUND=(a.^2+a'.^2)<16; BOUND(1:nx,[1 ny])=1;
10 %BOUND=zeros(nx,ny); BOUND(1:nx,1)=1;%open channel
11 BOUND=rand(nx,ny)>0.7; %extremely porous random domain
```

Posto isto, na linha 12 (Algoritmo 3.1.2.) o vector “ON” identifica quais os nós que assumem valor “1”; sendo depois definidas as matrizes “TO-REFLECT” e “REFLECTED” que constituirão elemento para a concretização do *half-way bounce-back*, já que identificam os nós junto à fronteira do meio sólido do de fluido. Seguidamente, na linha 18 são definidos o “prevavu” e o “avu”, que servirão para a definição dos instantes de tempo necessários à execução da simulação; por sua vez o “deltaU” é um incremento de pressão que servirá de condição inicial ao movimento do fluido na entrada do domínio.

Algoritmo 3.1.2. - Excerto das linhas 12 à 17 do código D2Q9

```

12 ON=find(BOUND); %matrix offset of each Occupied Node
13 TO_REFLECT=[ON+CI(1) ON+CI(2) ON+CI(3) ON+CI(4) ...
14 ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8)];
15 REFLECTED= [ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8) ...
16 ON+CI(1) ON+CI(2) ON+CI(3) ON+CI(4)];
17 avu=1; prevavu=1; ts=0; deltaU=1e-7; numactivenodes=sum(sum(1-BOUND));

```

Na linha 18 (Algoritmo 3.1.3.) do código inicia-se o ciclo “while”, com a indicação do modo de quantificação dos instantes de tempo. Logo a seguir, da linha 19 até à 23 encontram-se definidas as hipóteses de propagação, ou seja são definidas as 8 diferentes direcções em que as partículas de fluido se podem movimentar.

O *half-way bounce-back* é definido para o próximo instante de tempo pela matriz “BOUNCEDBACK” e logo depois a densidade em cada nó é calculada (matriz “DENSITY”) como tratado no Capítulo 2 (eq. 2.8). Nas linhas 26 e 27 surgem as velocidades presentes em cada nó segundo a respectiva direcção “UX” e “UY”; e de seguida ao impor um aumento de pressão (“deltaU”) ao fluido na primeira coluna da matriz de velocidades, surge a forma como o movimento do fluido é induzido à entrada no domínio; ainda em relação às velocidades, na linha 30 surgem as velocidades das restantes direcções *lattice* que vão entrar na equação de equilíbrio das respectivas direcções.

Algoritmo 3.1.3. – Excerto das linhas 18 à 30 do código D2Q9

```

18 while(ts<4000 & 1e-10<abs((prevavu-avu)/avu)) | ts<100
19 % Propagate
20 F(:,4)=F([2:nx 1],[ny 1:ny-1],4);F(:,3)=F(:,ny 1:ny-1,3);
21 F(:,2)=F([nx 1:nx-1],[ny 1:ny-1],2);F(:,5)=F([2:nx 1],,5);
22 F(:,1)=F([nx 1:nx-1],,1);F(:,6)=F([2:nx 1],[2:ny 1],6);
23 F(:,7)=F(:,2:ny 1,7); F(:,8)=F([nx 1:nx-1],[2:ny 1],8);
24 BOUNCEDBACK=F(TO_REFLECT); %Densities bouncing back at next timestep
25 DENSITY=sum(F,3);
26 UX=(sum(F(:,,[1 2 8]),3)-sum(F(:,,[4 5 6]),3))./DENSITY;
27 UY=(sum(F(:,,[2 3 4]),3)-sum(F(:,,[6 7 8]),3))./DENSITY;
28 UX(1,1:ny)=UX(1,1:ny)+deltaU; %Increase inlet pressure
29 UX(ON)=0; UY(ON)=0; DENSITY(ON)=0;
30 U_SQU=UX.^2+UY.^2; U_C2=UX+UY; U_C4=-UX+UY; U_C6=-U_C2; U_C8=-U_C4;

```

Depois, na linha 32 (Algoritmo 3.1.4.) começam os cálculos referentes às funções de distribuição de equilíbrio (eq. 2.11), especificamente para a posição de equilíbrio (posição *lattice* c_9); até à linha 42 aplica-se a mesma equação para cada direcção *lattice*. Depois disso, temos na linha 43 o cálculo de “F” a dar conta da posição e estado do domínio em cada nó depois de atingido o equilíbrio no instante em causa; enquanto na linha seguinte se identificam os nós em que a condição de *half-way bounce back* se verificou no instante de tempo que finda. Seguidamente é averiguado quando é que o número de nós de fluido (“numactivenodes”) igualam o número de nós que possuem velocidades associadas, caso essa condição não se verifique o ciclo “while” recomeçam os cálculos de propagação, velocidades, colisões e funções de distribuição de equilíbrio enunciadas; quando atingida a condição necessária o ciclo “while” termina.

Algoritmo 3.1.4. - Excerto das linhas 31 à 46 do código D2Q9

```

31 % Calculate equilibrium distribution: stationary
32 FEQ(:,9)=t1*DENSITY.*(1-U_SQU/(2*c_squ));
33 % nearest-neighbours
34 FEQ(:,1)=t2*DENSITY.*(1+UX/c_squ+0.5*(UX/c_squ).^2-U_SQU/(2*c_squ));
35 FEQ(:,3)=t2*DENSITY.*(1+UY/c_squ+0.5*(UY/c_squ).^2-U_SQU/(2*c_squ));
36 FEQ(:,5)=t2*DENSITY.*(1-UX/c_squ+0.5*(UX/c_squ).^2-U_SQU/(2*c_squ));
37 FEQ(:,7)=t2*DENSITY.*(1-UY/c_squ+0.5*(UY/c_squ).^2-U_SQU/(2*c_squ));
38 % next-nearest neighbours
39 FEQ(:,2)=t3*DENSITY.*(1+U_C2/c_squ+0.5*(U_C2/c_squ).^2-U_SQU/(2*c_squ));
40 FEQ(:,4)=t3*DENSITY.*(1+U_C4/c_squ+0.5*(U_C4/c_squ).^2-U_SQU/(2*c_squ));
41 FEQ(:,6)=t3*DENSITY.*(1+U_C6/c_squ+0.5*(U_C6/c_squ).^2-U_SQU/(2*c_squ));
42 FEQ(:,8)=t3*DENSITY.*(1+U_C8/c_squ+0.5*(U_C8/c_squ).^2-U_SQU/(2*c_squ));
43 F=omega*FEQ+(1-omega)*F;
44 F(REFLECTED)=BOUNCEDBACK;
45 prevavu=avu;avu=sum(sum(UX))/numactivenodes; ts=ts+1;
46 end

```

Por fim, as últimas linhas de código (Algoritmo 3.1.5.) definem o modo de apresentação dos resultados; na linha 47 define-se que o domínio apenas apresenta a cor preta e branca, de modo a distinguir os nós de fluido (brancos) dos sólidos (pretos); na linha 48 é implementada a função “quiver” que é responsável pela apresentação de um gráfico de vectores velocidade azuis representantes da grandeza e direcção de cada partícula/nó de fluido; por fim na linha 49 apenas se inserem as legendas e título na figura a gerar. Adiante, na Figura 3.1 será apresentado um resultado típico (já que o algoritmo gera uma matriz aleatória para definir o meio poroso) do código enunciado.

Algoritmo 3.1.5. - Excerto das linhas 47 à 49 do código D2Q9

```

47 figure;colormap(gray(2));image(2-BOUND');hold on;
48 quiver(2:nx,1:ny,UX(2:nx,:),UY(2:nx,:));
49 title(['Flow field after ',num2str(ts),'\deltat']);xlabel('x');ylabel('y');

```

O Algoritmo 3.2, referente ao modelo D3Q19, foi concebido exactamente assente na mesma metodologia do descrito anteriormente. Todo o código possui a mesma configuração, destacando-se as inevitáveis diferenças provenientes de mais uma componente dimensional, “eixo z”, no caso dos vectores de velocidade, e no modo de visualização que recorre a cortes em duas secções do escoamento para apresentar os resultados da simulação. Os cortes feitos ao campo do escoamento tridimensional são úteis, já que ilustram os vectores de velocidade segundo as direcções do sistema de eixos dos respectivos cortes. Mais adiante, nas Figuras 3.2 e 3.3 serão apresentados os resultados obtidos deste algoritmo.

Algoritmo 3.2 – Código LBM para o modelo D3Q19

```

1 % 3D Lattice Boltzmann (BGK) model of a fluid.
2 % D3Q19 model. At each timestep, particle densities propagate
3 % outwards in the directions indicated in the figure. An
4 % equivalent 'equilibrium' density is found, and the densities
5 % relax towards that state, in a proportion governed by omega.
6 % Iain Haslam, March 2006.
7 nx=12;ny=nx;nz=nx; omega=1.0; density=1.0;t1=1/3; t2=1/18; t3=1/36;
8 F= repmat(density/19,[nxny nz]); FEQ=F; matsize=nx*ny*nz;
9 CI=[0:matsize:matsize*19];
10 BOUND=zeros(nx,ny,nz);
11 for i=1:nx, for j=1:ny, for k=1:nz
12 BOUND(i,j,k)=((i-5)^2+(j-6)^2+(k-7)^2)<6;
13 end , end, end
14 BOUND(:,,1)=1;BOUND(:,,2)=1;
15 ON=find(BOUND); %matrix offset of each Occupied Node
16 TO_REFLECT=[ON+CI(2) ON+CI(3) ON+CI(4) ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8) ...
17 ON+CI(9) ON+CI(10) ON+CI(11) ON+CI(12) ON+CI(13) ON+CI(14) ON+CI(15) ...
18 ON+CI(16) ON+CI(17) ON+CI(18) ON+CI(19)];
19 REFLECTED=[ON+CI(3) ON+CI(2) ON+CI(5) ON+CI(4) ON+CI(7) ON+CI(6) ON+CI(11) ...
20 ON+CI(10) ON+CI(9) ON+CI(8) ON+CI(15) ON+CI(14) ON+CI(13) ON+CI(12) ...
21 ON+CI(19) ON+CI(18) ON+CI(17) ON+CI(16)];
22 avu=1; prevavu=1; ts=0; deltaU=1e-7; numactivenodes=sum(sum(sum(1-BOUND)));
23 while (ts<4000 & 1e-10<abs((prevavu-avu)/avu)) | ts<100
24 % Propagate
25 %nearest-neighbours
26 F(:,,2)=F(:,,[nz 1:nz-1],2);
27 F(:,,3)=F(:,,[2:nz 1],3);
28 F(:,,4)=F(:,,[ny 1:ny-1],,4);
29 F(:,,5)=F(:,,[2:ny 1],,5);
30 F(:,,6)=F([nx 1:nx-1],,6);
31 F(:,,7)=F([2:nx 1],,7);
32 %next-nearest neighbours
33 F(:,,8)= F([nx 1:nx-1],[ny 1:ny-1],,8);
34 F(:,,9)= F([nx 1:nx-1],[2:ny 1],,9);
35 F(:,,10)=F([2:nx 1],[ny 1:ny-1],,10);
36 F(:,,11)=F([2:nx 1],[2:ny 1],,11);
37 F(:,,12)=F([nx 1:nx-1],[nz 1:nz-1],12);
38 F(:,,13)=F([nx 1:nx-1],[2:nz 1],13);
39 F(:,,14)=F([2:nx 1],[nz 1:nz-1],14);
40 F(:,,15)=F([2:nx 1],[2:nz 1],15);
41 F(:,,16)=F(:,,[ny 1:ny-1],[nz 1:nz-1],16);
42 F(:,,17)=F(:,,[ny 1:ny-1],[2:nz 1],17);
43 F(:,,18)=F(:,,[2:ny 1],[nz 1:nz-1],18);
44 F(:,,19)=F(:,,[2:ny 1],[2:nz 1],19);
45 BOUNCEDBACK=F(TO_REFLECT); %Densities bouncing back at next timestep
46 % Relax; calculate equilibrium state (FEQ) with equivalent speed and density toF

```

```

47 DENSITY = sum(F,4);
48 UX=(sum(F(:,:,,[6 8 9 12 13]),4)-sum(F(:,:,,[7 10 11 14 15]),4))./DENSITY;
49 UY=(sum(F(:,:,,[4 8 10 16 17]),4)-sum(F(:,:,,[5 9 11 18 19]),4))./DENSITY;
50 UZ=(sum(F(:,:,,[2 12 14 16 18]),4)-sum(F(:,:,,[3 13 15 17 19]),4))./DENSITY;
51 UX(1,:)=UX(1,:)+deltaU; %Increase inlet pressure
52 UX(ON)=0; UY(ON)=0; UZ(ON)=0; DENSITY(ON)=0; U_SQU=UX.^2+UY.^2+UZ.^2;
53 U8=UX+UY;U9=UX-UY;U10=-UX+UY;U11=-U8;U12=UX+UZ;U13=UX-UZ;
54 U14=-U13;U15=-U12;U16=UY+UZ;U17=UY-UZ;U18=-U17;U19=-U16;
55 % Calculate equilibrium distribution: stationary
56 FEQ(:,:,,1)=t1*DENSITY.*(1-3*U_SQU/2);
57 % nearest-neighbours
58 FEQ(:,:,,2)=t2*DENSITY.*(1 + 3*UZ + 9/2*UZ.^2 - 3/2*U_SQU);
59 FEQ(:,:,,3)=t2*DENSITY.*(1 - 3*UZ + 9/2*UZ.^2 - 3/2*U_SQU);
60 FEQ(:,:,,4)=t2*DENSITY.*(1 + 3*UY + 9/2*UY.^2 - 3/2*U_SQU);
61 FEQ(:,:,,5)=t2*DENSITY.*(1 - 3*UY + 9/2*UY.^2 - 3/2*U_SQU);
62 FEQ(:,:,,6)=t2*DENSITY.*(1 + 3*UX + 9/2*UX.^2 - 3/2*U_SQU);
63 FEQ(:,:,,7)=t2*DENSITY.*(1 - 3*UX + 9/2*UX.^2 - 3/2*U_SQU);
64 % next-nearest neighbours
65 FEQ(:,:,,8) =t3*DENSITY.*(1 + 3*U8 + 9/2*(U8).^2 - 3*U_SQU/2);
66 FEQ(:,:,,9) =t3*DENSITY.*(1 + 3*U9 + 9/2*(U9).^2 - 3*U_SQU/2);
67 FEQ(:,:,,10)=t3*DENSITY.*(1 + 3*U10 + 9/2*(U10).^2 - 3*U_SQU/2);
68 FEQ(:,:,,11)=t3*DENSITY.*(1 + 3*U11 + 9/2*(U11).^2 - 3*U_SQU/2);
69 FEQ(:,:,,12)=t3*DENSITY.*(1 + 3*U12 + 9/2*(U12).^2 - 3*U_SQU/2);
70 FEQ(:,:,,13)=t3*DENSITY.*(1 + 3*U13 + 9/2*(U13).^2 - 3*U_SQU/2);
71 FEQ(:,:,,14)=t3*DENSITY.*(1 + 3*U14 + 9/2*(U14).^2 - 3*U_SQU/2);
72 FEQ(:,:,,15)=t3*DENSITY.*(1 + 3*U15 + 9/2*(U15).^2 - 3*U_SQU/2);
73 FEQ(:,:,,16)=t3*DENSITY.*(1 + 3*U16 + 9/2*(U16).^2 - 3*U_SQU/2);
74 FEQ(:,:,,17)=t3*DENSITY.*(1 + 3*U17 + 9/2*(U17).^2 - 3*U_SQU/2);
75 FEQ(:,:,,18)=t3*DENSITY.*(1 + 3*U18 + 9/2*(U18).^2 - 3*U_SQU/2);
76 FEQ(:,:,,19)=t3*DENSITY.*(1 + 3*U19 + 9/2*(U19).^2 - 3*U_SQU/2);
77 F=omega*FEQ+(1-omega)*F;
78 F(REFLECTED)=BOUNCEDBACK;
79 prevavu=avu;avu=sum(sum(sum(UX)))/numactivenodes; ts=ts+1;
80 end
81 figure; zcut=5; colormap(gray(2)); image(2-BOUND(:,:,5)); hold on;
82 %Thanks to Thomas Wagner for correcting the transposed results plots
83 quiver(UY(:,:,zcut),UX(:,:,zcut)); xlabel('y'); ylabel('x');
84 title(['Flow field at z=', num2str(zcut), ', after ', num2str(ts), '\deltat']);
85 figure; ycut=5; colormap(gray(2)); image(2-squeeze(BOUND(:,ycut,:))); hold on;
86 quiver(squeeze(UZ(:,ycut,:)),squeeze(UX(:,ycut,:))); xlabel('z'); ylabel('x');
87 title(['Flow field at y=', num2str(ycut), ', after ', num2str(ts), '\deltat']);

```

Como foi referido, o código D2Q9 (Algoritmo 3.1. apresentado em excertos) gera um meio poroso (meio em que pode haver circulação de fluidos por entre objectos ou estruturas sólidas) por intermédio da matriz “BOUND” de 31x31 nós. O meio poroso exibido na Figura 3.1 possui os elementos sólidos a preto (nós sólidos) e elementos correspondentes a partículas de fluido representados por vectores a azul cuja grandeza e direcção é a que foi calculada pelo código para cada nó em questão. Com facilidade é possível obter a velocidade do escoamento em qualquer posição nodal, por exemplo, UX[5,5] é a componente “x” da velocidade no nó (5,5) e UY[5,5] é a componente “y” da velocidade no nó (5,5). De notar ainda que na parte superior é indicado os intervalos de tempo necessários à representação do escoamento em causa.

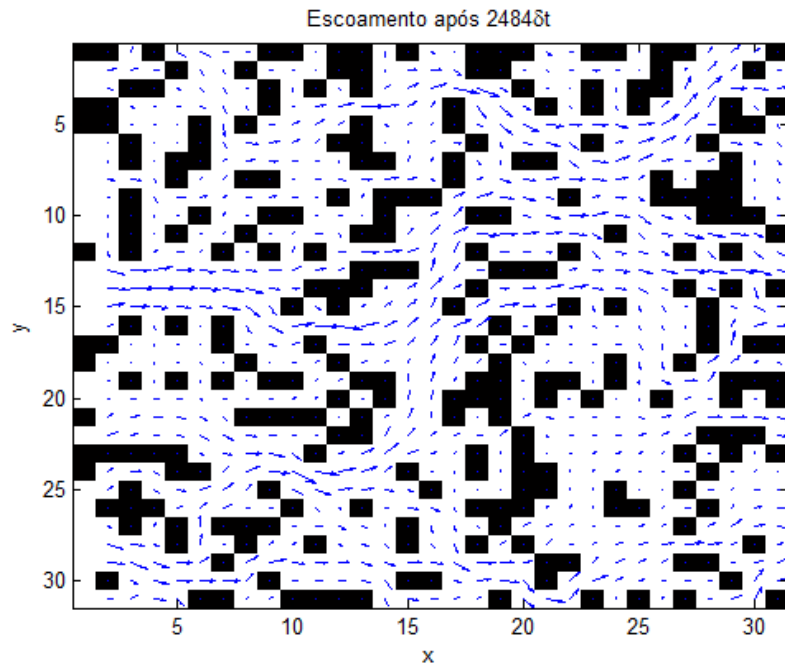


Figura 3.1 - Resultado do escoamento de fluido em meio poroso.

No que concerne ao código D3Q19 apresentado no Algoritmo 3.2, o resultado da simulação é um escoamento tridimensional em torno de esfera (matriz "BOUND" do Algoritmo 3.2) que devido à pouca definição do nós em causa (12x12) se aproxima a uma cruz no plano $z = 5$, como ilustrado na Figura 3.2; além disso, na mesma figura pode-se observar que o escoamento se desenvolve paralelamente a uma parede em $y = 1$ no plano xz .

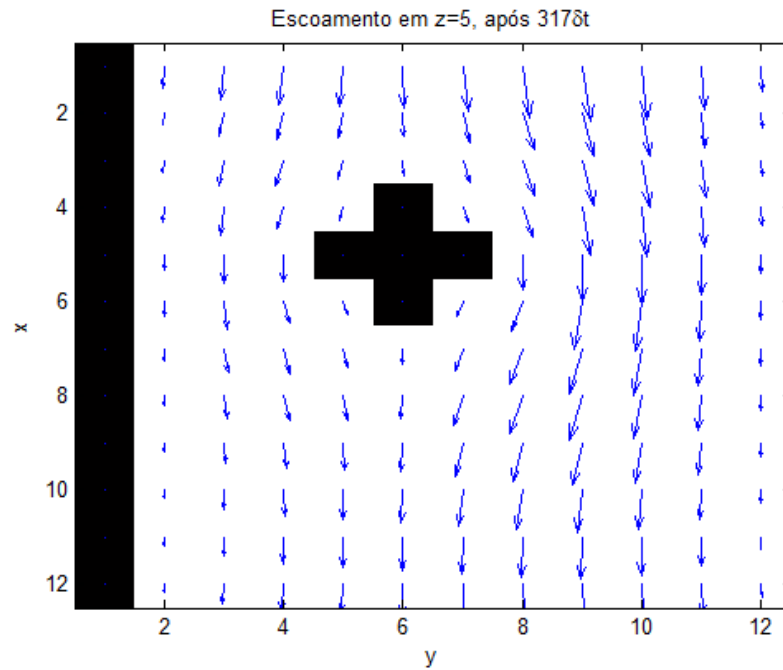


Figura 3.2 - Resultado do escoamento de fluido obtido na secção $z = 5$.

Por outro lado, a Figura 3.3 apresenta o escoamento no plano xz em $y = 5$. É possível também constatar a presença de mais uma parede, agora no plano xy para $z = 1$. Referência por fim para os 317 instantes de tempo corridos durante a simulação.

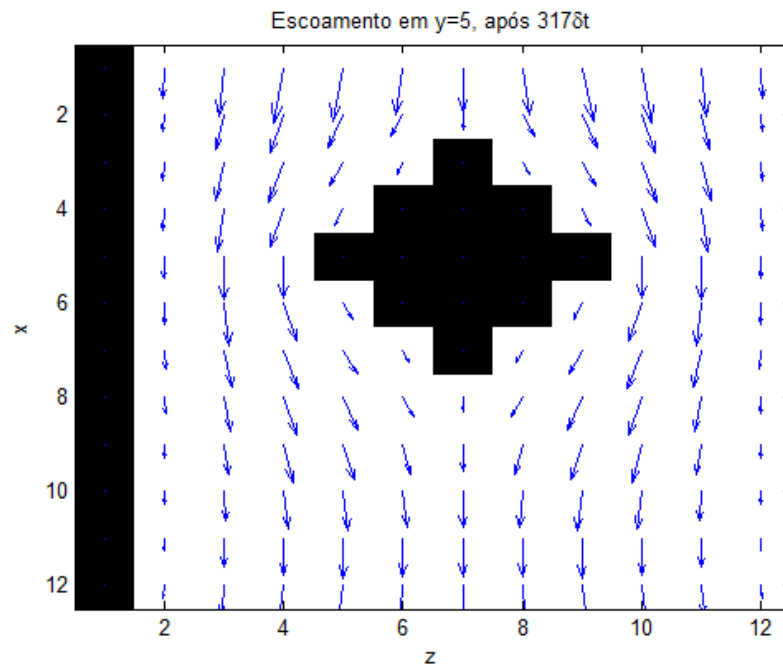


Figura 3.3 - Resultado do escoamento de fluido obtido na secção $y = 5$.

4. TESTES E VALIDAÇÃO

4.1. Simulação Numérica do Escoamento de *Poiseuille*

Na tentativa de estudar os resultados provenientes dos códigos e em particular a influência da resolução da malha, foram realizados pequenos testes.

A primeira geometria testada consistiu numa conduta bidimensional de modo a reproduzir um escoamento completamente desenvolvido. Assim, o objectivo passou por comparar as soluções analíticas do perfil de velocidades do escoamento de *Poiseuille*, com o perfil de velocidades do escoamento de soluções numéricas. Foram testados diferentes níveis de nós activos ao longo da altura, sendo os nós relativos ao comprimento irrelevantes para o caso devido à condição fronteira periódica imposta ao escoamento (não havendo presença de obstáculos o escoamento é completamente desenvolvido na entrada e saída). A Figura 4.1 apresenta a configuração do perfil de velocidades à saída do escoamento para uma simulação num domínio com 21x9 nós, a simulação conclui-se com 581 instantes de tempo e à saída é possível verificar o perfil parabólico da velocidade do escoamento (típico do escoamento de *Poiseuille*).

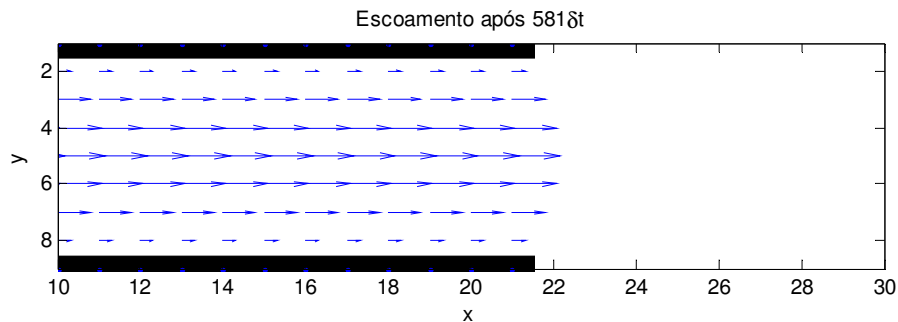


Figura 4.1 - Representação à saída de escoamento bidimensional entre duas paredes.

A solução analítica no escoamento de *Poiseuille* numa conduta bidimensional, (idêntica à representada na Figura 4.1), de altura $2L$ é descrita pela parábola:

$$u(y) = \frac{K}{2\nu} (L^2 - y^2) \quad (2.1)$$

Esta solução pressupõe a presença da condição de não escorregamento nas paredes, ou seja, $u(y) = 0$ em $y = -L$ e $y = L$. Considera também que o escoamento é constante

($\partial/\partial t = 0$); que a velocidade é descrita por $u = (u, v) = (u(y), 0)$; pressão constante (P e $\rho = \text{constante}$); e inexistência de variações na direcção x ($\partial/\partial x = 0$). Associado ao comportamento do escoamento encontra-se ainda a força externa K e a viscosidade cinemática ν . O código gerador das soluções analíticas para os diferentes testes executados encontram-se no Anexo I.

Simularam-se escoamentos para 21, 41, 61, 81 e 101 nós ao longo da altura da conduta. Os resultados obtidos são ilustrados nas Figuras 4.2, 4.3, 4.4, 4.5, 4.6, estando representada a solução analítica (a azul) e os pontos discretos da solução numérica (a vermelho) no gráfico do lado esquerdo; e o erro relativo ao longo da altura da conduta do lado direito (com a indicação do erro relativo máximo e médio).

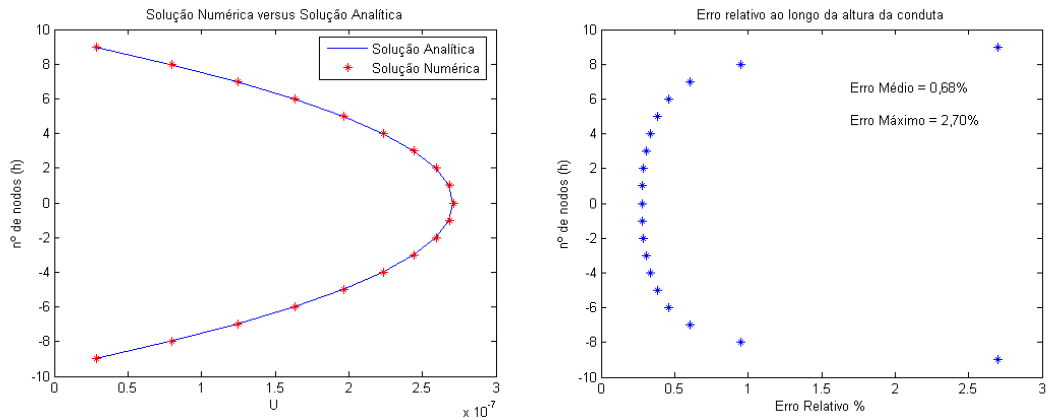


Figura 4.2 - Solução numérica e solução analítica do perfil de velocidades do escoamento de *Poiseuille* para simulação de 21 nós ao longo da altura da conduta.

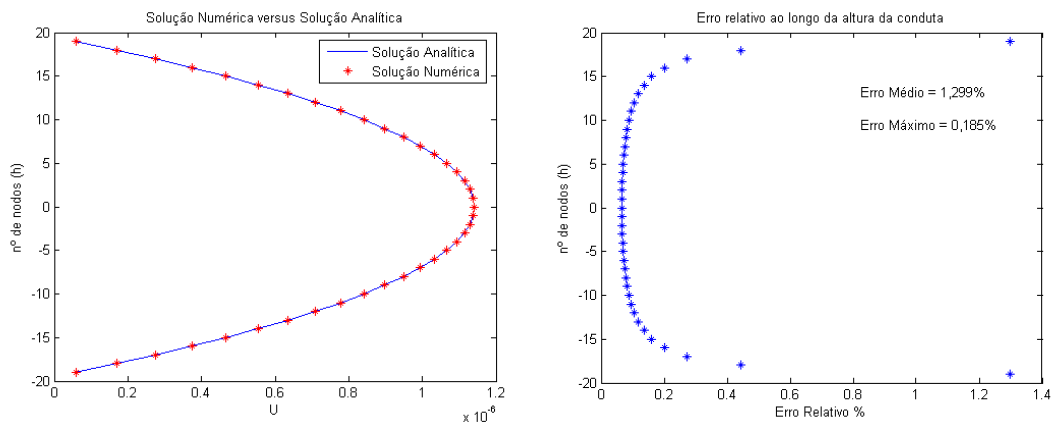


Figura 4.3 - Solução numérica e solução analítica do perfil de velocidades do escoamento de *Poiseuille* para simulação de 41 nós ao longo da altura da conduta.

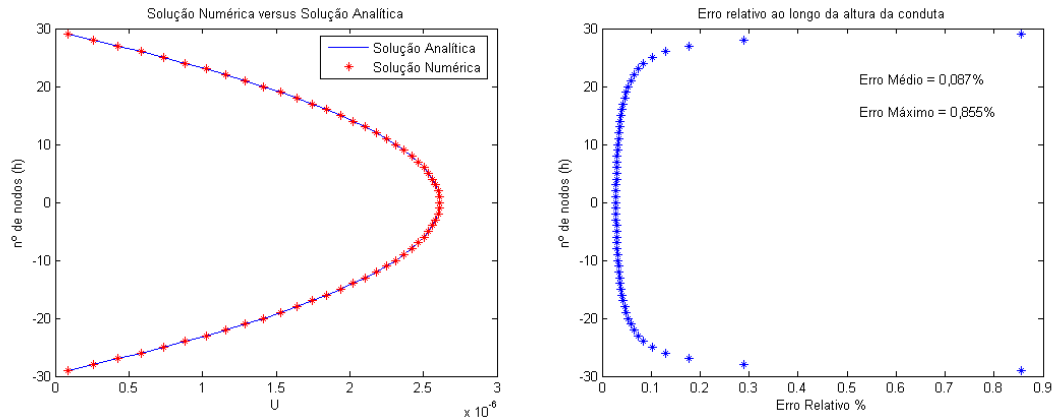


Figura 4.4 - Solução numérica e solução analítica do perfil de velocidades do escoamento de *Poiseuille* para simulação de 61 nós ao longo da altura da conduta.

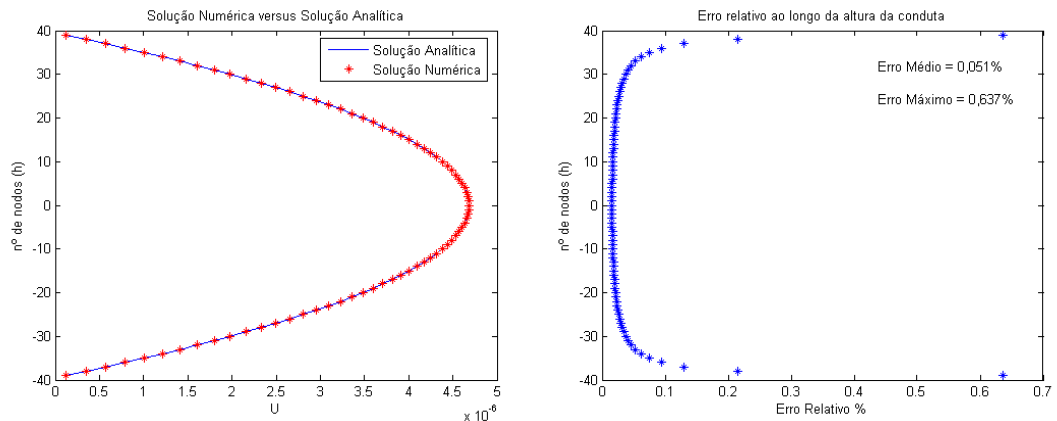


Figura 4.5 - Solução numérica e solução analítica do perfil de velocidades do escoamento de *Poiseuille* para simulação de 81 nós ao longo da altura da conduta.

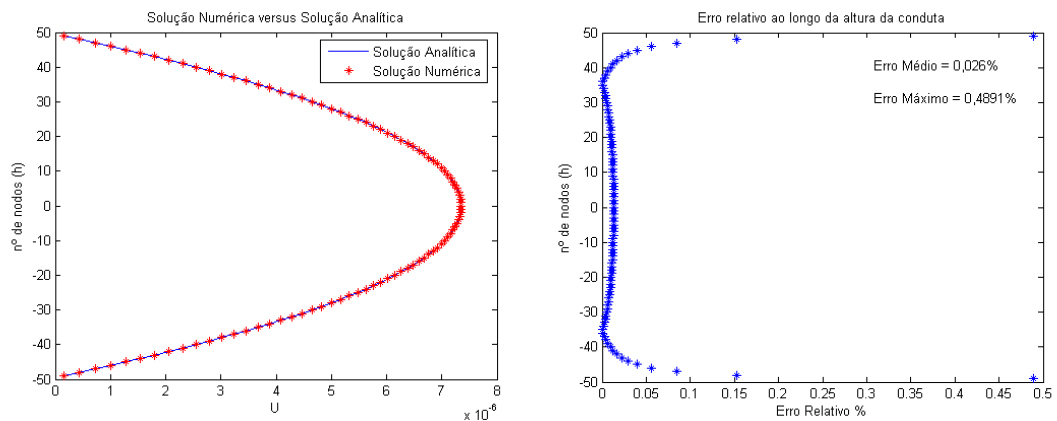


Figura 4.6 - Solução numérica e solução analítica do perfil de velocidades do escoamento de *Poiseuille* para simulação de 101 nós ao longo da altura da conduta.

Como anteriormente indicado, para cada uma das simulações realizadas, foram calculados o erro relativo máximo e o erro relativo médio ao longo do perfil. A Figura 4.7

apresenta o erro relativo máximo da solução numérica em função do número de nós da altura da conduta. Verifica-se assim, que o erro máximo encontra-se nos pontos junto à parede (fundo e no topo). Facto que é explicado pelo facto de que analiticamente a velocidade junto às paredes é nula, sendo que numericamente isso não acontece (nó de fluido imediatamente após a parede possui velocidade não nula). Constata-se assim que o erro é introduzido pela condição fronteira de ressalto, no caso “*half-way bounce-back*” que foi explicitado no Capítulo 2.

Reparando na Figura 4.7, verifica-se que para as simulações realizadas o erro relativo máximo situa-se aproximadamente entre os 0,5% (para 101 nós) e os 2,7 % (para 21 nós). Trata-se de uma gama de valores de erro relativo relativamente baixa, concordante com outros testes numéricos realizados previamente realizados [6, 46]. Da Figura 4.7 observa-se ainda que à medida que o número de nós ao longo da altura da conduta é aumentado regista-se a diminuição do erro relativo máximo.

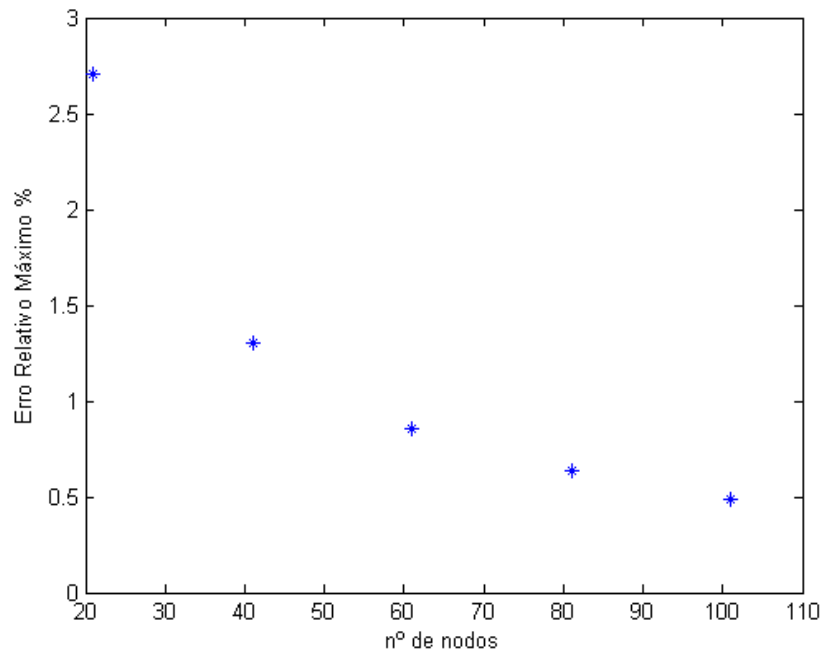


Figura 4.7 - Erro relativo máximo das simulações realizadas em função do número de nós presentes na altura da conduta.

Por sua vez, a Figura 4.8 apresenta o erro médio da solução numérica em função dos nós ao longo da altura da conduta. Constata-se que o erro relativo médio da simulação numérica diminui à medida que se aumenta o número de nós, sendo cerca de 0,7% para o caso de 21 nós e aproximadamente 0,03% para 101 nós.

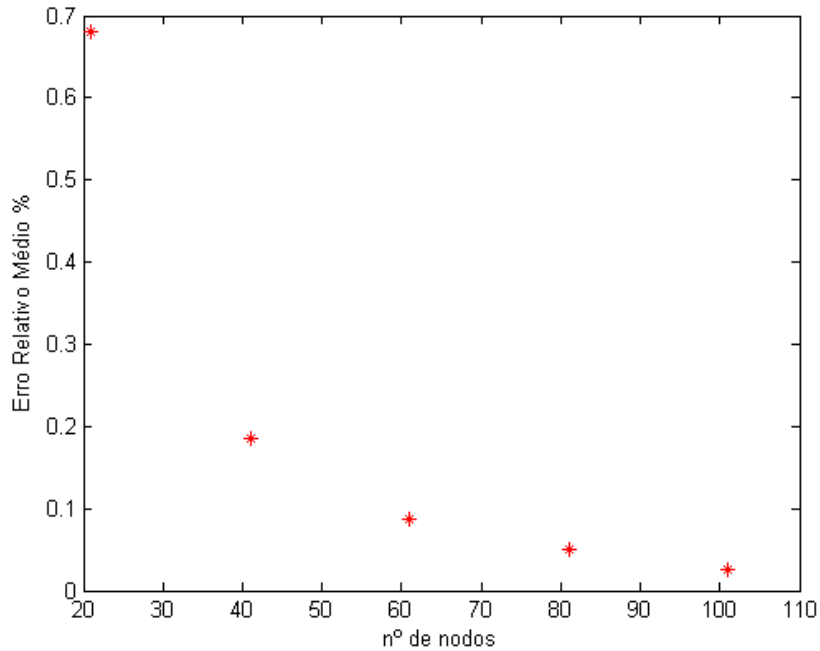


Figura 4.8 - Erro relativo médio das simulações realizadas em função do número de nós presentes na altura da conduta.

4.2. Verificação da Continuidade Usando o LBM

Realizou-se ainda um teste de conservação mássica, a fim de verificar o princípio da continuidade. Como já mencionado na presente dissertação, para escoamentos incompressíveis 3-D as equações de Navier-Stokes têm a si associada a equação da continuidade (eq. 2.16). Contrariamente, na formulação do LBM, o princípio da continuidade não aparece explicitado, sendo apenas recuperado pela referida expansão multi-escala de Chapman-Enskog. Nesse sentido o estudo da presente secção visa verificar até que ponto o simulador LBM do presente trabalho satisfaz este princípio.

Para tal foi criada uma geometria com uma zona de estrangulamento de modo a induzir um aumento de velocidade no escoamento.

Para o cálculo do caudal mássico nas condutas temos que:

$$\dot{m} = \rho \left[\Delta x \left(\sum_{i=1}^{n-1} v_i \right) + \left(\frac{3}{4} \Delta x \right) (v_0 + v_n) \right] \quad (2.2)$$

Em que \dot{m} é o caudal mássico na secção da conduta e ρ a massa volúmica, as variáveis da equação distância *lattice* entre nós Δx e v velocidade nodal, são mais facilmente explicadas pela Figura 5.25, em que ilustra uma coluna de nós (vectores de velocidade):

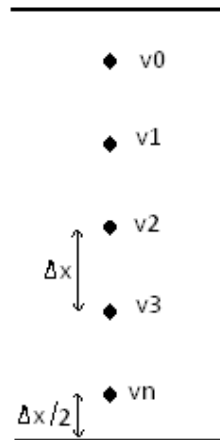


Figura 4.9 - Representação esquemática de uma coluna de vetores de velocidade das simulações.

Executaram-se 4 simulações distintas nas quais foi fixado o número de nós na direcção longitudinal e variado o número de nós na direcção transversal para cada. Assim analisaram-se escoamentos com 100x20 nós (Figura 4.10), 100x30 (Figura 4.11), 100x40 (Figura 4.12), 100x50 (Figura 4.13).

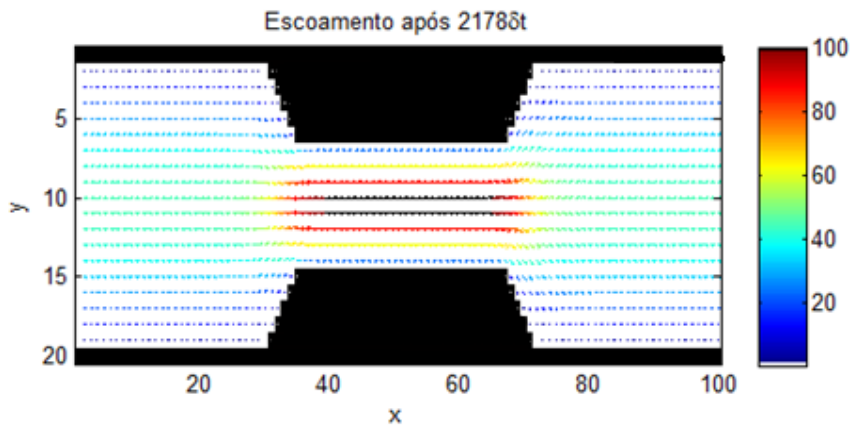


Figura 4.10 - Canal com contração e expansão de 100x20 nós.

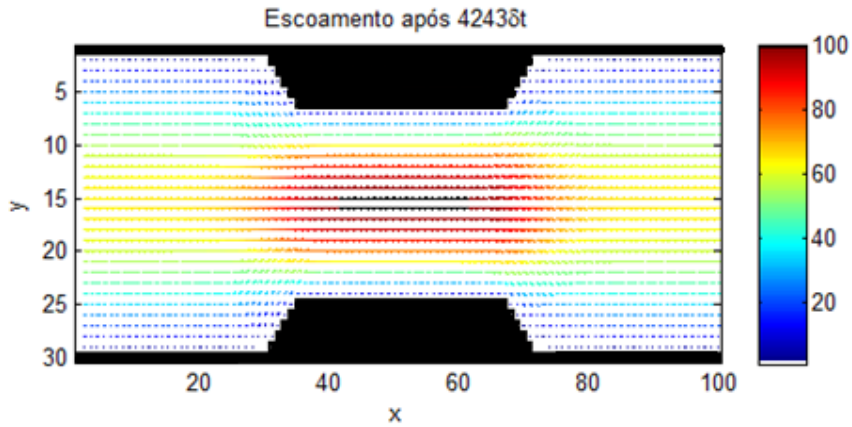


Figura 4.11 - Canal com contracção e expansão de 100x30 nós.

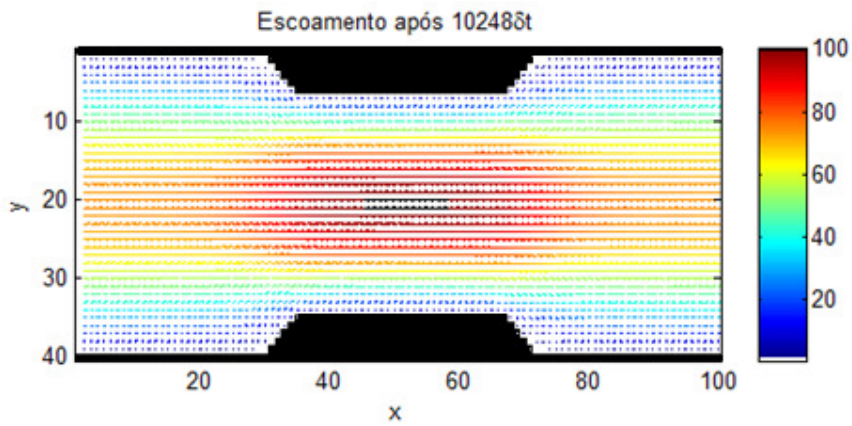


Figura 4.12 - Canal com contracção e expansão de 100x40 nós.

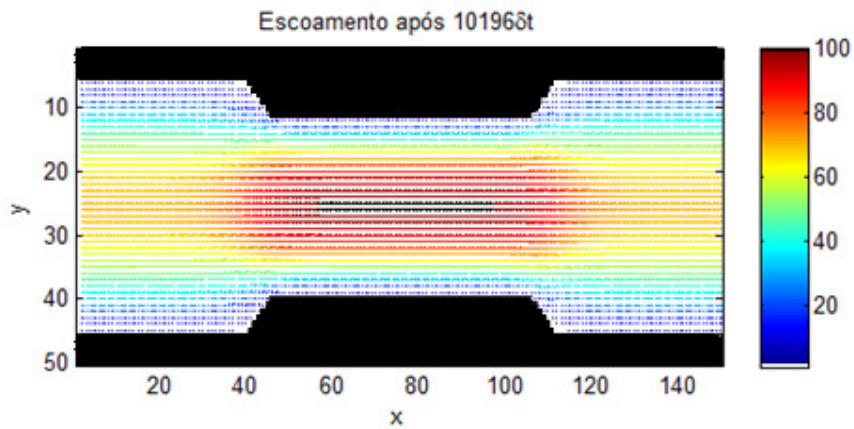


Figura 4.13 - Canal com contracção e expansão de 100x50 nós.

Para cada simulação calcularam-se os respectivos caudais perto da entrada (2ª coluna de nós, já que a primeira está afectada pelo aumento de pressão que induz movimento ao fluido “deltaU”), no meio (zona de estrangulamento) e à saída. Os resultados obtidos

encontram-se apresentados na Tabela 4.1 e o código gerador das soluções é apresentado no Anexo II.

Tabela 4.1 - Caudais mássicos presentes no domínio ao longo das condutas.

Geometria	Caudal mássico (unidades <i>lattice</i> ¹)			Erro relativo Max. %
	Zona de Entrada	Zona de Estrangulamento	Zona de Saída	
100x20 nós	$6,052 \times 10^{-7}$	$5,945 \times 10^{-7}$	$6,053 \times 10^{-7}$	1,780
100x30 nós	$5,177 \times 10^{-6}$	$5,164 \times 10^{-6}$	$5,176 \times 10^{-6}$	0,250
100x40 nós	$1,672 \times 10^{-5}$	$1,671 \times 10^{-5}$	$1,672 \times 10^{-5}$	0,085
100x50 nós	$3,813 \times 10^{-5}$	$3,812 \times 10^{-5}$	$3,813 \times 10^{-5}$	0,038

A Figura 4.14 apresenta o erro relativo em função do nº de nós ao longo da secção transversal do domínio. Constatou-se que à medida que o domínio possui maior número de nós ao longo da secção transversal o erro relativo máximo em termos de diferença de caudais diminui. Atendendo ao facto do LBM não explicitar a equação da continuidade na sua formulação, estes resultados indicam que o LBM respeita o princípio da continuidade, visto que com domínios com mais de 20 nós na secção transversal os erros relativos entre caudais mantêm-se abaixo de 1%.

¹As unidades *lattice*, são usadas nas simulações LBM. Para converter em unidades físicas é necessária a incorporação de alguns factores de conversão. Por exemplo em termos de distância as unidades são convertidas em função do número de nós *lattice* e dimensões do domínio; em termos de velocidade, esta é convertida tendo por base a velocidade do som. Posteriormente será exibida uma Tabela (Tabela 5.2) com alguns factores de conversão.

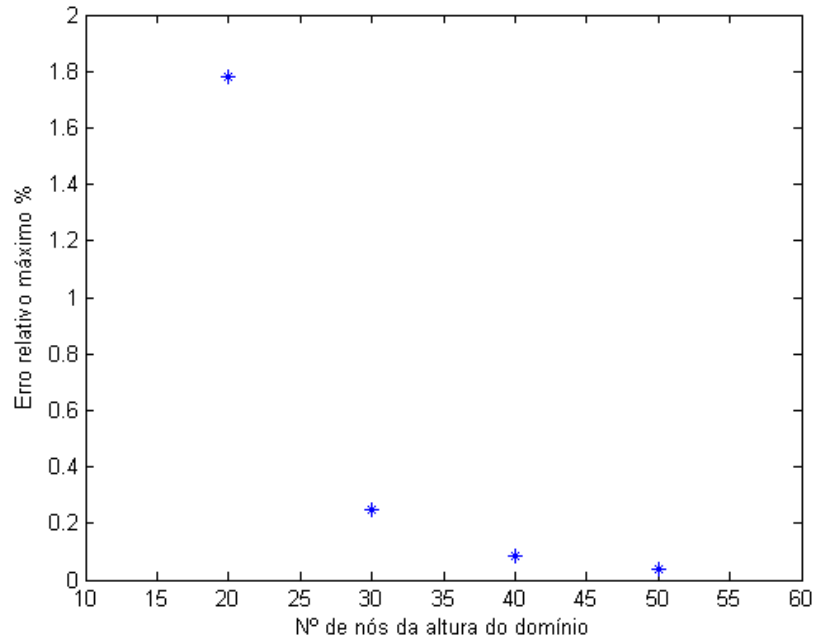


Figura 4.14 – Erro relativo máximo entre diferenças de caudais em domínios com diferentes números de nós ao longo da secção transversal ao escoamento.

5. TRATAMENTO DOS RESULTADOS

O presente capítulo apresenta a metodologia efectuada tendo em vista a obtenção de simulações de escoamentos em geometrias variadas, objectivo principal do presente projecto.

Para além do aspecto indicado, outros ajustes e melhorias foram implementados nos códigos, tendo sido concebidas linhas de código que permitem melhorias no modo de visualização dos escoamentos, especialmente no código 3D (Algoritmo 3.2.) em que apenas se visualizam cortes de secções das geometrias.

5.1. Integração de Novas Geometrias

Como referido no Capítulo 3 a matriz booleana “BOUND” é responsável pela definição do cariz dos nós do domínio: elementos de valor “1” são nós sólidos e elementos de valor “0” são nós ocupados por partículas de fluido. No caso do modelo D2Q9 a matriz é 2D, enquanto para o modelo D3Q19 é 3D.

Para mudar a geometria é necessário proceder à alteração a matriz “BOUND”. Nesse sentido, inicialmente procedeu-se à captação de metodologias capazes de criar o tipo de matrizes pretendido para o código 2D. O primeiro procedimento experimentado consistiu na criação de matrizes por via de um processo de concatenação, isto é, para determinado tipo de geometria pretendido, procedeu-se à definição dos “0’s” e “1’s” correspondentes aos elementos da matriz em questão e manualmente, ou por funções MATLAB geradoras de matrizes de “0’s” ou “1’s” iam-se juntando as matrizes de forma a construir a matriz final da geometria pretendida para proceder à simulação do escoamento. Concluiu-se tratar de um processo moroso, cuja aplicação na criação de geometrias mais complexas torna o processo inviável.

Um segundo método testado consistiu na criação das matrizes da geometria em Excel, importando-se de seguida o ficheiro para o MATLAB de modo a integrar a dita matriz no código. Apesar de mais rápido que o procedimento anterior, este mostrou-se pouco flexível em termos de elaboração de matrizes complexas.

Atendendo às principais limitações dos procedimentos enunciados, procuraram-se métodos mais flexíveis e eficazes para a geração das geometrias. Assim, depois da consulta de alguns conceitos [72], constatou-se que o MATLAB aceita diversos formatos de arquivo de imagem comerciais por meio da função “imread”. No MATLAB as imagens consistem em matrizes de dados, normalmente com uma matriz de mapa de cores associada. Assim,

depois de incorporadas algumas linhas de código no programa referentes à leitura de imagens e adaptação das mesmas ao domínio, viabilizou-se a importação de qualquer tipo de imagem em formato: 'jpg', 'tif', 'gif', 'bmp', 'png', 'hdf', 'pcx', 'xwd', 'cur' e 'ico' prontas a correr no código 2D. As Figuras 5.1, 5.2, 5.3 e 5.4 exibem algumas das imagens testadas. Todas possuem o mesmo modo de apresentação de resultados que foi enunciado nos resultados do capítulo anterior (Figura 3.1), em que os vectores de velocidade são apresentados a azul; com a sua dimensão e direcção a estarem associadas ao comportamento do fluido em cada nó. Nota também para a presença dos instantes de tempo apresentados no título da figura e para os eixos x e y que indicam o número de nós considerados em cada domínio.

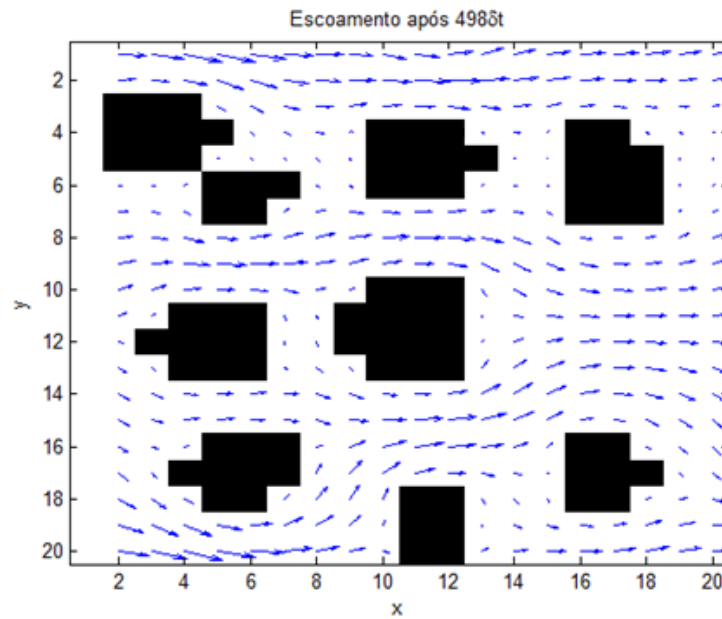


Figura 5.1 - Simulação a um escoamento em torno da "geometria 1" com resolução de 20x20 píxeis/nós.

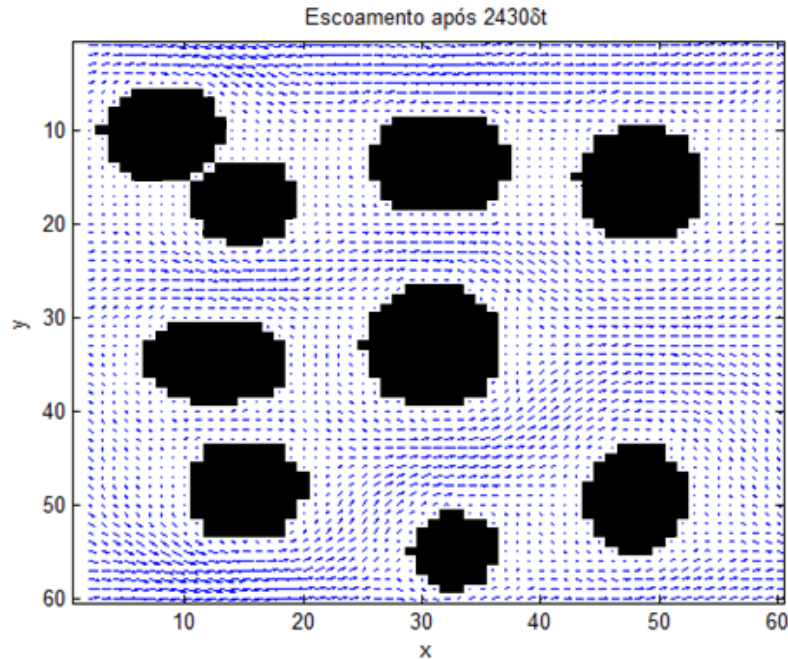


Figura 5.2 - Simulação a um escoamento em torno da “geometria 1” com resolução de 60x60 pixéis/nós.

Como se pode constatar ambas as figuras representam simulações da mesma geometria, “geometria 1” no caso. Verifica-se que na Figura 5.1 os vectores de velocidade são mais perceptíveis, devido ao facto da menor definição do domínio (20x20). Contudo a aproximação que é feita aos nós sólidos da geometria é muito grosseira quando comparada com a aproximação evidenciada na Figura 5.2 em que o detalhe da geometria é maior devido à também maior definição do domínio (60x60). No entanto os vectores velocidade apresentam-se mais pequenos, em muitos casos imperceptíveis. Este problema de visualização pode ser contornado utilizando a ferramenta “zoom” pertencente à barra de ferramentas de imagens do MATLAB ou mediante um outro procedimento a indicar posteriormente. A partir da ferramenta “zoom” é possível destacar figuras com pormenores de interesse do domínio, como é demonstrado na Figura 5.3, em que se seleccionou uma zona intermédia do escoamento para visionar com maior detalhe os vectores de velocidade. De referir que o código gerador das simulações à “geometria 1” se encontra apresentado no Anexo III.

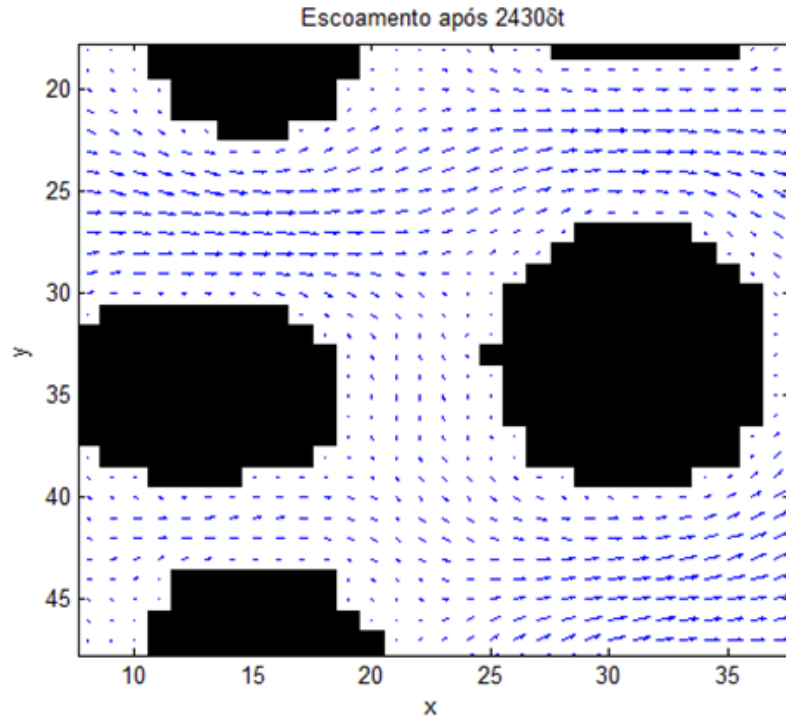


Figura 5.3 - Aproximação a pormenor da simulação de escoamento em torno da "geometria 1" com resolução de 60x60 pixels/nós.

A Figura 5.4 apresenta uma outra geometria ("geometria 2") testada, enquanto a Figura 5.5 apresenta um detalhe do escoamento simulado nesse teste. A mensagem que se pretende passar com as mesmas é a de possibilidade de verificação pormenorizada de pequenos detalhes associados aos escoamentos mediante a utilização da ferramenta "zoom".

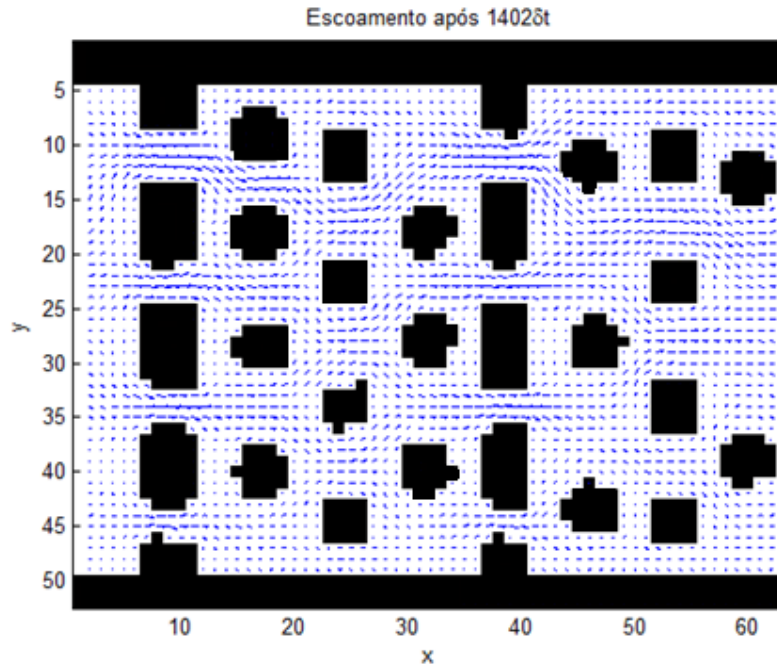


Figura 5.4 - Simulação a um escoamento em torno da “geometria 2” com resolução de 63x52 pixéis/nós.

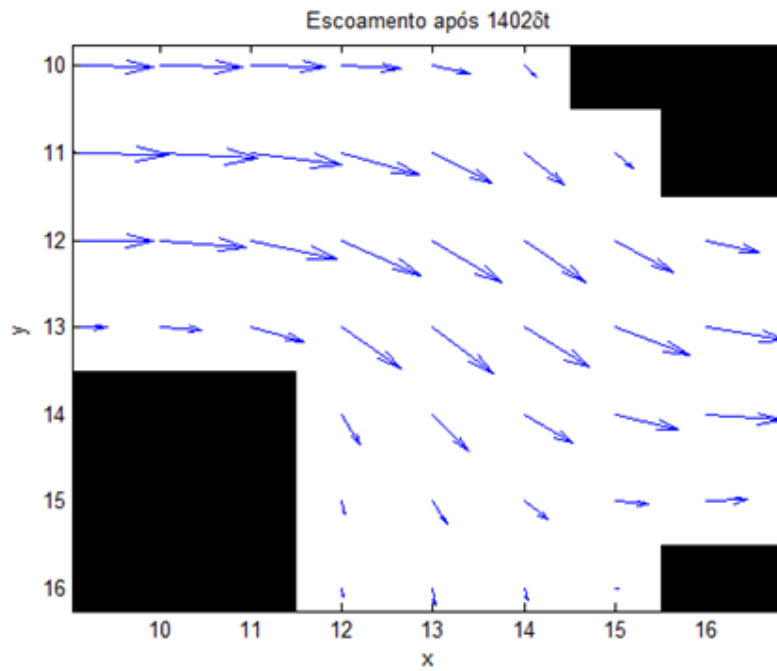


Figura 5.5 - Aproximação a pormenor da simulação de escoamento em torno da “geometria 2” com resolução de 63x52 pixéis/nós.

As Figuras apresentadas demonstram as seguintes capacidades adquiridas pelo código:

- Integração de qualquer geometria 2D de ficheiros de imagem no Algoritmo 3.1;

- Flexibilidade em termos de definição de imagem, ou seja, hipótese de redimensionar domínios de nós/pixéis consoante as necessidades da geometria/simulação, diferença exposta entre a Figura 5.1 e 5.2;
- Capacidade de observar pormenores do escoamento em diferentes partes do domínio mediante a utilização da ferramenta “zoom”, casos da Figura 5.3 e 5.5;

5.2. Visualização dos Vectores de Velocidade

Um outro ponto que suscitou interesse aquando da análise dos códigos teve a ver com o modo de visualização dos resultados, nomeadamente em termos dos vectores de velocidade. Nesse sentido pensou-se criar um meio que permitisse dar uma melhor noção da velocidade do escoamento ao longo do domínio.

O procedimento executado resultou numa adequação da tonalidade de cor dos vectores de velocidade de acordo com a dimensão dos mesmos, ficando desse modo os vectores de maior grandeza mais escuros (cores mais quentes) e os de menor dimensão a cores mais “frias”. Tratou-se de um processo algo complicado, já que a função “quiver”, responsável pela criação dos vectores de velocidade, apenas permite que uma cor se associe aos respectivos vectores. Assim, mediante a procura, na página da Internet da *MathWorks* “Matlab Central” [73], de programas relacionados com a função “quiver”, conseguiu-se obter uma função “quiverc” que corresponde uma versão modificada da função “quiver” para criar vectores de velocidade com as componentes (u,v) nos pontos (x,y) com um mapa de cores associado à grandeza do vector em causa. Contudo a dita função não possuía um mapa de cores adequada ao efeito pretendido. Mapas de cores, no MATLAB, são matrizes com três colunas que representam valores de cores, estando associado a cada linha o que se designa por valores RGB (*red-green-blue*): a intensidade das componentes vermelha, verde e azul de uma dada cor. A Tabela 5.1 ilustra a correspondência entre os valores numéricos num mapa de cores e as cores.

Tabela 5.1 - Correspondência entre valores numéricos de RGB num mapa de cores.

<u>Vermelho</u>	<u>Verde</u>	<u>Azul</u>	<u>Cor</u>
1	0	0	Vermelho
0	1	0	Verde
0	0	1	Azul
1	1	0	Amarelo
1	0	1	Magenta
0	1	1	Ciano
0	0	0	Preto
1	1	1	Branco
0,5	0,5	0,5	Cinzentos
0,67	0	1	Violeta
0,5	0,4	0	Laranja

Devido ao problema relacionado com o mapa de cores, foi necessária a execução de alguns ajustes à função “quiverc”, procedendo-se nomeadamente, à construção de uma nova matriz de cor e incorporando-a na dita função. Os resultados obtidos em termos de visualização das simulações são apresentados nas Figuras 5.6, 5.7, 5.8, 5.9 e 5.10. Todas as figuras apresentam o escoamento através de uma mesma geometria (“geometria 3” no caso), com definições de 20x20, 40x40, 60x60, 120x120 e 200x200 nós, respectivamente. A barra vertical colocada do lado direito do escoamento, apresenta o mapa de cores utilizado, ilustrando os níveis de cores relacionados com a velocidade dos vectores. Assim, para o nível de cor “100” está associado aos vectores de velocidade máxima enquanto o “0” está associado aos vectores de velocidade mais reduzida.

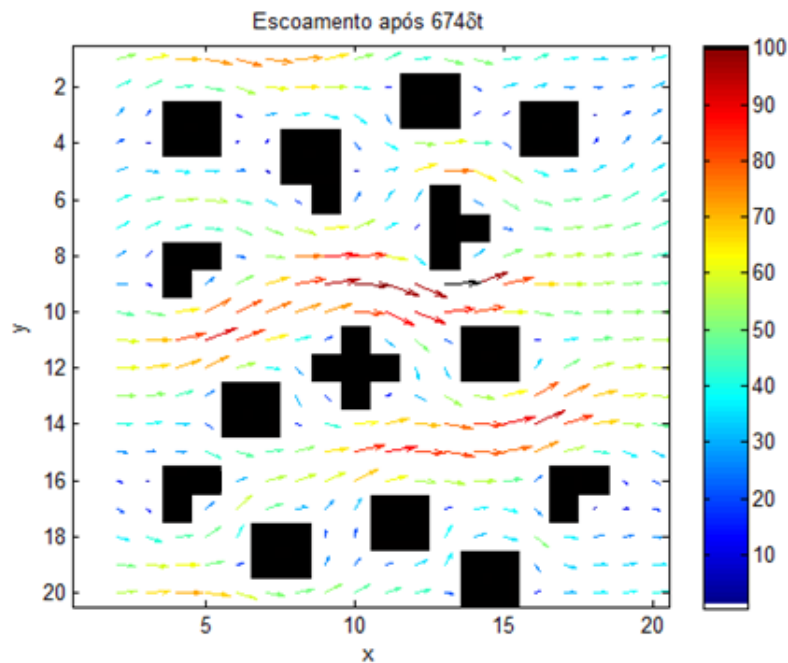


Figura 5.6 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 20x20 pixels/nós.

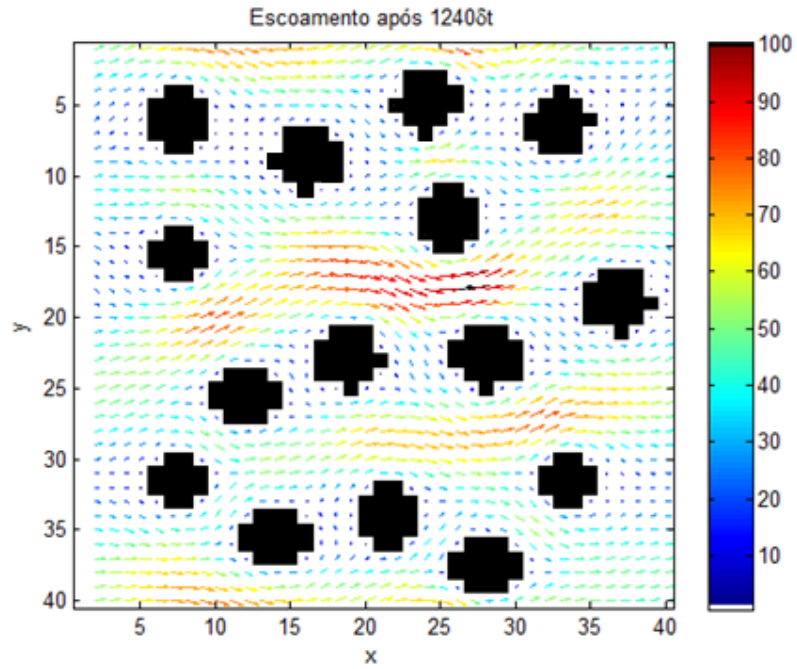


Figura 5.7 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 40x40 pixels/nós.

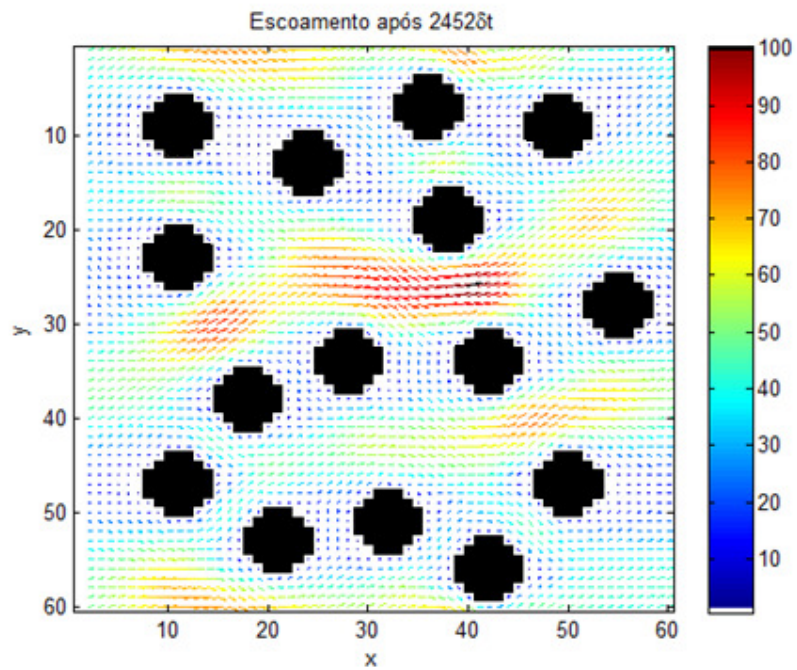


Figura 5.8 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 60x60 pixels/nós.

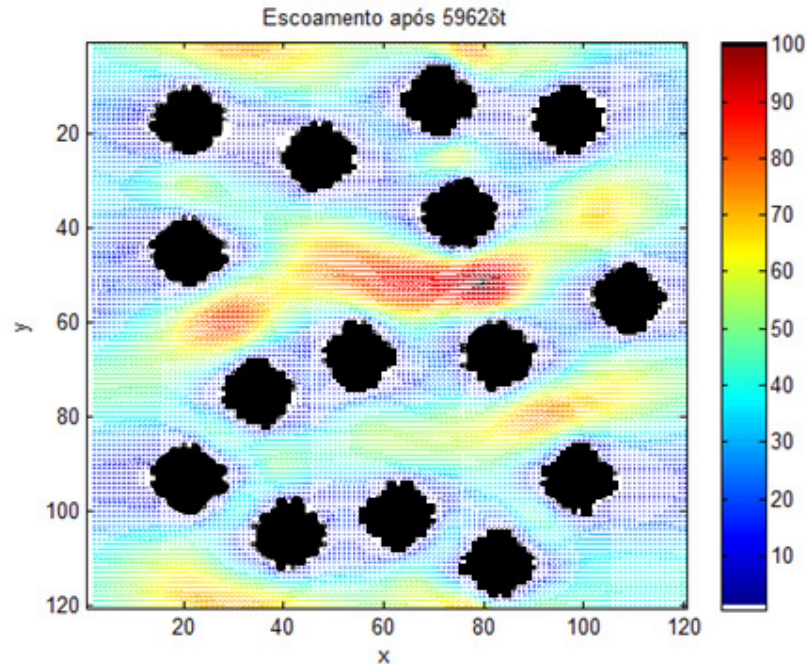


Figura 5.9 - Simulação do escoamento através da “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 120x120 pixéis/nós.

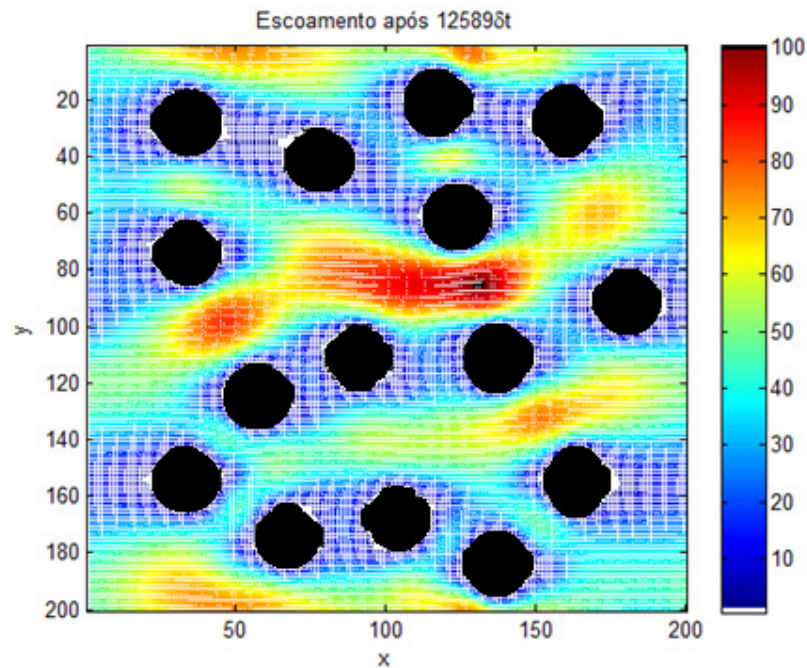


Figura 5.10 - Simulação do escoamento por entre a “geometria 3”, com alteração nos vectores de velocidade com uma resolução de 200x200 pixéis/nós.

Para este tipo de simulação encontrou-se ainda um meio capaz de apresentar os resultados de maneira a que a geometria a simular possua elevada definição, tendo em

simultâneo os vectores de velocidade perceptíveis sem que seja necessária a intervenção da ferramenta “zoom”. Para tal, bastou se seleccionarem valores das matrizes de velocidade (uma segundo X e outra segundo Y) com determinado passo, retirando, por exemplo, valores de 5 em 5 nas linhas e colunas. O resultado do procedimento enunciado pode ser visto na simulação de 120x120 nós/pixéis da “geometria 3” apresentada na Figura 5.11, assim como na simulação de 200x200 nós/pixéis da mesma geometria na Figura 5.12.

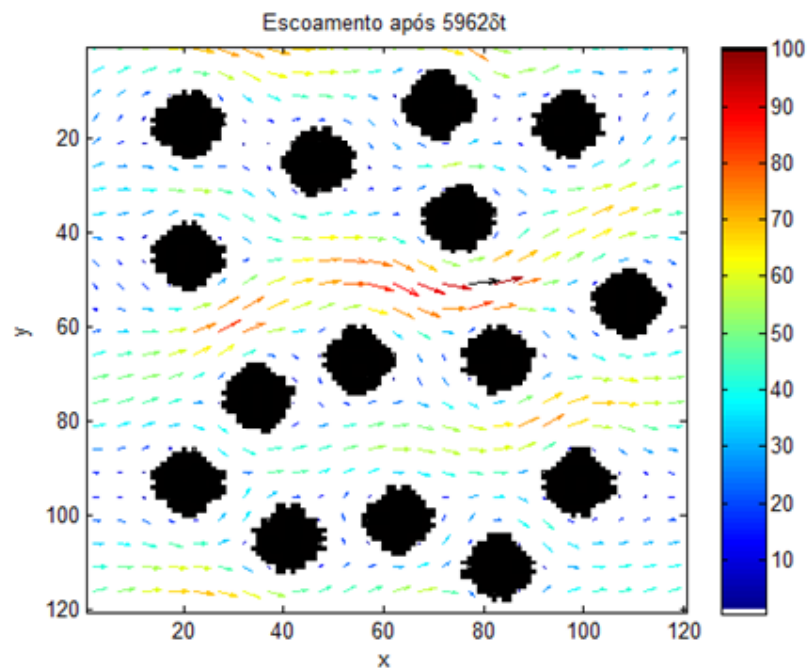


Figura 5.11 - Simulação do escoamento através da “geometria 3”, com diferente definição nos vectores de velocidade (120x120 pixéis/nós).

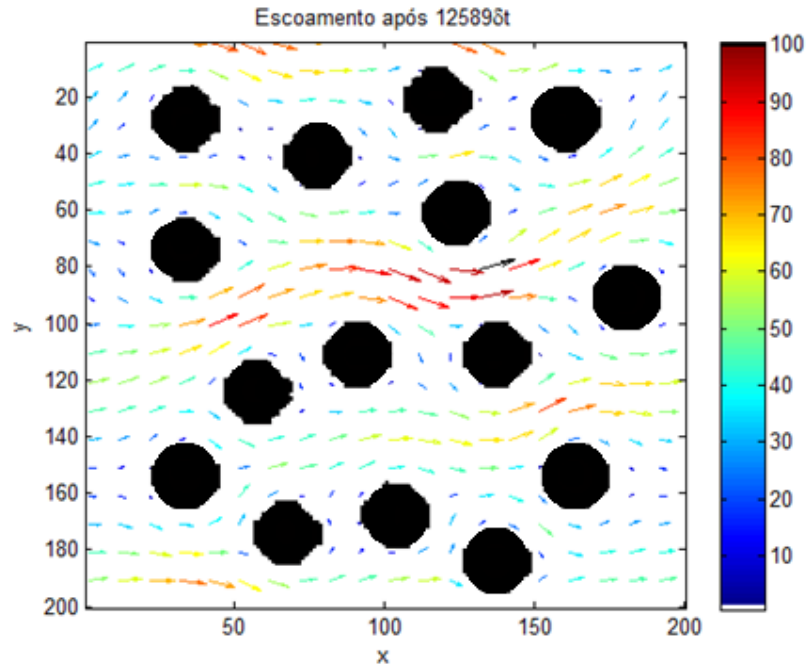


Figura 5.12 - Simulação do escoamento através da “geometria 3”, com diferente definição nos vectores de velocidade (200x200 pixéis/nós).

5.3. Visualização 3D

Outro aspecto que suscitou interesse teve a ver com o facto do código D3Q19 (Algoritmo 3.2) apenas apresentar duas secções do escoamento simulado, sendo deste modo pouco ilustrativo da geometria em causa.

Na tentativa de tornar os resultados mais compreensíveis conceberam-se algumas linhas de código capazes de gerar geometrias 3D de modo a permitir uma melhor compreensão da configuração (tridimensional) das diferentes simulações.

A geração de geometrias a integrar no código 3D também foi um dos aspectos tratados. Ao contrário do que ocorre com os ficheiros de imagem, o MATLAB não aceita ficheiros CAD de objectos tridimensionais. Contudo duas estratégias foram elaboradas com algum sucesso:

- a inserção de geometrias sólidas mediante equações;
- utilização da função do MATLAB “repmat” para prolongar na terceira dimensão matrizes de imagens importadas;

A primeira estratégia consiste na definição de ciclos “for” para determinadas partes do domínio. Esses ciclos restringem os nós nos quais equações geométricas atribuirão valor “1” aos nós sólidos. A equação 5.1 referente a superfícies esféricas e a equação 5.2 para superfícies cilíndricas são exemplos de equações que foram testadas no código, x, y e z são

as coordenadas cartesianas dos eixos de referência e a , b e c são pontos pertencentes ao mesmo referencial.

$$(x - a)^2 + (y - b)^2 + (z - c)^2 = r^2 \quad (5.1)$$

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \quad (5.2)$$

Na Figura 5.13 é apresentado o escoamento tridimensional em torno de uma esfera num domínio de $12 \times 12 \times 12$ nós, mediante o processo de geração mencionado. Já a Figura 5.14, apresenta os cortes em $y = 9$ e $z = 9$ da simulação em causa.

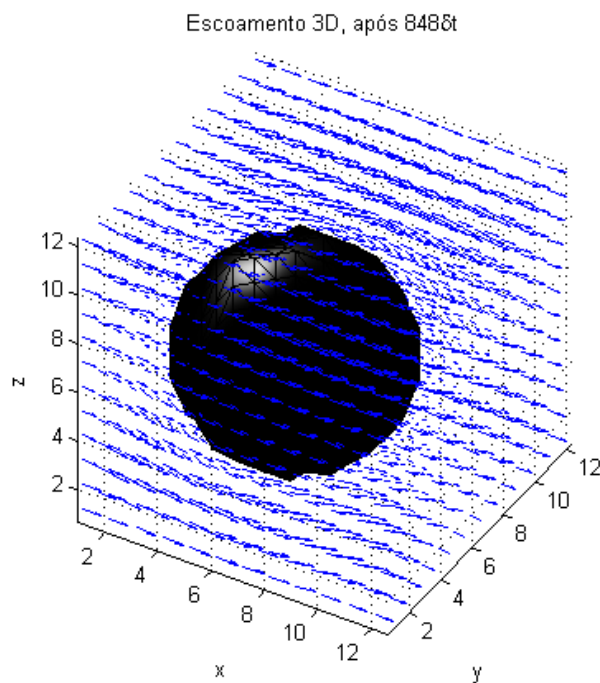


Figura 5.13 - Simulação de escoamento tridimensional em torno de uma esfera, com definição de $12 \times 12 \times 12$ nós.

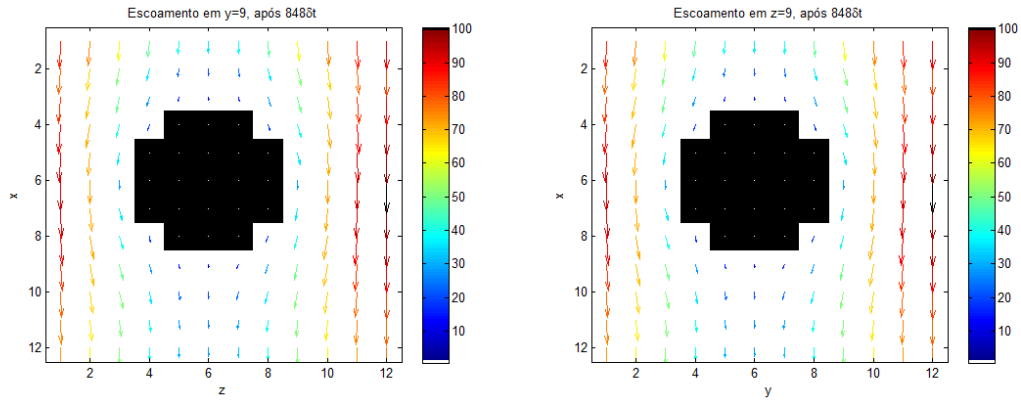


Figura 5.14 - Simulação de escoamento tridimensional em torno de uma esfera, com definição de 12x12x12 nós; cortes em $y = 9$ e $z = 9$

Mediante o mesmo processo, foi também construída uma geometria composta por 5 esferas num domínio de 12x12x12 nós, encontrando-se a respectiva simulação 3D representada na Figura 5.15. Na Figura 5.16 são apresentados cortes do mesmo escoamento em $z = 9$ e $y = 6$.

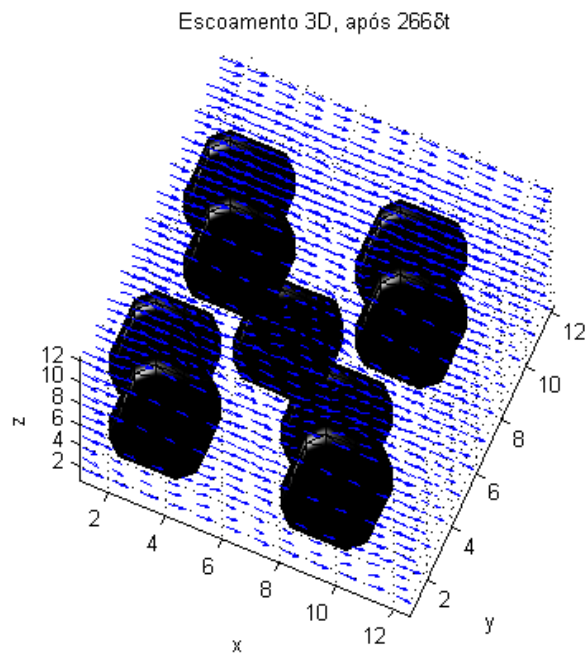


Figura 5.15 - Escoamento tridimensional através de uma geometria de composta por 5 objectos esféricos, com um domínio de 12x12x12 nós.

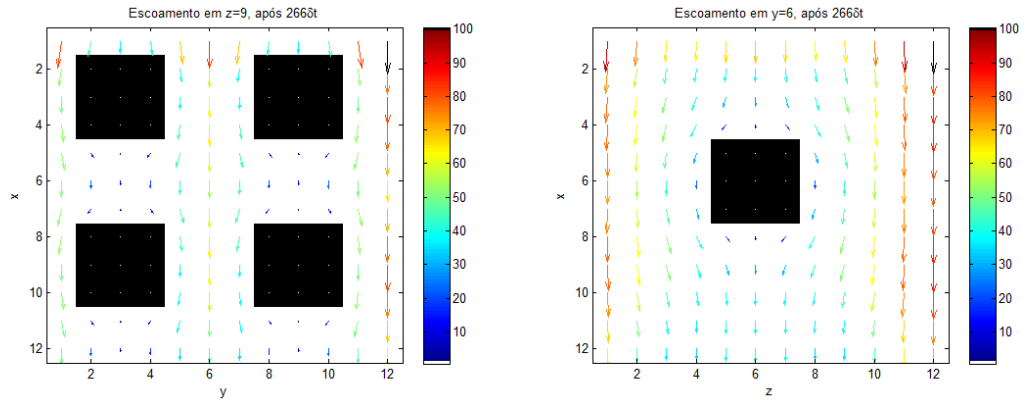


Figura 5.16 - Simulação do escoamento tridimensional através de uma geometria composta por 5 objectos esféricos; cortes em $z = 9$ e $y = 6$.

Por outro lado, a segunda estratégia adoptada para gerar geometrias 3D consistiu em se prolongar na terceira dimensão matrizes de imagem mediante a função do MATLAB “repmat”. Para melhor explicar o procedimento adoptado, a Figura 5.17 apresenta uma imagem (Imagem 1) a partir da qual se obteve a geometria tridimensional presente na Figura 5.18. É de notar que este processo pode ser executado como mais que uma imagem, resultando a geometria final num conjunto de “imagens que foram repetidas na terceira dimensão”.

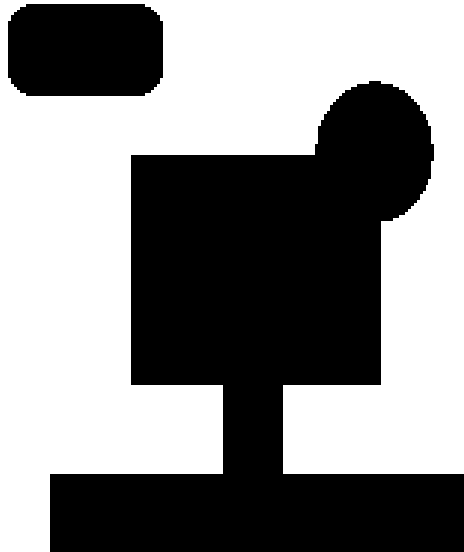


Figura 5.17 - Imagem ilustrativa do procedimento de geração de geometrias 3D.

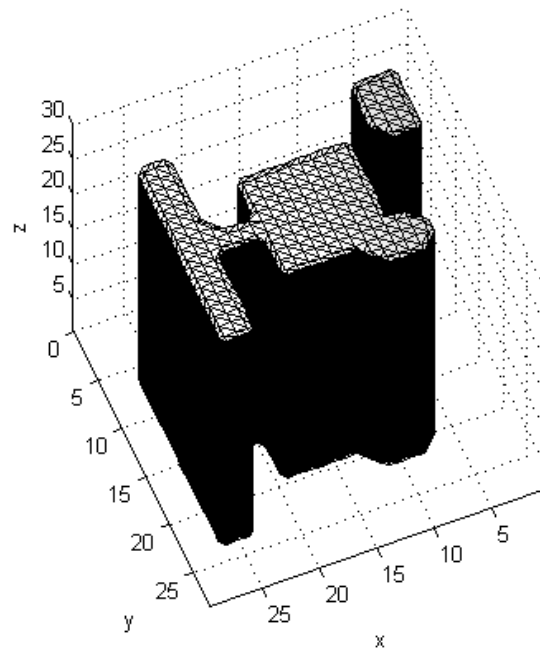


Figura 5.18 - Geometria 3D gerada mediante o procedimento de prolongação na terceira dimensão de imagens 2D.

A Figura 5.19 ilustra uma simulação 3D obtida mediante procedimento enunciado para a geometria apresentada na Figura 5.18, num domínio de $30 \times 30 \times 30$ nós. Na Figura 5.20 são apresentados cortes do escoamento em $y = 6$ e $z = 6$.

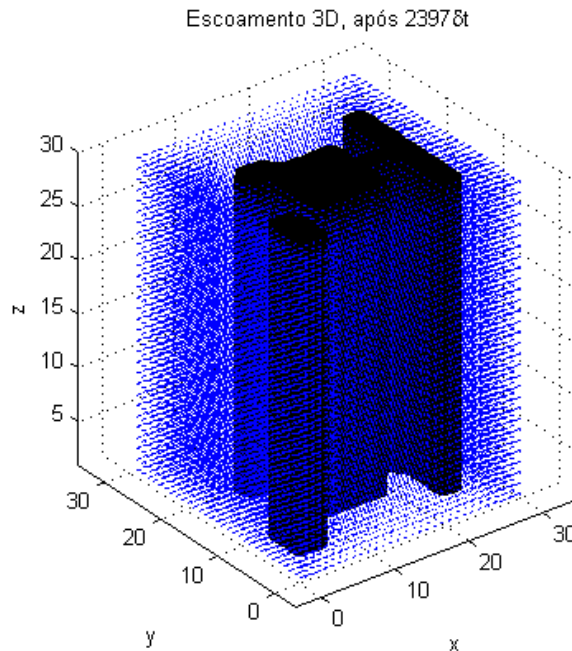


Figura 5.19 - Escoamento em torno da Geometria 3D gerada mediante expansão na terceira dimensão da "Imagem 1".

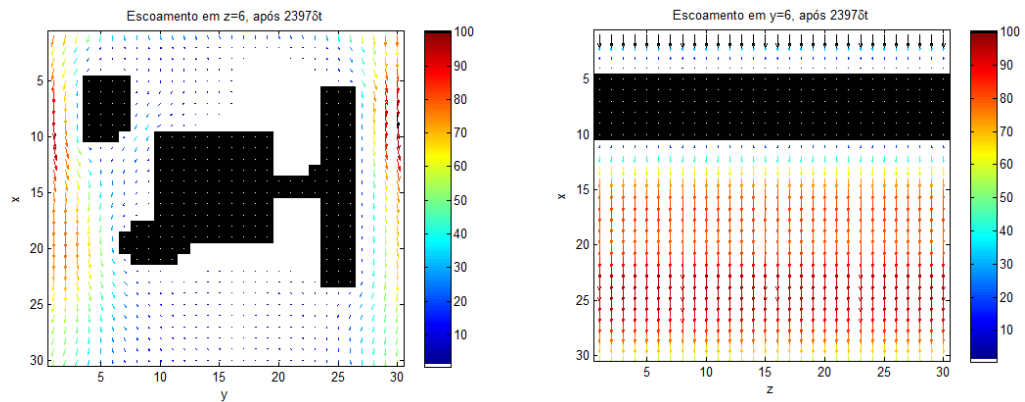


Figura 5.20 - Geometria obtida mediante expansão na terceira dimensão da "Imagem 1"; cortes em z = 6 e y = 6.

Foi ainda experimentado o procedimento enunciado para várias imagens, ou seja tentou-se obter geometrias 3D mediante o aglomerar de várias imagens expandidas na terceira dimensão. Por exemplo, a Figura 5.21 apresenta uma simulação numa geometria obtida através da criação de matrizes aleatórias de "0's" e "1's" com a função do MATLAB "rand" (à semelhança do que acontece no Algoritmo 3.1), tendo sido as mesmas expandidas na terceira dimensão, ao todo foram 4 as geometrias 3D definidas e integradas por intermédio deste procedimento. Por seu turno, a Figura 5.22 apresenta o comportamento do mesmo escoamento nos cortes y = 7 do lado esquerdo e x = 3 no lado direito. De notar que a simulação em causa foi feita num domínio com 14x14x14 nós.

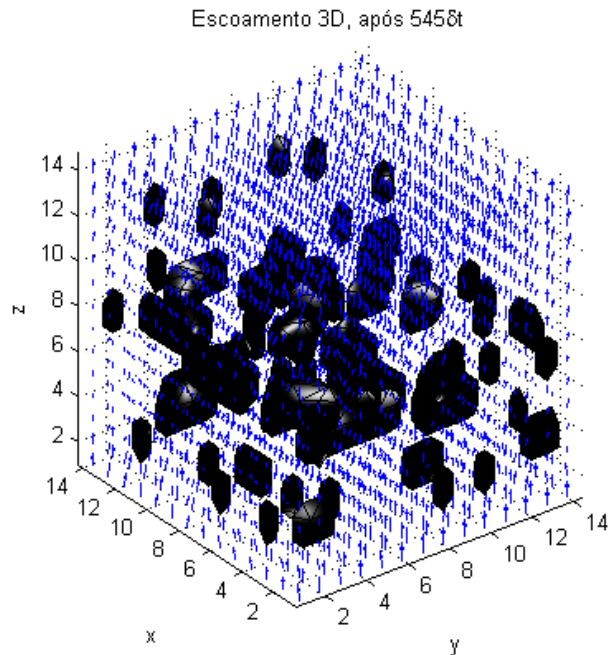


Figura 5.21- Escoamento tridimensional através de uma geometria porosa, com definição de 14x14x14 nós.

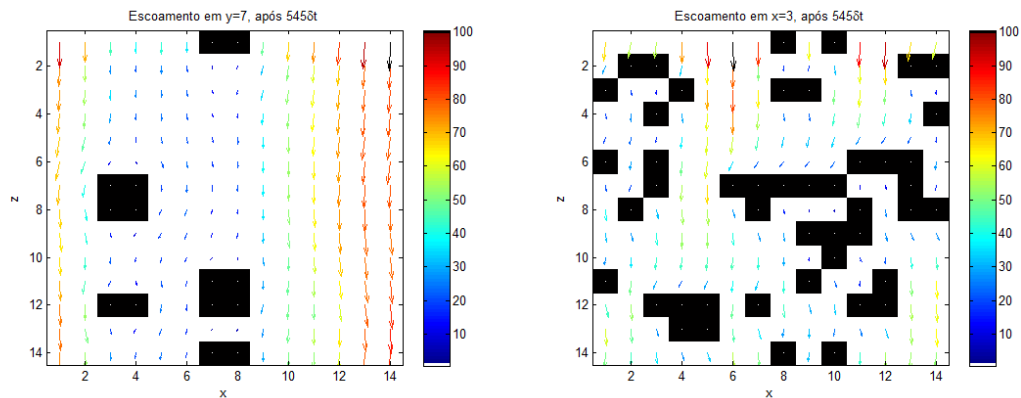


Figura 5.22 - Escoamento tridimensional através de uma geometria porosa, com definição de 14x14x14 nós; cortes em $y = 7$ e $x = 3$.

Por fim, foram ainda concebidas algumas geometrias resultantes da integração dos dois procedimentos enunciados, ou seja mediante a expansão de imagens na terceira dimensão e mediante a definição de equações de superfícies de sólidos. Assim, na Figura 5.23 pode ser vista a simulação tridimensional através de uma geometria composta por seis cilindros intersectados, num domínio com 18x18x18 nós. Esta geometria resultou da implementação no plano xy de seis equações de superfícies cilíndricas e na definição no plano xz da expansão na terceira dimensão de uma imagem constituída por círculos. Na Figura 5.24 observa-se o comportamento do escoamento nos cortes $z = 9$ e $y = 6$ na dita simulação; no corte $z = 9$ verifica-se que o fluido ao passar pela geometria perde velocidade, sendo a velocidade do escoamento entre as partes sólidas baixa, no corte $y = 6$

verifica-se o mesmo mas de forma ainda mais acentuada, quase não havendo circulação de fluido entre os cilindros.

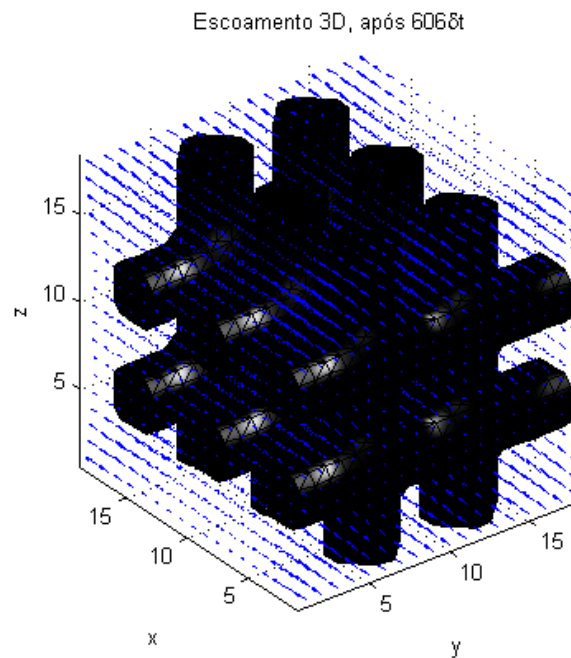


Figura 5.23 - Escoamento tridimensional através de uma geometria de objectos cilíndricos cruzados, com uma definição de 18x18x18 nós.

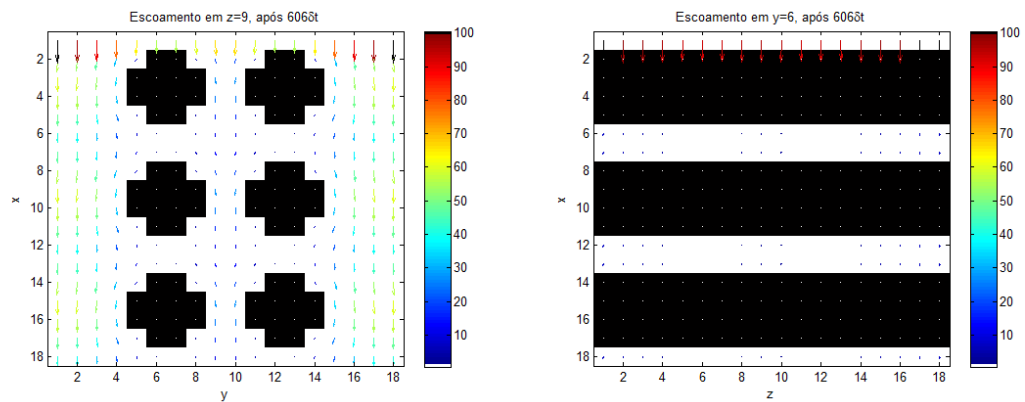


Figura 5.24 - Simulação de escoamento tridimensional através de objectos cilíndricos cruzados, com definição de 18x18x18 nós; cortes em $z = 9$ e $y = 6$.

É de notar que os resultados apresentados incorporam os incrementos feitos no código abordados nos pontos anteriores. Assim, com o trabalho elaborado conseguiram-se alcançar os seguintes resultados:

- Possibilidade de visualizar de forma perceptível geometrias tridimensionais a simular no código LBM;

- Caso se pretenda dar mais atenção a determinado pormenor ou secção do escoamento, facilmente se podem efectuar cortes e aproximações (“zoom”);
- Os cortes efectuados a secções do escoamento encontram-se actualizadas com um mapa de cores que ilustra melhor o comportamento do escoamento;
- Existe ainda a possibilidade de integração de vários tipos de geometria mediante a importação de imagens 2D que podem ser estendidas em mais uma direcção e mediante a formulação de equações geométricas de superfícies.

5.4. Escoamentos em Meios Porosos

A permeabilidade é a medida da capacidade de circulação de um fluido através de dado meio sem alterar a sua estrutura interna. Em muitos casos a determinação da permeabilidade não é de fácil acesso, tornando-se uma tarefa morosa para investigadores, principalmente devido à complexidade e escalas reduzidas dos meios em estudo. Existe assim grande dependência da permeabilidade ao meio em estudo, não existindo, por esse motivo, um modelo geral de permeabilidade em função do tipo de porosidade do meio. Tipicamente, a determinação da permeabilidade requer bastante tempo e trabalho experimental.

Para o cálculo da permeabilidade de determinado meio é assumido um gradiente de pressão constante no meio poroso e uma velocidade média do escoamento, para depois se aplicar a Lei de Darcy [56]:

$$\langle u \rangle = -\frac{K}{\mu} \nabla P \quad (5.3)$$

Em que $\langle u \rangle$ é a velocidade média do escoamento no domínio em causa, K o tensor de permeabilidade, ∇P o gradiente de pressão e μ a viscosidade dinâmica do fluido. De notar que esta relação apenas é válida para números de Reynolds muito baixos ($Re \ll 1$).

Diversos modelos de permeabilidade em meios porosos são baseados em dados experimentais. Contudo, devido à sua natureza macroscópica, este tipo de abordagem não consegue reproduzir com grande acerto o comportamento dos escoamentos em meios porosos (escala mesoscópica). Para além da dificuldade apontada estas abordagens são caras e demoradas.

Outros tipos de abordagem utilizados baseiam-se em métodos analíticos e numéricos que visam prever a permeabilidade de determinado meio calculando ou simulando o escoamento do fluido através dos interstícios do domínio em causa. Este tipo de procedimento é, em traços gerais, caracterizado pelas seguintes etapas [56]:

1. Definição do gradiente de pressão presente no meio;
2. Cálculo/Simulação do escoamento através do meio poroso;

3. Determinação da permeabilidade do meio usando a Lei de Darcy;

Relativamente a abordagens experimentais, procedimentos de base numérica, desde que sejam devidamente validadas, apresentam as seguintes vantagens [56]:

- Geometria do meio pode ser facilmente alterada, já que pode ser digitalizada ou construída virtualmente;
- Maior rapidez na obtenção de resultados;
- Maior capacidade/flexibilidade para explorar variáveis de interesse;

Um desafio importante presente na simulação de escoamentos em meios porosos prende-se com a representação correcta da estrutura porosa de dado meio real, já que esta condição afecta todas as etapas de estudo posteriores. Técnicas de imagem como por exemplo Tomografia Computorizada, têm sido desenvolvidas no sentido de capturar meios porosos reais para dessa forma se obterem geometrias virtuais fidedignas; contudo esses procedimentos podem-se tornar caros, demorados e não aplicáveis a escalas mais reduzidas. Em alternativa, a forma mais utilizada para reproduzir meios porosos para efeito de simulação numérica passa pela reconstrução de estruturas virtualmente, recorrendo a *softwares* e códigos computacionais para esse âmbito. Este tipo de procedimento encontra-se dependente da complexidade da estrutura, dos recursos computacionais disponíveis, bem como da natureza do problema em estudo. Geralmente, os meios porosos reconstruídos virtualmente podem ser aleatórios ou arranjos ordenados de obstáculos bi ou tridimensionais.

Na maioria dos estudos analíticos em meios porosos recorre-se à equação de *Stokes* (simplificação da equação de *Navier-Stokes* válida em escoamentos com $Re \ll 1$) aplicada a um domínio específico com condições periódicas. Os progressivos desenvolvimentos computacionais e de algoritmos numéricos tornou possível a realização de simulações numéricas de escoamentos em meios porosos de geometrias bastante próximas às reais, utilizando-se para isso diferentes configurações geométricas, tais como: esferas, cilindros, paralelepípedos que de forma ordenada ou aleatória servem para reconstruir estruturas porosas. Referência ainda para a dependência existente entre a configuração geométrica dos elementos, natureza e aplicação do meio poroso a simular.

5.4.1. Previsão da Permeabilidade em Arranjos de Esferas

Com o intuito de obter resultados físicos do trabalho de construção de geometrias e visualização desenvolvido na presente dissertação, fez-se uma análise de previsão de permeabilidade em determinados meios porosos. Os meios porosos simulados consistiram em conjuntos de esferas ordenadas colocadas no interior dum domínio cúbico, no caso o arranjo Cúbico Simples (CS), Cúbico de Corpo Centrado (CCC) e Cúbico de Faces Centradas (CFC). Tratam-se de estruturas que representam com algum acerto estruturas porosas de interesse, por esse motivo, na literatura existem desde algum tempo artigos e trabalhos relacionados com o assunto [74 - 76].

Na presente secção utilizaram-se as correlações analíticas introduzidas por Sangani e Acrivos [74] que prevêem a permeabilidade em função da porosidade para os 3 tipos de arranjos de esferas mencionados (CS, CFC e CCC), utilizando para o efeito soluções de séries expandidas para a solução do escoamento de *Stokes*, do seguinte modo:

$$K = \frac{L^3}{6\beta\pi a C_d} \quad (5.4)$$

Em que L é o comprimento da aresta do domínio cúbico que envolve as esferas, K a permeabilidade, e a o raio de cada esfera, β corresponde ao número de esferas que cabem no volume cúbico L^3 sendo igual a 1, 2 e 4 para os arranjos CS, CCC e CFC respectivamente; já o C_d é o coeficiente de arrasto e é calculado do seguinte modo:

$$C_d = \sum_{n=0}^{30} \alpha_n \chi^n \quad (5.5)$$

Em que α_n são coeficientes para cada tipo de arranjo [74] e χ é função da fracção volúmica sólida do domínio, c , escrita do seguinte modo:

$$\chi = \left(\frac{c}{c_{m\acute{a}x}} \right)^{\frac{1}{3}} \quad (5.6)$$

$c_{m\acute{a}x}$ é a fracção sólida máxima de cada arranjo determinado a partir do raio esférico máximo presente num cubo de lado L no qual esferas se encontrem encostadas sem se sobreporem. Para os arranjos CS, CCC e CFC $c_{m\acute{a}x}$ é igual a $\pi/6$, $\sqrt{3}\pi/8$ e $\sqrt{2}\pi/6$, respectivamente. Por sua vez a porosidade das diferentes estruturas é calculada mediante a equação 5.7:

$$\varphi = 1 - \frac{V_{s\acute{o}lido}}{V_{total}} \quad (5.7)$$

Em que $V_{s\acute{o}lido}$ é o volume ocupado pelos objectos esféricos no domínio cúbico e V_{total} o volume total do domínio cúbico, no caso L^3 .

De modo a comparar resultados numéricos provenientes do código utilizado com valores das correlações analíticas referidas, foram construídas as 3 estruturas cúbicas mencionadas, nas quais, para se obter diferentes valores de porosidade, se fez variar o raio a entre 10 e 24 para a estrutura CS, 8 a 20 para a CCC e 6 a 17 para a CFC. A Tabela 5.2 apresenta o raio das esferas e respectiva porosidade das estruturas simuladas. Referir ainda que o domínio das simulações foi de 51x51x51 nós, por se tratar de um domínio que representa as superfícies esféricas com um nível de definição razoável sem requerer tempos de processamento demasiadamente elevados.

Tabela 5.2 - Raios de esferas e respectivas porosidades utilizadas nas simulações de $L = 51$.

CS		CCC		CFC	
Raio [l.u.]	Porosidade ϕ	Raio [l.u.]	Porosidade ϕ	Raio [l.u.]	Porosidade ϕ
24	0,56	20	0,49	17	0,38
22	0,66	18	0,63	16	0,48
20	0,75	16	0,74	15	0,57
16	0,87	14	0,83	14	0,65
14	0,91	12	0,89	12	0,78
10	0,97	10	0,94	10	0,87
		8	0,97	8	0,94
				6	0,97

As Figuras 5.25, 5.26 e 5.27 apresentam exemplos de geometrias construídas para cada tipo de estrutura cúbica.

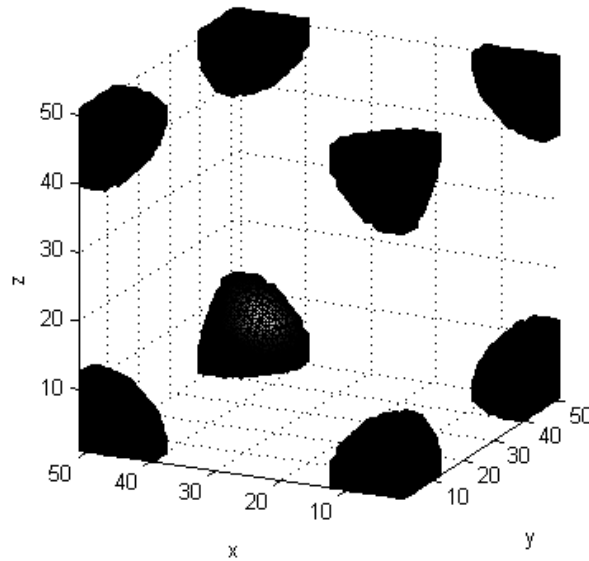


Figura 5.25 - Geometria cúbica simples de raio $a = 12$, num domínio de $L = 51$, com porosidade $\phi = 0,95$.

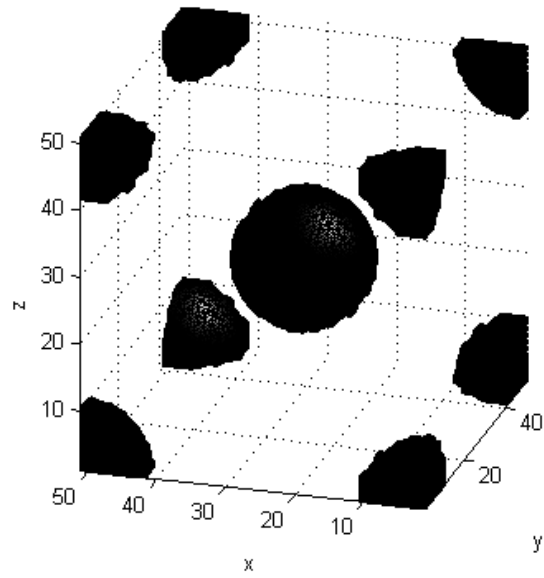


Figura 5.26 - Geometria cúbica de corpo centrado de raio $a = 10$, num de domínio de $L = 51$, com porosidade $\phi = 0,94$.

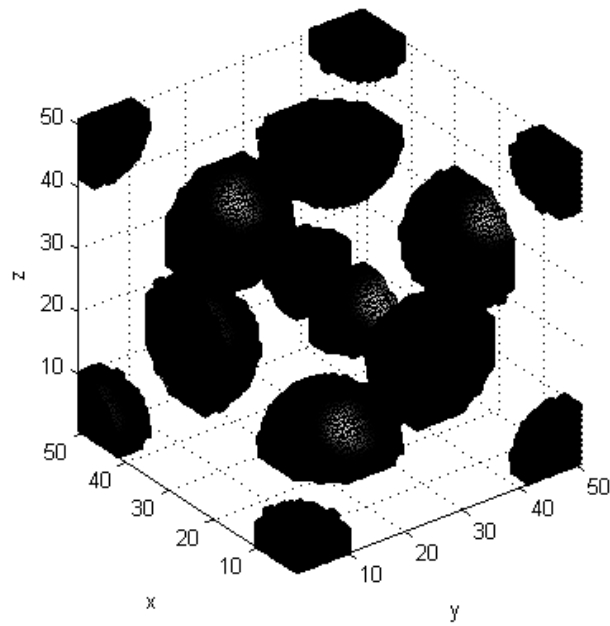


Figura 5.27 - Geometria cúbica de faces centradas de raio $a = 10$, num domínio de $L = 51$, com porosidade $\phi = 0,87$.

Para o cálculo da permeabilidade mediante os resultados numéricos foi utilizada a Lei de Darcy (eq. 5.3). Para tal extraiu-se a velocidade média do escoamento $\langle u \rangle$ da média da componente em "x" de cada nó de fluido do domínio; a viscosidade dinâmica do fluido é calculada da seguinte forma [11]:

$$\mu = \rho \times \frac{\tau - 0,5}{3} \quad (5.7)$$

Em que ρ é a densidade do fluido e τ o tempo de relaxação. Por fim o gradiente de pressão obtém-se através da equação 5.8 [10]:

$$\Delta P = \frac{\Delta u \times \rho}{\tau \times A} \quad (5.8)$$

Em que Δu é a diferença de velocidade média de entrada em "x", ρ a densidade do fluido, τ o tempo de relaxação e A a superfície de objectos sólidos na direcção do escoamento. Isto porque segundo a 2ª lei de Newton:

$$F = ma = m \frac{du}{dt} \quad (5.9)$$

Sabendo que a densidade é proporcional à massa e que o tempo de relaxação é o tempo entre colisões, a equação (5.9) pode ser reescrita da seguinte forma:

$$\Delta u = \frac{\tau \times F}{\rho} \quad (5.10)$$

Seguidamente para obter a equação (5.8) basta dividir ambos os membros por A , obtendo-se assim P resultado de $P = F/A$. De notar ainda que estes cálculos têm todos de ter em conta a conversão das unidades *lattice* (l.u.) para unidas físicas S.I., as mesmas encontram-se apresentadas na Tabela 5.3, [11]:

Tabela 5.3 - Conversão de unidades *lattice* para unidades físicas [11].

Grandeza	Unidades <i>lattice</i>	Unidades S.I.
Distância (dx)	1 l.u.	1×10^{-5} [m]
Tempo (dt)	1 l.u.	1×10^{-7} [s]
Velocidade (u)	1 l.u.	1×10^2 [m/s]
Viscosidade (μ)	1 l.u.	1×10^{-3} [m ² /s]

As Figuras 5.28 e 5.29 apresentam imagens das simulações realizadas para o arranjo CCC e CFC, respectivamente. Ambas possuem $L = 51$, e $a = 16$. Seguidamente as Figuras 5.30, 5.31, 5.32 e 5.33 apresentam cortes de ambas as simulações em $y = 1$ e $y = 26$. Pode-se observar que junto às superfícies sólidas os vectores de velocidade possuem menor

grandeza indicando que o escoamento se desenvolve com menor rapidez, devido a questões relacionadas com atrito e colisões com as superfícies sólidas; é ainda possível verificar que em zonas de menor secção o escoamento desenvolve-se com maior velocidade como já foi anteriormente visto, nomeadamente no teste executado no Capítulo 4 ao princípio da continuidade.

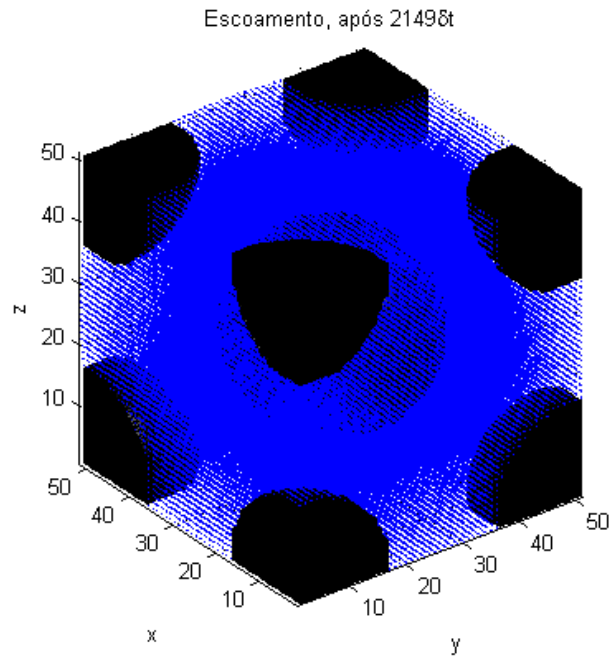


Figura 5.28 - Simulação numérica de escoamento através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\phi = 0,74$).

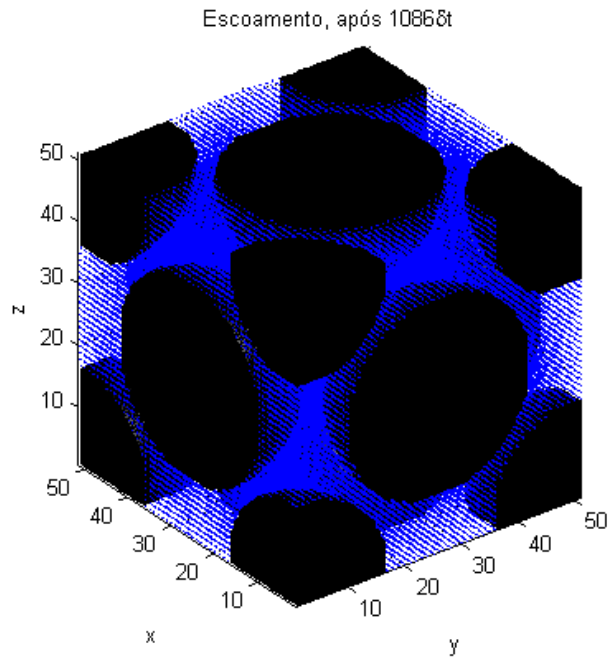


Figura 5.29 - Simulação numérica de escoamento através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\varphi = 0,48$).

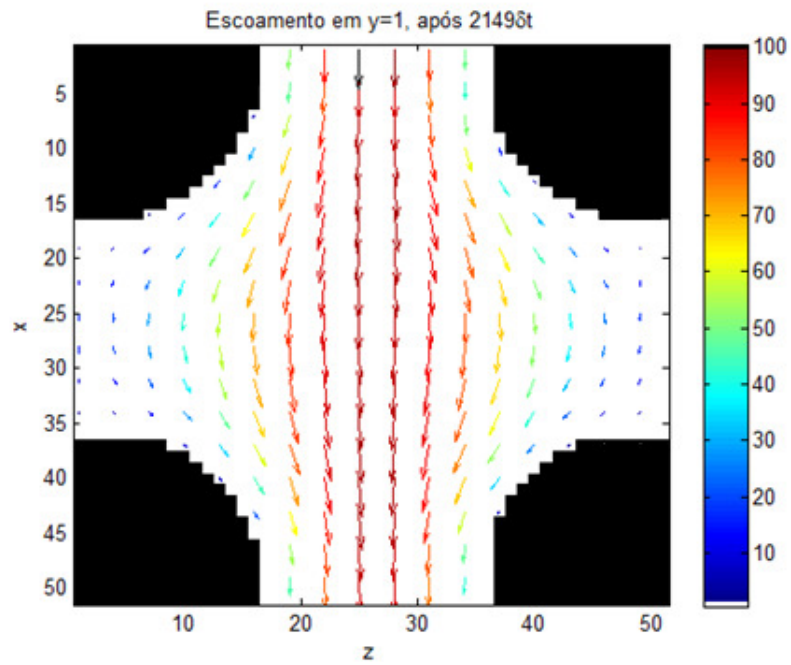


Figura 5.30 - Simulação de escoamento tridimensional através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\varphi = 0,74$); corte em $y = 1$.

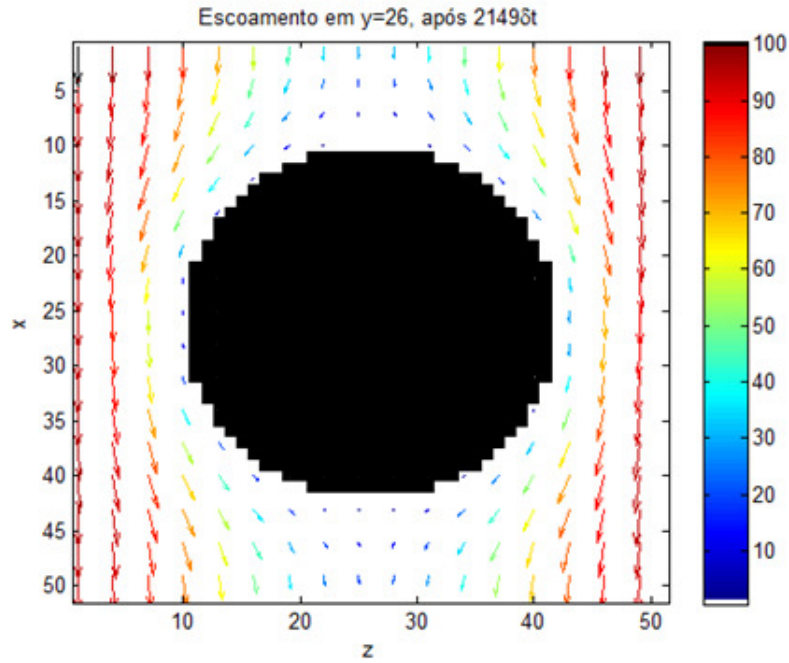


Figura 5.31 - Simulação de escoamento tridimensional através da estrutura CCC com $L = 51$, $a = 16$ (porosidade $\varphi = 0,74$); corte em $y = 26$.

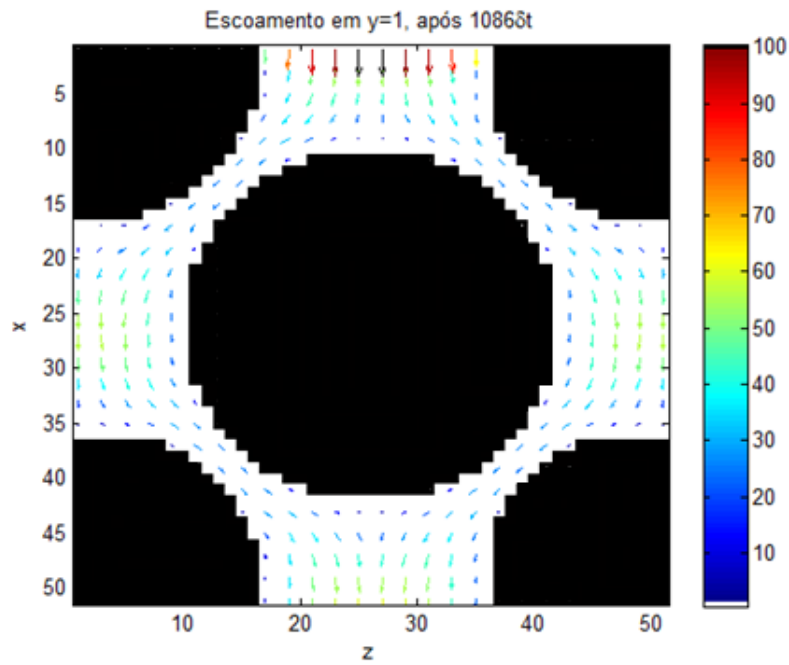


Figura 5.32 - Simulação de escoamento tridimensional através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\varphi = 0,48$); corte em $y = 1$.

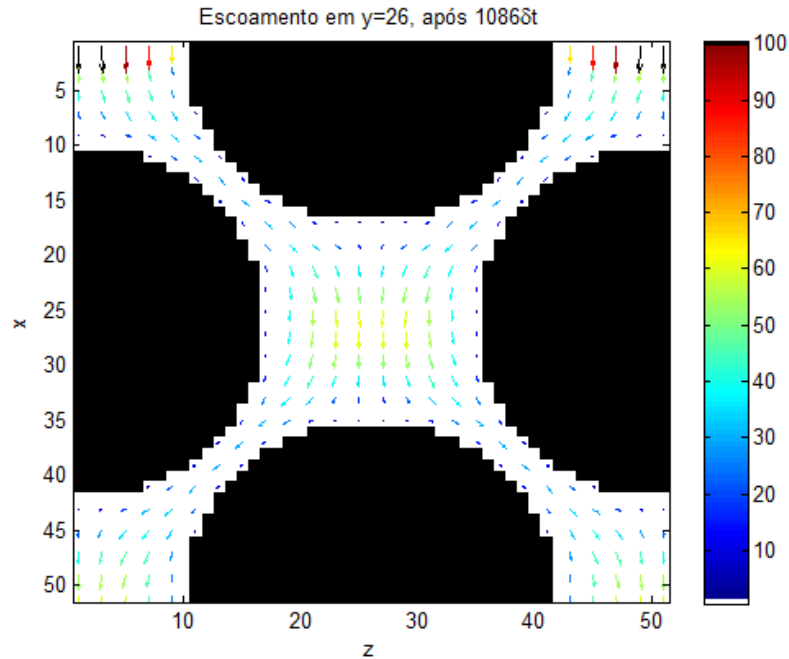


Figura 5.33 - Simulação de escoamento tridimensional através da estrutura CFC com $L = 51$, $a = 16$ (porosidade $\phi = 0,48$); corte em $y = 26$.

As figuras apresentadas, revelam baixa definição nos contornos dos círculos e semi-círculos dos cortes efectuados às esferas e semi-esferas devido ao facto do domínio possuir $51 \times 51 \times 51$ nós. Com o intuito de demonstrar que o código tem condições para elaborar geometrias de melhor qualidade, a Figura 5.34 apresenta uma geometria CFC de um domínio de $291 \times 291 \times 291$ com semi-esferas de 40 l.u. de raio; já as Figuras 5.35 e 5.36 apresentam cortes de uma geometria CFC de um domínio de $291 \times 291 \times 291$ na qual se pode ver contornos mais circulares nos cortes das esferas e semi-esferas de raio 80 l.u.. De notar que não foram executadas simulações para domínios desta ordem devido ao facto do tempo de simulação ser extremamente elevado, não se justificando para aplicação em causa.

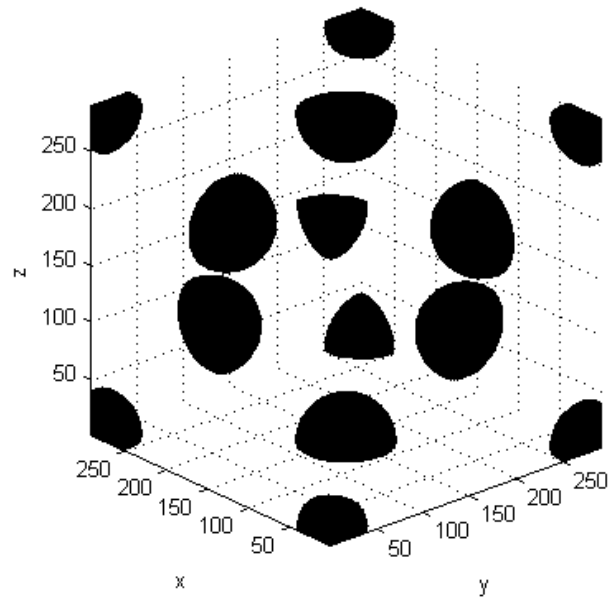


Figura 5.34 - Geometria cúbica da faces centradas de raio $a = 40$, num domínio $L = 291$, com porosidade $\phi = 0,96$.

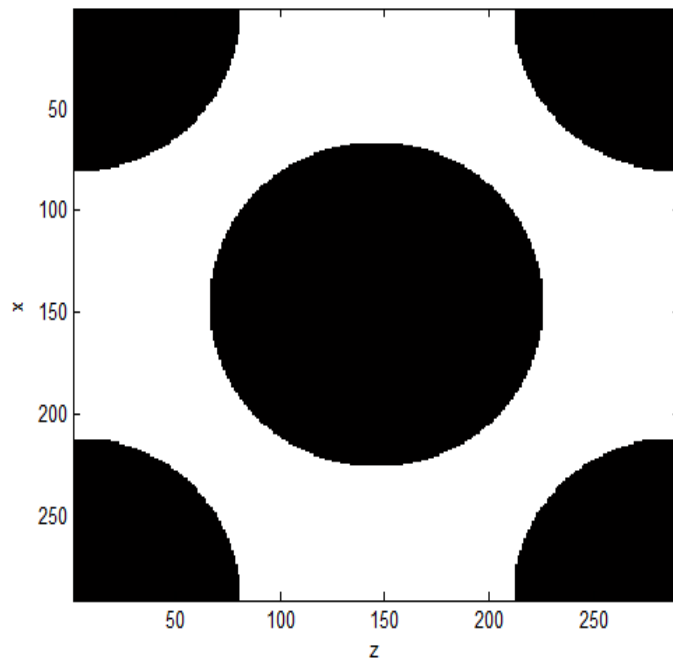


Figura 5.35 - Geometria CFC de $L = 291$, $a = 80$ (porosidade $\phi = 0,65$); corte em $y = 1$.

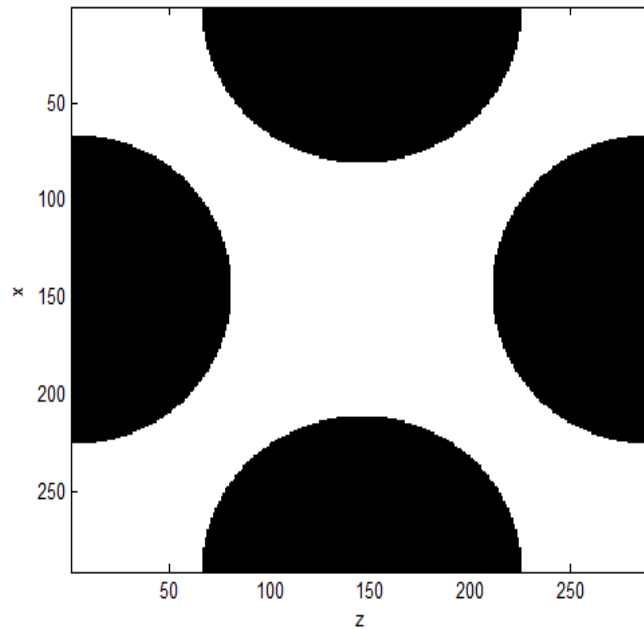


Figura 5.36 - Geometria CFC de $L = 291$, $a = 80$ (porosidade $\phi = 0,65$); corte em $y = 146$.

A Figura 5.37 compara valores de permeabilidade previstos a partir resultados numéricos (Lei de Darcy) com os calculados mediante a correlação de Sangani e Acrivos [74]. É visível uma concordância bastante boa entre ambas as soluções, notando-se apenas um desvio para porosidades inferiores a 0,75 na estrutura CS e ainda um pequeno desvio nas estruturas CCC e CFC para porosidades mais reduzidas. Estes desvios podem ser explicados pelo facto da solução analítica não ser exacta para baixas porosidades, como atestado por Sangani e Acrivos [74]. Deve ser ainda considerado que as condições fronteira implementadas possam também contribuir para as discrepâncias mencionadas.

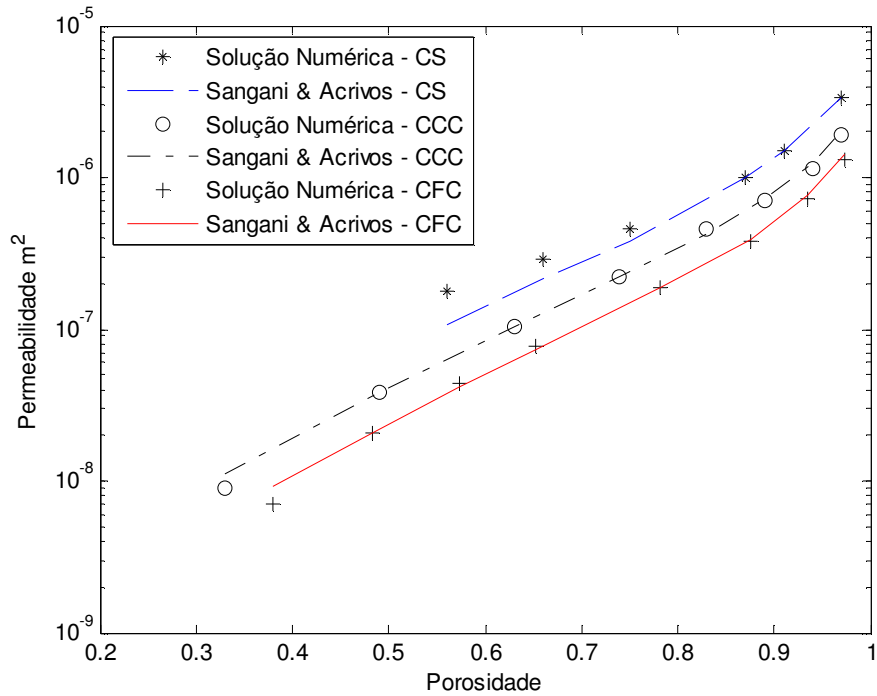


Figura 5.37 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos, em oposição à correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 51$.

A permeabilidade prevista pode ser normalizada em função da porosidade ϕ , da seguinte forma [74]:

$$\frac{K}{a^2} = \exp(a_3\Phi^3 + a_2\Phi^2 + a_1\Phi + a_0) \quad (5.11)$$

$$\Phi = \ln\left(\frac{\phi}{1-\phi}\right) \quad (5.12)$$

Em que os parâmetros de ajuste a_3 , a_2 , a_1 e a_0 são iguais a 0,0118, -0,2093, 2,2648 e -4,6254, respectivamente. A correlação mencionada é válida para os três tipos de estruturas cúbicas tratadas. A Figura 5.38 apresenta o comportamento das soluções numéricas obtidas em relação à curva de permeabilidade normalizada. Constata-se uma concordância muito boa entre as curvas analíticas e as soluções numéricas, exceptuando os três pontos representativos da permeabilidade prevista para porosidades menores da estrutura CS, como seria de esperar pelos resultados apresentados na Figura 5.37.

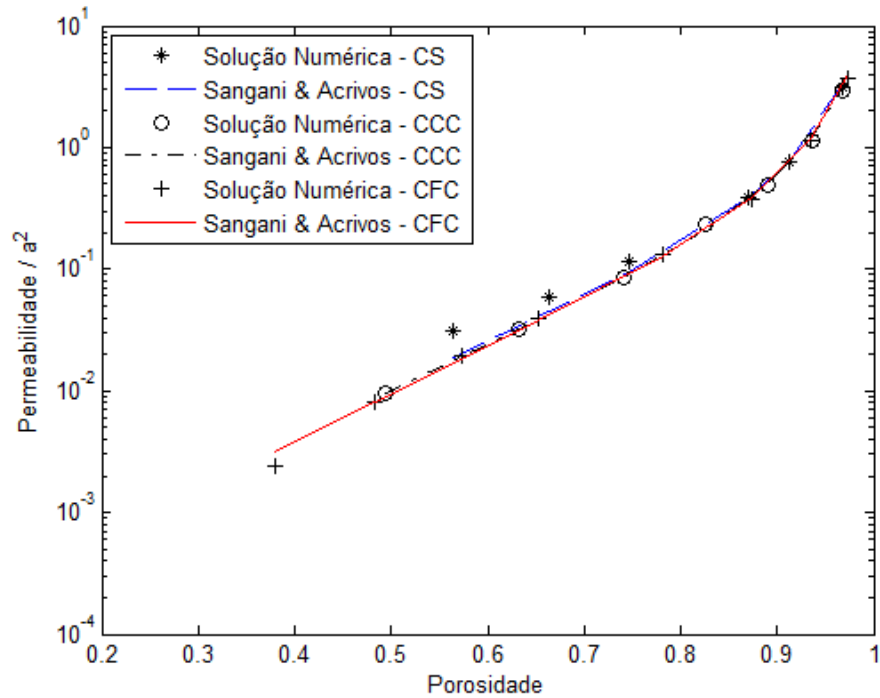


Figura 5.38 - Permeabilidade prevista normalizada pelo raio das esferas em função dos diferentes arranjos de esferas: CS, CC e CFC.

De modo a analisar o comportamento da permeabilidade prevista em domínios de diferentes dimensões, realizaram-se dois estudos semelhantes ao acima descrito. Para o efeito fez-se variar o comprimento de aresta L e os raios das esferas a . Os dados apresentados na Tabela 5.4 dizem respeito a simulações realizadas em estruturas cúbicas de $L = 31$, estando nela discriminados os raios a das esferas e os respectivos níveis de porosidade dos vários domínios simulados. Por outro lado a Tabela 5.5 apresenta os dados referentes às simulações realizadas em estruturas cúbicas de $L = 81$.

Tabela 5.4 - Raios de esferas com respectivas porosidades utilizados nas simulações de $L = 31$.

CS		CCC		CFC	
Raio [l.u.]	Porosidade (ϕ)	Raio [l.u.]	Porosidade (ϕ)	Raio [l.u.]	Porosidade (ϕ)
14	0,61	13	0,38	10	0,44
13	0,69	12	0,51	9	0,59
12	0,76	11	0,63	8	0,71
11	0,81	10	0,72	7	0,81
8	0,93	9	0,79	6	0,88
6	0,97	8	0,86	5	0,93
		6	0,94	4	0,96
				3	0,98

Tabela 5.5 - Raios de esferas com respectivas porosidades utilizados nas simulações de $L = 81$.

CS		CCC		CFC	
Raio [l.u.]	Porosidade (ϕ)	Raio [l.u.]	Porosidade (ϕ)	Raio [l.u.]	Porosidade (ϕ)
38	0,57	34	0,38	26	0,45
36	0,63	32	0,48	24	0,56
32	0,74	30	0,57	22	0,66
30	0,79	24	0,78	20	0,75
20	0,94	20	0,87	18	0,82
16	0,97	16	0,94	16	0,87
		12	0,97	14	0,91
				10	0,97

Os resultados de permeabilidade prevista para os vários domínios de $L = 31$ encontram-se apresentados na Figura 5.39. Nota-se uma boa concordância entre as soluções numéricas e as curvas analíticas, não havendo variações significativas destes resultados em relação aos homólogos de domínio $L = 51$. Referência apenas para o facto dos valores de permeabilidade serem ligeiramente inferiores que os calculados para o domínio de $L = 51$, isto explica-se pelo facto deste domínio ser de dimensão mais reduzida.

Por seu turno a Figura 5.40 apresenta os resultados de permeabilidade prevista para os vários domínios indicados na Tabela 5.5. A concordância com as soluções analíticas de Sangani e Acrivos continuam a ser boas, não existindo diferenças visíveis entre em termos da concordância destes resultados comparativamente com os exibidos nos anteriores domínios de $L = 51$ e $L = 31$, respectivamente. Nota apenas para o facto dos valores de permeabilidade serem superiores aos dos domínios anteriores já que o espaço para o fluido se deslocar aumentou (mesmo para porosidades iguais); ou seja a permeabilidade não varia apenas com a porosidade, havendo domínios de porosidade igual mas com dimensões/escalas diferentes os valores de permeabilidade também serão diferentes.

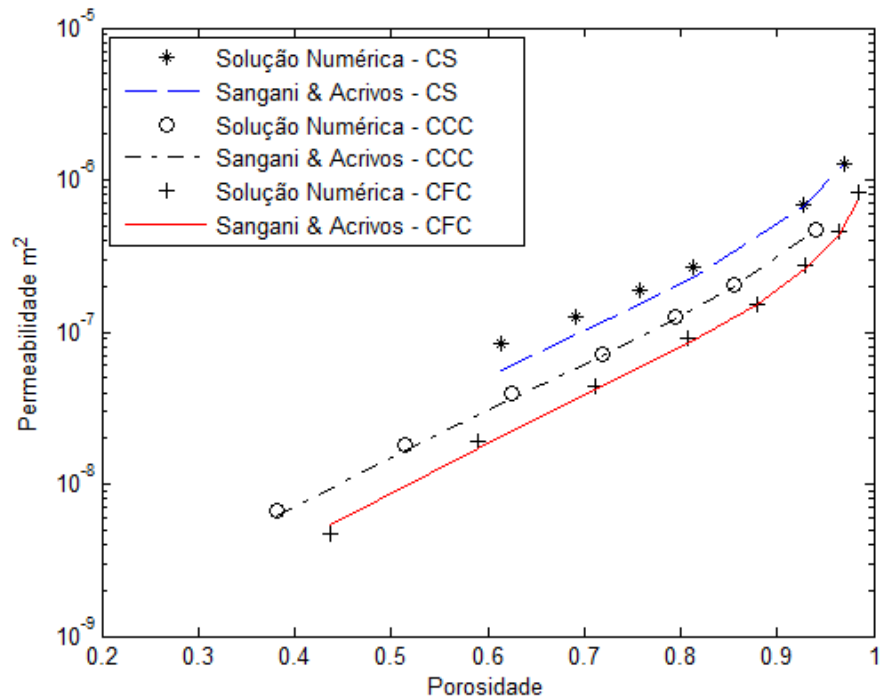


Figura 5.39 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos com a correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 31$.

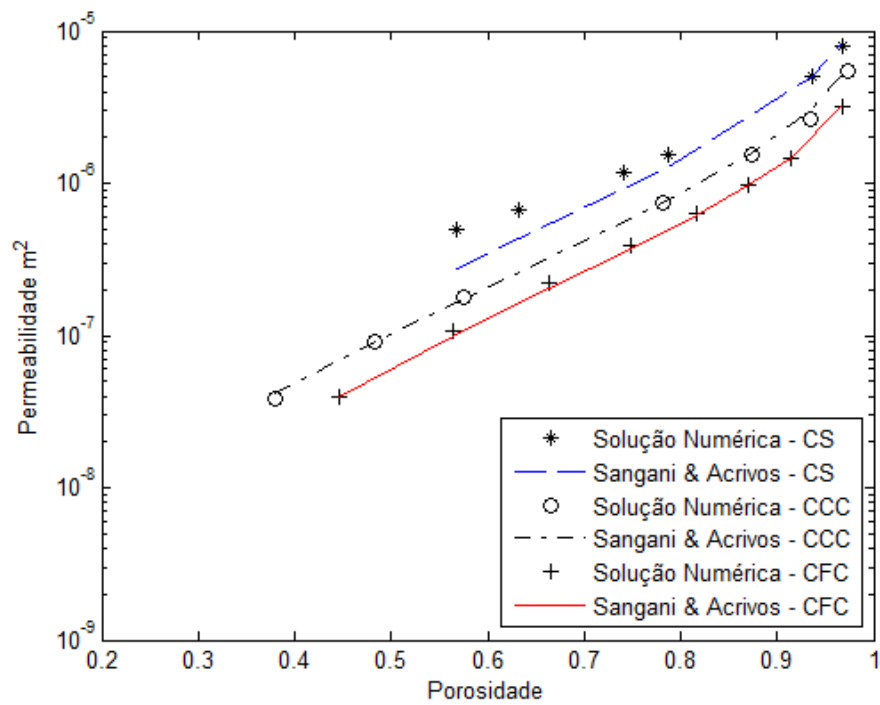


Figura 5.40 - Comparação entre valores de permeabilidade previstos pelos resultados numéricos, com a correlação analítica de Sangani e Acrivos [74], para estruturas cúbicas: CS, CCC e CFC de $L = 81$.

Por fim, para aferir de forma sintetizada e mais clara o comportamento dos vários resultados obtidos, apresenta-se na Figura 5.41 todas as soluções normalizadas. Repara-se que todas têm um comportamento muito idêntico e concordante com as curvas analíticas de Sangani e Acrivos, exceptuando os valores de permeabilidade prevista para as porosidades mais reduzidas das geometrias CS.

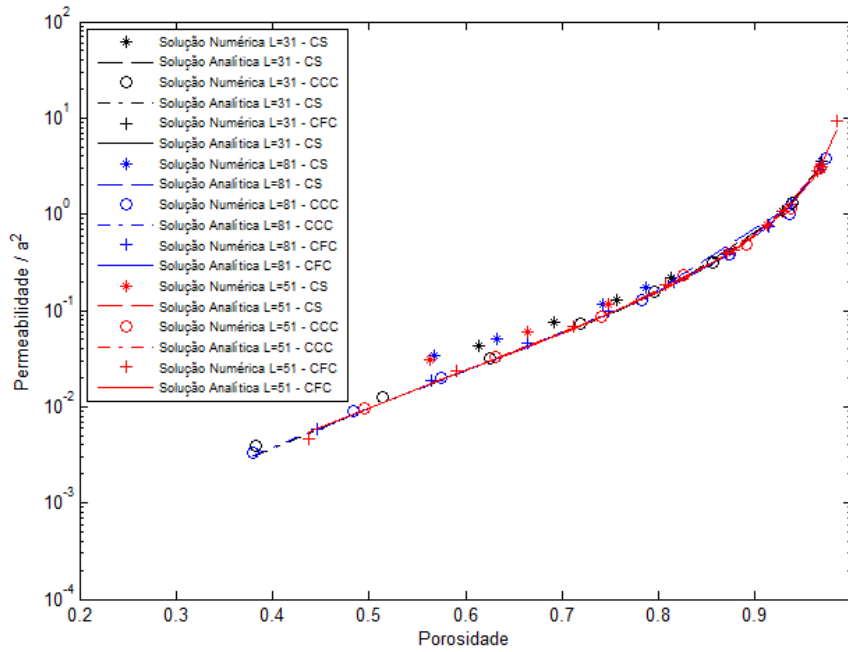


Figura 5.41 - Permeabilidade prevista normalizada pelo raio das esferas em função dos diferentes arranjos de esferas: CS, CC e CFC para domínios de L = 31, L = 51 e L = 81.

6. CONCLUSÕES E TRABALHOS FUTUROS

A presente dissertação abordou o LBM enquanto ferramenta para simulações de escoamentos de fluidos em geometrias complexas, fazendo assim uso da sua natureza mesoscópica. Utilizando-se um código existente na literatura aberta, foi objectivo da presente dissertação simular alguns escoamentos em geometrias complexas.

De forma a validar o código utilizado, realizaram-se dois testes. O primeiro deles passou pela construção de uma geometria representativa de uma conduta bidimensional, para testar o comportamento do escoamento de *Poiseuille*, para tal a solução analítica foi comparada com simulações numéricas com diferentes números de nós ao longo da altura do domínio. Os resultados numéricos apresentaram um erro relativo máximo inferior a 3% para 21 nós ao longo da altura da conduta que por sua vez decresce mediante o aumento número de nós nessa mesma direcção, fixando-se nos 0,5% para uma simulação de 101 nós. Quanto ao erro relativo médio verificou-se que o mesmo também diminui à medida que se aumenta o número de nós da secção. Para as simulações efectuadas, este indicador ronda 1% para 21 nós e cerca 0,03% para a simulação de 101 nós (ao longo da altura). Os resultados obtidos revelam haver consistência com resultados anteriormente divulgados [6, 46], tratando-se de uma gama de erros relativamente baixos comparativamente com outros métodos numéricos [6, 10, 46]. O segundo teste realizado consistiu na verificação da sensibilidade do algoritmo trabalhado à lei da continuidade, para tal integrou-se no código uma geometria com um estreitamento de forma a variar a velocidade do fluido ao longo da conduta. Foram feitas 4 simulações distintas, fixando-se em cada o número de nós da secção longitudinal da conduta e fazendo variar os da secção transversal. Para cada simulação foram feitas 3 medições de caudal (duas em cada extremidade dos domínios e uma na parte central). Apesar de não constar na formulação do LBM, o princípio da continuidade foi verificado através dos resultados numéricos, pois para domínios com mais de 20 nós na secção transversal os erros relativos entre caudais mantém-se abaixo de 1%.

Seguidamente procuraram-se procedimentos capazes de integrar diferentes geometrias nos códigos utilizados. Assim, para o caso de simulações 2D, conseguiu-se a integração no código de geometrias modeladas em ficheiros de imagem, tendo ficado o código capaz de fazer a leitura de imagens mapeando de forma automática geometrias com objectivo final de conseguir correr o escoamento definido pelo algoritmo LBM nas mesmas. Foram também tratados meios capazes de apresentar de forma clara os resultados obtidos por parte dos códigos LBM em que se trabalhou, conseguindo-se efeitos visuais interessantes, caso da elaboração de um mapa de cores associado às simulações 2D e de

um meio capaz de apresentar imagens de simulações 3-D. Na presente dissertação foram ainda realizadas diversas simulações a escoamentos com o intuito de prever a permeabilidade em alguns domínios cúbicos preenchidos por arranjos de esferas; o procedimento passou pela utilização da Lei de Darcy (eq. 5.3) para calcular soluções numéricas de previsão de permeabilidade comparando-as com soluções analíticas conhecidas na literatura [74], no caso para as três seguintes estruturas: Cúbica Simples (CS), Cúbica de Corpo Centrado (CCC) e Cúbica de Faces Centradas (CFC). Os resultados obtidos revelaram uma concordância bastante boa entre ambas as soluções, sendo apenas notória uma ligeira discrepância entre as soluções em porosidades mais reduzidas nas estruturas Cúbicas Simples, facto que pode ser explicado pelo facto da solução analítica não ser exacta para porosidades mais reduzidas.

Em termos de trabalhos futuros poderá ser interessante integrar no modelo novas condições fronteira em particular, ter a flexibilidade de poder fixar a velocidade na entrada do domínio e/ou a pressão na saída [67]. Deste modo, conseguir-se-ia executar testes e simulações sem a condicionante do escoamento à entrada ser igual ao de saída (imposição de condições fronteira periódicas). Julga-se ainda interessante testar o algoritmo trabalhado em geometrias pré-definidas que tenham resultados experimentais associados, (de preferência de escoamentos de Moldação por Transferência de Resina, que foi assunto na presente dissertação), de forma a validar/comparar resultados. É por fim proposto a elaboração de trabalhos dedicados à obtenção de malhas directamente dos resultados experimentais obtidos via NMR/MRI - ressonância magnética nuclear/imagens por ressonância magnética.

REFERÊNCIAS

- [1] Oliveira, L. A., Lopes, A., G., *Mecânica do Fluidos*. ETEP - Edições Técnicas e Profissionais, 2006.
- [2] Temam, R., *Navier-Stokes Equations - Theory and Numerical Analyses*. AMS Chelsea Publishing, 1984, Reprinted 2001.
- [3] Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics*. Vol. 1, 2ª Ed, Springer, 1991, 6th Printing 2005.
- [4] Espedal, A. M., Fasaro, A., Mikelić, A. M., *Filtration in porous media and industrial application: lectures given at the 4th session of the Centro Internazionale Matematico Estivo*. Cetraro, 1998.
- [5] Spanos, T. J. T., *The Thermophysics of Porous Media*. Chapman & Hall/CRC, Florida, 2002.
- [6] Wolf-Gladrow, D. A., *Cellular Automata and Lattice Boltzmann Models – An Introduction*. Springer – Verlag Berlin Heidelberg, 2000.
- [7] Rudd, C.D., Long, A.C., Kendall, K.N., Mangin, C., *Liquid Moulding Technologies - Resin Transfer Moulding, Structural Reaction Injection Moulding and Related Processing Techniques*, Woodhead Publishing, 1997.
- [8] Nabovati, A., Llewellyn, E. W., Sousa, A. C. M., *Through-thickness permeability prediction of three-dimensional multifilament woven fabrics*. Composites part A – applied science and manufacturing, 41 (4): 453-463, April, 2010.
- [9] Nabovati, A., Llewellyn, E. W., Sousa, A. C. M., *A general model for the permeability of fibrous porous media based on fluid flow simulations*. Composites part A – applied science and manufacturing, 40 (6-7): 860-869, July, 2009.
- [10] Suckop, M. C., Thorne Jr., D. T., *Lattice Boltzmann Modeling An Introduction for Geoscientists and Engineers*. Springer, Miami (Florida, USA), 2005.
- [11] Succi, S., *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Clarendon Press, Oxford, 2001.
- [12] Abhijit, S. J., Ying, S., *Wetting dynamics and particle deposition for an evaporating colloidal drop: A lattice Boltzmann study*. Physical Review E 82, 2010.

- [13] Teng, S., Chen, Y., Ohashi, H., *Lattice Boltzmann simulation of multiphase fluid flows through the total variation diminishing with artificial compression scheme*. International Journal of Heat and Fluid Flow 21, 112-121, 2000.
- [14] Zheng, Z. W., Shu, C., Chew, Y.T., *A lattice Boltzmann model for multiphase flows with large density ratio*. Journal of Computational Physics 218, 353-371, 2006.
- [15] Attar, E., Körner, C., *Lattice Boltzmann model free surface flows with liquid-solid phase transition*. International Journal of Heat and Fluid Flow 32, 156-163, 2011.
- [16] Wang, M., Wang, J., Chen, S., *Roughness and cavitations effects on electro-osmotic flows in rough microchannels using the lattice Poisson-Boltzmann methods*. Journal of Computational Physics 226, 836-851, 2007.
- [17] Chen, X.-P., Zhong, C.-W., Yuan, X.-L., *Lattice Boltzmann simulation of cavitating bubble growth with large density ration*. Computers and Mathematics with Applications, 2010. <http://www.sciencedirect.com/science/article/pii/S0898122110004943>, acedido a 4 de Abril de 2011.
- [18] Tseng, C.-C., Wei, Y.-J., Wang, G., Shyy, W., *Review: Modeling of Turbulent, Isothermal and Cryogenic Cavitation Under Attached Conditions*. The Chinese Society of Theoretical and Applied Mechanics and Springer-Verlag GmbH, 2010. http://shyylab.engin.umich.edu/files/papers/Cavitation_review_paper.pdf, acedido a 4 de Abril de 2011.
- [19] Suckop, M. C., Anwar, S., Lee, J. S., Cunningham, K. J., Langevin, C. D., *Modeling Ground-water Flow and Solute Transport in Karst with Lattice Boltzmann Methods*. 2008. <http://pubs.usgs.gov/sir/2008/5023/pdf/32sukop.pdf>, acedido a 4 de Abril de 2011.
- [20] Anwar, S., Cortis, A., Suckop, M., C., *Lattice Boltzmann simulation of solute transport in heterogeneous porous media with conduits to estimate macroscopic continuous time random walk model parameters*. Progress in Computational Fluid Dynamics, Vol. 8, Nos. 1-4, pp. 213-22, 2008.
- [21] Tan, Y.-F., Zhou, Z.-F., *Simulation of solute transport in a parallel single fracture with LBM/MMP mixed method*. Journal of Hydrodynamics, 20 (3), 365-372, 2008.
- [22] Jeong, H. K., Yoon, H. K., Ha, M. Y., *An immersed boundary-thermal lattice Boltzmann method using an equilibrium internal energy density approach for the simulation of flows with heat transfer*. Journal of Computational Physics 229, 2526-2543, 2010.

- [23] Gokaltun, S., Dulikravich, G. S., *Lattice Boltzmann computations of incompressible laminar flow and heat transfer in a constricted channel*. Computers and Mathematics with Applications 59, 2431-2441, 2010.
- [24] Wang, C.-C., Yau, H.-T., Lin, C.-N, Cheng, P.-J., Hung,W.-M., *Application of lattice Boltzmann Method and field synergy principle to the heat transfer analyses of channel flow with obstacles inside*. Thermal Science, Vol.15, Suppl. 1, S75-S80, 2011.
- [25] Sousa, A. C. M., Hadavand, M., Nabovati, A., *Three-dimensional simulation of slip-flow and heat transfer in a microchannel using the lattice Boltzmann method*. Advanced computational methods and experiments in heat transfer XI, 68:75-85, 2010.
- [26] Sousa, A. C. M., Nabovati, A., *LBM mesoscale modelling of porous media*. Advanced computational methods and experiments in heat transfer X, 61:59-68, 2008.
- [27] Nabovati, A., Sousa, A. C. M., *Fluid flow in random porous media at pore level using Lattice Boltzmann Method*. New trends in fluid mechanics research – proceedings on the fifth international conference of fluid mechanics: 518-521, Shanghai, 2007.
- [28] Nabovati, A., Sousa, A. C. M., *Fluid flow simulation at open-porous medium interface using the Lattice Boltzmann Method*. International journal for numerical methods on fluids, 56 (8):1449-1456, 2008.
- [29] Taehun, L., Liu, L., *Lattice Boltzmann simulations of micron-scale drop impact on dry surfaces*. Journal of Computational Physics 229, 8045-8053, 2010
- [30] Blackmor, J., *Ludwing Boltzmann His later life, and philosophy, 1900-1906*. Kluwer Academic Publishers, Netherlands, 1995.
- [31] Brush, S. G., *Kinetic Theory*. Vol. 3.Pergamon Press, Hungary, 1972.
- [32] Chirila, D. B., *Introduction to Lattice Boltzmann Methods*. 2010. http://www.awi.de/fileadmin/user_upload/Research/Research_Divisions/Climate_Sciences/Paleoclimate_Dynamics/Modelling/Lessons/Einf_Ozeanographie/lecture_19_Jan_2010.pdf , acedido a 21 de Fevereiro de 2011.
- [33] Trobec, R, Vajtersic, M, Zinterhof, P., *Parallel Computing Numerics, Applications, and Trends*. Springer, London, 2009.

- [34] Alves, A., *Processamento de placas compósitas pela técnica de moldagem por infusão de resina de Seemann (SCRIMP)*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.
- [35] Hedley, C. W., *Mold filling parameters in resin transfer molding of composites*. MSc Dissertation, Montana State University, Montana, 1994.
- [36] Hakan, Í., *Modeling of Resin Transfer Molding for composites manufacturing*. MSc Dissertation, The Graduate School of Natural and Applied Sciences, Middle East Technical University, Ankara, 2005.
- [37] Potter, K., *Resin Transfer Moulding*. Chapman & Hall, London, 1997.
- [38] Benjamin A., *Massively parallel simulations of multiphase- and multicomponent flows using lattice Boltzmann methods*. PhD Thesis, Faculty of Architecture, Civil Engineering and Environmental Sciences, Technical University Carolo-Wilhelmina zu Braunschweig, Hannover, 2009.
- [39] Persson, P., Strang G., *A simple mesh generator in Matlab*. SIAM Review, Volume 46 (2), pp. 329-345, 2004.
- [40] Pengzhi, L., *Numerical Modeling of Water Waves*, Taylor & Francis, New York, 2008.
- [41] Bernsdorf, J. M., *Simulation of Complex Flows and Multi-Physics with the Lattice Boltzmann Method*. PhD Thesis, University of Amsterdam, Amsterdam, 2008.
- [42] Imagem retirada do endereço da Internet: <http://www.hanleyinnovations.com/racecarpack.html>, acessado a 15 de Abril de 2010.
- [43] Bhatnagar, P.L., Gross, E.P., and Krook, M., *A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems*, Physical Review, 94(3) pp. 511-525, 1954.
- [44] Hardy, J., Pomeau, Y., Pazzis, de O., *Molecular Dynamics of a classical lattice gas: Transport properties and time correlation functions*. Phys. Rev. A, 13: 1949 - 1961, 1976.
- [45] Frisch, U., Hasslacher, B., Pomeau, Y., *Lattice-Gas Automata for the Navier-Stokes Equation*. Phys. Rev. Lett. 56 (14): 1505, 1986.
- [46] Nabovati, A., *Pore level simulation of single and two phase flow in porous media using lattice Boltzmann method*. PhD Thesis, Department of Mechanical Engineering, The University of New Brunswick, Canada, 2009.

- [47] d’Humières, D., Lallemand, P., Frisch, U., *Lattice Gas Models for 3D Hydrodynamics*. Europhys. Lett. 2 (4): 291, 1986.
- [48] Wolfram, S., *Cellular automaton fluids 1: Basic theory*. Journal of Statistical Physics 45 (3-4): 471, 1986.
- [49] Frisch, U., d’Humières, D., Hasslacher, B., Lallemand, P., Pomeau, Y., Rivet, J.-P., *Lattice Gas Hydrodynamics in Two and Three Dimensions*. Complex Systems 1 (4): 649, 1987.
- [50] Hénon, M., *Isometric Collision Rules for the Four-Dimensional FCHC Lattice Gas*. Complex Systems 1 (3): 475, 1987.
- [51] d’Humières, D., Lallemand, P., Searby, G., *Numerical Experiments on Lattice Gases: Mixtures and Galilean Invariance*. Complex Systems 1 (4): 633, 1987.
- [52] Hénon, M., *Viscosity of a Lattice Gas*, Complex Systems 1 (4): 763, 1987.
- [53] Kadanoff, L., McNamara, G., Zanetti, G., *A Poiseuille Viscometer for Lattice Gas Automata*. Complex Systems 1 (4): 791, 1987.
- [54] McNamara, G., Zanetti, G., *Analysis of the Boltzmann Equation to Simulate Lattice-Gas Automata*. Phys. Rev. Lett. 61 (20): 2332, 1988.
- [55] Quian, Y., H., d’Humières, D., Lallemand, P., *Lattice BGK models for navier-stokes equation*. Europhys. Lett., 17(6), 1992.
- [56] Chen, H., Chen, S., Mattheaus, W., *Recovery of navier-stokes equations using a lattice-gas Boltzmann method*. Phys. Rev. A, 45(8), 1992.
- [57] Scholarpedia – the peer reviewed open-access encyclopedia [http://www.scholarpedia.org/article/Lattice Boltzmann Method](http://www.scholarpedia.org/article/Lattice_Boltzmann_Method), acedido a 10 de Maio de 2011.
- [58] Thürey, N., *A single-phase free-surface Lattice Boltzmann Method*. MSc Dissertation, Department of Computer Science, University of Nuremberg, 2002.
- [59] Chapman, S. *On the law distribution of molecular velocities, and on the theory of viscosity and thermal conduction, in a non-uniform simple monatomic gas*. Phil. Trans. Roy. Soc., A216:279-348, 1916.

- [60] Chapman, S. *On kinetic theory of a gas. Part II. – A composite monatomic gas: Diffusion, viscosity, and thermal conduction.* Phil. Trans. Roy. Soc., A217:115-197, 1918.
- [61] Carrington, M., E., Defu, H., Kobes., R., *Chapman-Enskog expansion of the Boltzmann equation and its diagrammatic interpretation.* University of Winnipeg, Manitoba, R3B 2E9 Canada, 2001.
- [62] Sbragaglia, M., Succi, S., *A note on the lattice Boltzmann method beyond the Chapman-Enskog limits.* Europhysics Letters, 73 (3), pp.370-376, 2006.
- [63] Ziegler, D. P., *Boundary Conditions for Lattice Boltzmann Simulations.* Journal of Statistical Physics, Vol. 71, Nos. 5/6, 1993.
- [64] Ginzburg, I., Adler, P., *Boundary flow condition analysis for the three-dimensional lattice Boltzmann model.* J. Phys. II France 4 (2): 191, 1994.
- [65] Chen, S., Martínez, D., *On boundary conditions in lattice Boltzmann methods.* Phys. Fluids, Vol. 8, No. 9, 1996.
- [66] Maier, R. S., Bernard, R. S., Grunau, D. N. *Boundary conditions for the lattice Boltzmann method.* Phys. Fluids, Vol. 8, No. 7, 1996.
- [67] Zou, Q., He, X, *On pressure and velocity flow boundary conditions and bounceback for the lattice Boltzmann BGK model.* Phys. Fluids, Vol. 9, No. 6, 1997.
- [68] Zhaoli, G., Zheng, C., *An extrapolation method for boundary conditions in lattice Boltzmann method.* Phys. Fluids, Vol. 14, No. 6, 2002
- [69] Ginzburg, I., d’Humières, D., *Multireflection boundary conditions for lattice Boltzmann models.* Physical Review E 68, 066614, 2003.
- [70] Mattila, K., *Implementation Techniques for the Lattice Boltzmann Method.* Academic Dissertation, Jyväskylä Studies in Computing 17, Faculty of Information Technology, University of Jyväskylä, 2010.
- [71] Códigos Numéricos em: <http://www.exolete.com/code/lbm>, acessado a 25 de Fevereiro de 2011.
- [72] Hanselman, D., Littlefield, B., *MATLAB 6 Curso Completo.* Prentice Hall, São Paulo, 2003.

[73] Centro de troca de informação da MathWorks: <http://www.mathworks.com/matlabcentral/fileexchange/>, acessado a 31 de Março de 2011.

[74] Sangani, A. S., Acrivos, A., *Slow Flow through a Periodic Array of Spheres*. International Journal of Multiphase Flow, 8(4) pp. 343-60, 1982.

[75] Saeger, R., B., *Transport processes in periodic porous media*. Journal of Fluid Mechanics, 299: 1-15, 1995.

[76] Boutin, C., Geindreau, C., *Estimates and bounds of dynamic permeability of granular media*. Journal of the Acoustical Society of America, Vol. 124, 6, pp. 3576-3593, 2008.

ANEXO I – Código de Validação - Escoamento de *Poiseuille*

Solução Analítica

```
L=(ny-2)/2;
nodos_altura=[-(L-0.5):(L-0.5)];
EXACT=-((nodos_altura.^2-L^2);
tau = 1/omega;
temp_F = density * deltaU / tau;
temp_viscosity = (tau-0.5)/3;
calcfactor=temp_F/(2*temp_viscosity);
EXACT=EXACT*calcfactor;
```

Solução Numérica

```
NUM=[UX(1,2:ny-1)];
```

Gráfico Solução Analítica vs Solução Numérica

```
figure;
plot(EXACT,nodos_altura);hold on;
plot(NUM,nodos_altura,'r*');
legend('Solução Analítica','Solução Numérica');
xlabel('U'); ylabel('nº de nodos (h)');
title('Solução Numérica versus Solução Analítica');
```

Erros

```
figure;
plot(abs(1-(NUM./EXACT)).*100,nodos_altura,'*');
maxerro=max(abs(1-(NUM./EXACT)).*100);
errormedio=mean(abs(1-(NUM./EXACT)).*100);
title('Erro relativo ao longo da altura da conduta');
xlabel('Erro Relativo %'); ylabel('nº de nodos (h)');
```

ANEXO II – Código de Validação - Princípio da Continuidade

Cálculo de caudais nos 3 pontos do domínio 100x50

```
caudal1=sum(UX(2,3:ny-2))+0.75.*(UX(2,2)+UX(2,ny-1));  
caudal2=sum(UX(nx/2,8:ny-7))+0.75.*(UX(nx/2,7)+UX(nx/2,ny-6));  
caudal3=sum(UX(nx,3:ny-2))+0.75.*(UX(nx,2)+UX(nx,ny-1));  
caudal1 = 3.8134e-005;  
caudal2 = 3.8120e-005;  
caudal3 = 3.8134e-005;
```

Erro relativo máximo do domínio 100x50

```
erro_relativo_max=(caudal3-caudal2)/caudal3;  
erro_relativo_max = 3.8430e-004;
```

Gráfico dos erros relativos máximos

```
y=[1.780 0.25 0.085 0.038];  
x=[20 30 40 50];  
figure;  
plot(x,y, '*');  
axis([10 60 0 2]);  
xlabel('Nº de nós da altura do domínio');  
ylabel('Erro relativo máximo %');
```

ANEXO III - Código D2Q9 Gerador da Simulação da “Geometria 1”

Definição do domínio

```
nx=60;
ny=60;
F=repmat(density/9,[nxny 9]);
FEQ=F;
msize=nx*ny;
CI=[0:msize:msize*7];
```

Parâmetros de entrada

```
omega=1.0;
density=1.0;
t1=4/9;
t2=1/9;
t3=1/36;
c_squ=1/3;
avu=1; prevavu=1; ts=0;
deltaU=1e-7;
numactivenodes=sum(sum(sum(1-BOUND)));
```

Integração da Geometria

```
A=imread('Malha4.jpg');
B=rgb2gray(A);
C=B<128;
BOUND=C';
```

Reconhecimento dos nós para aplicação do Bounce Back

```
ON=find(BOUND);
TO_REFLECT=[ON+CI(1) ON+CI(2) ON+CI(3) ON+CI(4) ...
ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8)];
REFLECTED=[ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8) ...
ON+CI(1) ON+CI(2) ON+CI(3) ON+CI(4)];
```

Ciclo de propagação e distribuição de partículas

```
while (ts<4000 & 1e-10<abs((prevavu-avu)/avu)) | ts<100
```

Hipóteses de propagação de partículas

```
F(:,4)=F([2:nx 1],[ny 1:ny-1],4);
F(:,3)=F(:,[ny 1:ny-1],3);
F(:,2)=F([nx 1:nx-1],[ny 1:ny-1],2);
F(:,5)=F([2:nx 1],:,5);
F(:,1)=F([nx 1:nx-1],:,1);
F(:,6)=F([2:nx 1],[2:ny 1],6);
```

```
F(:,7)=F(:,[2:ny 1],7);
F(:,8)=F([nx 1:nx-1],[2:ny 1],8);
```

Implementação do “bounceback” no presente instante de tempo

```
BOUNCEDBACK=F(TO_REFLECT);
```

Cálculo da densidade em cada nó

```
DENSITY = sum(F,4);
```

Cálculo da velocidade em cada nó

```
UX=(sum(F(:,:[1 2 8]),3)-sum(F(:,:[4 5 6]),3))./DENSITY;
UY=(sum(F(:,:[2 3 4]),3)-sum(F(:,:[6 7 8]),3))./DENSITY;
```

Condição inicial imposta ao fluido

```
UX(1,:)=UX(1,:)+deltaU;
```

Cálculo dos vectores de velocidade e respectivas grandezas

```
UX(ON)=0;
UY(ON)=0;
DENSITY(ON)=0;
U_SQU=UX.^2+UY.^2;
U_C2=UX+UY;
U_C4=-UX+UY;
U_C6=-U_C2; U_C8=-U_C4;
```

Cálculo das funções de distribuição de equilíbrio

```
FEQ(:,9)=t1*DENSITY.*(1-U_SQU/(2*c_squ));
FEQ(:,1)=t2*DENSITY.*(1+UX/c_squ+0.5*(UX/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,3)=t2*DENSITY.*(1+UY/c_squ+0.5*(UY/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,5)=t2*DENSITY.*(1-UX/c_squ+0.5*(UX/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,7)=t2*DENSITY.*(1-UY/c_squ+0.5*(UY/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,2)=t3*DENSITY.*(1+U_C2/c_squ+0.5*(U_C2/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,4)=t3*DENSITY.*(1+U_C4/c_squ+0.5*(U_C4/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,6)=t3*DENSITY.*(1+U_C6/c_squ+0.5*(U_C6/c_squ).^2-U_SQU/(2*c_squ));
FEQ(:,8)=t3*DENSITY.*(1+U_C8/c_squ+0.5*(U_C8/c_squ).^2-U_SQU/(2*c_squ));
```

Condições de passagem a novo instante de tempo ou fim de ciclo

```
F=omega*FEQ+(1-omega)*F;
F(REFLECTED)=BOUNCEDBACK;
prevavu=avu;avu=sum(sum(sum(UX)))/numactivenodes; ts=ts;
```

Código de visualização

```
figure;
colormap(gray(2));
```



```
image(BOUND'. *1000)
hold on;
quiverc(2:nx,1:ny,UX(2:nx,:),UY(2:nx,:));
colorbar
title(['Flow field after ',num2str(ts),'\deltat']);
xlabel('x'); ylabel('y');
```

ANEXO IV - Código D3Q19 para “Cilindros Intersectados”

Definição do domínio

```
nx=18;
ny=nx;
nz=nx;
F=repmat(density/19,[nxnynz 19]);
FEQ=F;
matsize=nx*ny*nz;
CI=[0:matsize:matsize*19];
```

Parâmetros de entrada

```
omega=1.0;
density=1.0;
t1=1/3;
t2=1/18;
t3=1/36;
avu=1;
prevavu=1;
ts=0;
deltaU=1e-7;
numactivenodes=sum(sum(sum(1-BOUND)));
```

Construção da Geometria

```
BOUND=zeros(nx,ny,nz);
for a=1:nx, for b=1:ny for c=1:nz
BOUND1(a,b,c)=((a-3)^2+(c-6)^2)<2^2;
end, end, end
for a=1:nx, for b=1:ny for c=1:nz
BOUND2(a,b,c)=((a-9)^2+(c-6)^2)<2^2;
end, end, end
for a=1:nx, for b=1:ny for c=1:nz
BOUND3(a,b,c)=((a-15)^2+(c-6)^2)<2^2;
end, end, end
for a=1:nx, for b=1:ny for c=1:nz
BOUND4(a,b,c)=((a-3)^2+(c-12)^2)<2^2;
end, end, end
for a=1:nx, for b=1:ny for c=1:nz
BOUND5(a,b,c)=((a-9)^2+(c-12)^2)<2^2;
end, end, end
for a=1:nx, for b=1:ny for c=1:nz
BOUND6(a,b,c)=((a-15)^2+(c-12)^2)<2^2;
end, end, end
BOUND=(BOUND+BOUND1+BOUND2+BOUND3+BOUND4+BOUND5+BOUND6);
A=imread('Cilindros.bmp');
```

```

B=rgb2gray(A)
C=B<128;
BOUNDDD=C';
BOUNDDD= repmat(BOUNDDD,[1 1 18]);

```

Preparação de nós para aplicação do Bounce Back

```

ON=find(BOUND);
TO_REFLECT=[ON+CI(2) ON+CI(3) ON+CI(4) ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8) ...
ON+CI(9) ON+CI(10) ON+CI(11) ON+CI(12) ON+CI(13) ON+CI(14) ON+CI(15) ...
ON+CI(16) ON+CI(17) ON+CI(18) ON+CI(19)];
REFLECTED=[ON+CI(3) ON+CI(2) ON+CI(5) ON+CI(4) ON+CI(7) ON+CI(6) ON+CI(11) ...
ON+CI(10) ON+CI(9) ON+CI(8) ON+CI(15) ON+CI(14) ON+CI(13) ON+CI(12) ...
ON+CI(19) ON+CI(18) ON+CI(17) ON+CI(16)];

```

Ciclo de propagação e distribuição de partículas

```

while (ts<4000 & 1e-10<abs((prevavu-avu)/avu)) | ts<100

```

Hipóteses de propagação de partículas

```

F(:, :, 2)=F(:, :, [nz 1:nz-1], 2);
F(:, :, 3)=F(:, :, [2:nz 1], 3);
F(:, :, 4)=F(:, :, [ny 1:ny-1], 4);
    F(:, :, 5)=F(:, :, [2:ny 1], 5);
F(:, :, 6)=F([nx 1:nx-1], :, 6);
F(:, :, 7)=F([2:nx 1], :, 7);
    F(:, :, 8)= F([nx 1:nx-1], [ny 1:ny-1], 8);
    F(:, :, 9)= F([nx 1:nx-1], [2:ny 1], 9);
    F(:, :, 10)=F([2:nx 1], [ny 1:ny-1], 10);
    F(:, :, 11)=F([2:nx 1], [2:ny 1], 11);
    F(:, :, 12)=F([nx 1:nx-1], :, [nz 1:nz-1], 12);
    F(:, :, 13)=F([nx 1:nx-1], :, [2:nz 1], 13);
    F(:, :, 14)=F([2:nx 1], :, [nz 1:nz-1], 14);
    F(:, :, 15)=F([2:nx 1], :, [2:nz 1], 15);
    F(:, :, 16)=F(:, [ny 1:ny-1], [nz 1:nz-1], 16);
    F(:, :, 17)=F(:, [ny 1:ny-1], [2:nz 1], 17);
    F(:, :, 18)=F(:, [2:ny 1], [nz 1:nz-1], 18);
F(:, :, 19)=F(:, [2:ny 1], [2:nz 1], 19);

```

Implementação do “bounceback” no presente instante de tempo

```

BOUNCEDBACK=F(TO_REFLECT);

```

Cálculo da densidade em cada nó

```

DENSITY = sum(F,4);

```

Cálculo da velocidade em cada nó

```

UX=(sum(F(:,:,,[6 8 9 12 13]),4)-sum(F(:,:,,[7 10 11 14 15]),4))./DENSITY;
UY=(sum(F(:,:,,[4 8 10 16 17]),4)-sum(F(:,:,,[5 9 11 18 19]),4))./DENSITY;
UZ=(sum(F(:,:,,[2 12 14 16 18]),4)-sum(F(:,:,,[3 13 15 17 19]),4))./DENSITY;

```

Condição inicial imposta ao fluido

```
UX(1,:)=UX(1,:)+deltaU;
```

Cálculo dos vectores de velocidade e respectivas grandezas

```

UX(ON)=0; UY(ON)=0; UZ(ON)=0; DENSITY(ON)=0; U_SQU=UX.^2+UY.^2+UZ.^2;
U8=UX+UY;U9=UX-UY;U10=-UX+UY;U11=-U8;U12=UX+UZ;U13=UX-UZ;
U14=-U13;U15=-U12;U16=UY+UZ;U17=UY-UZ;U18=-U17;U19=-U16;

```

Cálculo das funções de distribuição de equilíbrio

```

FEQ(:,:,,1)=t1*DENSITY.*(1-3*U_SQU/2);
FEQ(:,:,,2)=t2*DENSITY.*(1+3*UZ+9/2*UZ.^2-3/2*U_SQU);
FEQ(:,:,,3)=t2*DENSITY.*(1-3*UZ+9/2*UZ.^2-3/2*U_SQU);
FEQ(:,:,,4)=t2*DENSITY.*(1+3*UY+9/2*UY.^2-3/2*U_SQU);
FEQ(:,:,,5)=t2*DENSITY.*(1-3*UY+9/2*UY.^2-3/2*U_SQU);
FEQ(:,:,,6)=t2*DENSITY.*(1+3*UX+9/2*UX.^2-3/2*U_SQU);
FEQ(:,:,,7)=t2*DENSITY.*(1-3*UX+9/2*UX.^2-3/2*U_SQU);
FEQ(:,:,,8)=t3*DENSITY.*(1+3*U8+9/2*(U8).^2-3*U_SQU/2);
FEQ(:,:,,9)=t3*DENSITY.*(1+3*U9+9/2*(U9).^2-3*U_SQU/2);
FEQ(:,:,,10)=t3*DENSITY.*(1+3*U10+9/2*(U10).^2-3*U_SQU/2);
FEQ(:,:,,11)=t3*DENSITY.*(1+3*U11+9/2*(U11).^2-3*U_SQU/2);
FEQ(:,:,,12)=t3*DENSITY.*(1+3*U12+9/2*(U12).^2-3*U_SQU/2);
FEQ(:,:,,13)=t3*DENSITY.*(1+3*U13+9/2*(U13).^2-3*U_SQU/2);
FEQ(:,:,,14)=t3*DENSITY.*(1+3*U14+9/2*(U14).^2-3*U_SQU/2);
FEQ(:,:,,15)=t3*DENSITY.*(1+3*U15+9/2*(U15).^2-3*U_SQU/2);
FEQ(:,:,,16)=t3*DENSITY.*(1+3*U16+9/2*(U16).^2-3*U_SQU/2);
FEQ(:,:,,17)=t3*DENSITY.*(1+3*U17+9/2*(U17).^2-3*U_SQU/2);
FEQ(:,:,,18)=t3*DENSITY.*(1+3*U18+9/2*(U18).^2-3*U_SQU/2);
FEQ(:,:,,19)=t3*DENSITY.*(1+3*U19+9/2*(U19).^2-3*U_SQU/2);

```

Condições de passagem a novo instante de tempo ou fim de ciclo

```

F=omega*FEQ+(1-omega)*F;
F(REFLECTED)=BOUNCEDBACK;
prevavu=avu;avu=sum(sum(sum(UX)))/numactivenodes; ts=ts;

```

Código de visualização 3D

```

figure;
BOUND=smooth3(BOUND,'box',3);
[X,Y,Z]=meshgrid(1:nx,1:ny,1:nz)
quiver3(Y,X,Z,UY,UX,UZ)

```

```

hold on;
p=isosurface(BOUND,0.5)
P=patch(p)
pp=patch(isocaps(BOUND,0.5))
isonormals(BOUND,P)
view(3);axis vis3d
camlight; lighting phong
xlabel('y')
ylabel('x')
zlabel('z')
title(['3D flow field, after ',num2str(ts),'\deltat'])

```

Código de visualização dos cortes

```

figure;
zcut=9;
colormap(gray(2));
image(BOUND(:,zcut.*1000));
hold on;
quiverc(1:ny,1:nx,UY(:,zcut),UX(:,zcut));
colorbar
xlabel('y');
ylabel('x');
title(['Flow field at z=',num2str(zcut),', after ',num2str(ts),'\deltat']);

```

ANEXO V - Código D3Q19 para “Estrutura CFC de L = 51 e a = 16”

Definição do domínio

```
nx=51;
ny=nx;
nz=nx;
F=repmat(density/19,[nxnynz 19]);
FEQ=F;
matsize=nx*ny*nz;
CI=[0:matsize:matsize*19];
```

Parâmetros de entrada

```
omega=1.0;
density=1.0;
t1=1/3;
t2=1/18;
t3=1/36;
avu=1;
prevavu=1;
ts=0;
deltaU=1e-7;
numactivenodes=sum(sum(sum(1-BOUND)));
r=16
ni=nx-(r+2)
nf=nx-(r-2)
nm=(nx/2)+0.5
```

Construção da Geometria

```
BOUND1=zeros(nx,ny,nz);
for i=1:ni, for j=1:ni, for k=1:ni
    BOUND1(i,j,k)=((i-1)^2+(j-1)^2+(k-1)^2)<r^2;
end, end, end;
for i=nf:nx, for j=nf:ny for k=nf:nz;
    BOUND1(i,j,k)=((i-nx)^2+(j-ny)^2+(k-nz)^2)<r^2;
end, end, end;
for a=nf:nx, for b=nf:ny for c=1:ni;
    BOUND1(a,b,c)=((a-nx)^2+(b-ny)^2+(c-1)^2)<r^2;
end, end, end;
for a=nf:nx, for b=1:ni for c=nf:nz;
    BOUND1(a,b,c)=((a-nx)^2+(b-1)^2+(c-ny)^2)<r^2;
end, end, end;
for aa=1:ni, for bb=nf:ny for cc=nf:nz;
    BOUND1(aa,bb,cc)=((aa-1)^2+(bb-ny)^2+(cc-ny)^2)<r^2;
end, end, end;
for aa=nf:nx, for bb=1:ni for cc=1:ni;
```

```

    BOUND1(aa,bb,cc)=((aa-ny)^2+(bb-1)^2+(cc-1)^2)<r^2;
end, end, end;
for aaa=1:ni, for bbb=1:ni for ccc=nf:nz;
    BOUND1(aaa,bbb,ccc)=((aaa-1)^2+(bbb-1)^2+(ccc-ny)^2)<r^2;
end, end, end;
for aaa=1:ni, for bbb=nf:ny for ccc=1:ni;
    BOUND1(aaa,bbb,ccc)=((aaa-1)^2+(bbb-ny)^2+(ccc-1)^2)<r^2;
end, end, end;
BOUND2=zeros(nx,ny,nz);
for i=1:nx, for j=1:ny, for k=1:nz
    BOUND2(i,j,k)=((i-nm)^2+(j-nm)^2+(k-1)^2)<r^2;
end, end, end;
BOUND3=zeros(nx,ny,nz);
for ii=1:nx, for jj=1:ny for kk=1:nz;
    BOUND3(ii,jj,kk)=((ii-nm)^2+(jj-ny)^2+(kk-nm)^2)<r^2;
end, end, end;
BOUND4=zeros(nx,ny,nz);
for a=1:nx, for b=1:ny for c=1:nz;
    BOUND4(a,b,c)=((a-nx)^2+(b-nm)^2+(c-nm)^2)<r^2;
end, end, end;
BOUND5=zeros(nx,ny,nz);
for a=1:nx, for b=1:ny for c=1:nz;
    BOUND5(a,b,c)=((a-nm)^2+(b-nm)^2+(c-nx)^2)<r^2;
end, end, end;
BOUND6=zeros(nx,ny,nz);
for aa=1:nx, for bb=1:ny for cc=1:nz;
    BOUND6(aa,bb,cc)=((aa-nm)^2+(bb-1)^2+(cc-nm)^2)<r^2;
end, end, end;
BOUND7=zeros(nx,ny,nz);
for aa=1:nx, for bb=1:ny for cc=1:nz;
    BOUND7(aa,bb,cc)=((aa-1)^2+(bb-nm)^2+(cc-nm)^2)<r^2;
end, end, end;
BOUND=BOUND1+BOUND2+BOUND3+BOUND4+BOUND5+BOUND6+BOUND7

```

Preparação de nós para aplicação do Bounce Back

```

ON=find(BOUND);
TO_REFLECT=[ON+CI(2) ON+CI(3) ON+CI(4) ON+CI(5) ON+CI(6) ON+CI(7) ON+CI(8) ...
ON+CI(9) ON+CI(10) ON+CI(11) ON+CI(12) ON+CI(13) ON+CI(14) ON+CI(15) ...
ON+CI(16) ON+CI(17) ON+CI(18) ON+CI(19)];
REFLECTED=[ON+CI(3) ON+CI(2) ON+CI(5) ON+CI(4) ON+CI(7) ON+CI(6) ON+CI(11) ...
ON+CI(10) ON+CI(9) ON+CI(8) ON+CI(15) ON+CI(14) ON+CI(13) ON+CI(12) ...
ON+CI(19) ON+CI(18) ON+CI(17) ON+CI(16)];

```

Ciclo de propagação e distribuição de partículas

```

while (ts<30000 & 1e-10<abs((prevavu-avu)/avu)) | ts<100

```

Hipóteses de propagação de partículas

```
F(:, :, 2) = F(:, :, [nz 1:nz-1], 2);
F(:, :, 3) = F(:, :, [2:nz 1], 3);
F(:, :, 4) = F(:, [ny 1:ny-1], :, 4);
    F(:, :, 5) = F(:, [2:ny 1], :, 5);
F(:, :, 6) = F([nx 1:nx-1], :, :, 6);
F(:, :, 7) = F([2:nx 1], :, :, 7);
    F(:, :, 8) = F([nx 1:nx-1], [ny 1:ny-1], :, 8);
    F(:, :, 9) = F([nx 1:nx-1], [2:ny 1], :, 9);
    F(:, :, 10) = F([2:nx 1], [ny 1:ny-1], :, 10);
    F(:, :, 11) = F([2:nx 1], [2:ny 1], :, 11);
    F(:, :, 12) = F([nx 1:nx-1], :, [nz 1:nz-1], 12);
    F(:, :, 13) = F([nx 1:nx-1], :, [2:nz 1], 13);
    F(:, :, 14) = F([2:nx 1], :, [nz 1:nz-1], 14);
    F(:, :, 15) = F([2:nx 1], :, [2:nz 1], 15);
    F(:, :, 16) = F(:, [ny 1:ny-1], [nz 1:nz-1], 16);
    F(:, :, 17) = F(:, [ny 1:ny-1], [2:nz 1], 17);
    F(:, :, 18) = F(:, [2:ny 1], [nz 1:nz-1], 18);
F(:, :, 19) = F(:, [2:ny 1], [2:nz 1], 19);
```

Implementação do “bounceback” no presente instante de tempo

```
BOUNCEDBACK = F(TO_REFLECT);
```

Cálculo da densidade em cada nó

```
DENSITY = sum(F, 4);
```

Cálculo da velocidade em cada nó

```
UX = (sum(F(:, :, [6 8 9 12 13]), 4) - sum(F(:, :, [7 10 11 14 15]), 4)) ./ DENSITY;
UY = (sum(F(:, :, [4 8 10 16 17]), 4) - sum(F(:, :, [5 9 11 18 19]), 4)) ./ DENSITY;
UZ = (sum(F(:, :, [2 12 14 16 18]), 4) - sum(F(:, :, [3 13 15 17 19]), 4)) ./ DENSITY;
```

Condição inicial imposta ao fluido

```
UX(1, :, :) = UX(1, :, :) + deltaU;
```

Cálculo dos vectores de velocidade e respectivas grandezas

```
UX(ON) = 0; UY(ON) = 0; UZ(ON) = 0; DENSITY(ON) = 0; U_SQU = UX.^2 + UY.^2 + UZ.^2;
U8 = UX + UY; U9 = UX - UY; U10 = -UX + UY; U11 = -U8; U12 = UX + UZ; U13 = UX - UZ;
U14 = -U13; U15 = -U12; U16 = UY + UZ; U17 = UY - UZ; U18 = -U17; U19 = -U16;
```

Cálculo das funções de distribuição de equilíbrio

```
FEQ(:, :, 1) = t1 * DENSITY .* (1 - 3 * U_SQU / 2);
FEQ(:, :, 2) = t2 * DENSITY .* (1 + 3 * UZ + 9 / 2 * UZ.^2 - 3 / 2 * U_SQU);
FEQ(:, :, 3) = t2 * DENSITY .* (1 - 3 * UZ + 9 / 2 * UZ.^2 - 3 / 2 * U_SQU);
```



```

FEQ(:,:,4)=t2*DENSITY.*(1 + 3*UY + 9/2*UY.^2 - 3/2*U_SQU);
FEQ(:,:,5)=t2*DENSITY.*(1 - 3*UY + 9/2*UY.^2 - 3/2*U_SQU);
FEQ(:,:,6)=t2*DENSITY.*(1 + 3*UX + 9/2*UX.^2 - 3/2*U_SQU);
FEQ(:,:,7)=t2*DENSITY.*(1 - 3*UX + 9/2*UX.^2 - 3/2*U_SQU);
FEQ(:,:,8) =t3*DENSITY.*(1 + 3*U8 + 9/2*(U8).^2 - 3*U_SQU/2);
FEQ(:,:,9) =t3*DENSITY.*(1 + 3*U9 + 9/2*(U9).^2 - 3*U_SQU/2);
FEQ(:,:,10)=t3*DENSITY.*(1 + 3*U10 + 9/2*(U10).^2 - 3*U_SQU/2);
FEQ(:,:,11)=t3*DENSITY.*(1 + 3*U11 + 9/2*(U11).^2 - 3*U_SQU/2);
FEQ(:,:,12)=t3*DENSITY.*(1 + 3*U12 + 9/2*(U12).^2 - 3*U_SQU/2);
FEQ(:,:,13)=t3*DENSITY.*(1 + 3*U13 + 9/2*(U13).^2 - 3*U_SQU/2);
FEQ(:,:,14)=t3*DENSITY.*(1 + 3*U14 + 9/2*(U14).^2 - 3*U_SQU/2);
FEQ(:,:,15)=t3*DENSITY.*(1 + 3*U15 + 9/2*(U15).^2 - 3*U_SQU/2);
FEQ(:,:,16)=t3*DENSITY.*(1 + 3*U16 + 9/2*(U16).^2 - 3*U_SQU/2);
FEQ(:,:,17)=t3*DENSITY.*(1 + 3*U17 + 9/2*(U17).^2 - 3*U_SQU/2);
FEQ(:,:,18)=t3*DENSITY.*(1 + 3*U18 + 9/2*(U18).^2 - 3*U_SQU/2);
FEQ(:,:,19)=t3*DENSITY.*(1 + 3*U19 + 9/2*(U19).^2 - 3*U_SQU/2);

```

Condições de passagem a novo instante de tempo ou fim de ciclo

```

F=omega*FEQ+(1-omega)*F;
F(REFLECTED)=BOUNCEDBACK;
prevavu=avu;avu=sum(sum(sum(UX)))/numactivenodes; ts=ts;

```

Código de visualização 3D

```

figure;
BOUND=smooth3(BOUND,'box',3);
[X,Y,Z]=meshgrid(1:nx,1:ny,1:nz)
quiver3(Y,X,Z,UY,UX,UZ)
hold on;
p=isosurface(BOUND,0.5)
P=patch(p)
pp=patch(isocaps(BOUND,0.5))
isonormals(BOUND,P)
view(3);axis vis3d
camlight; lighting phong
xlabel('y')
ylabel('x')
zlabel('z')
title(['3D flow field, after ',num2str(ts),'\deltat']);

```

Código de visualização de um corte

```

figure;
ycut=nx;
colormap(gray(2));
image(BOUND(:,zcut).*1000);
hold on;

```

```

quiverc(1:3:ny,1:3:nx,UY(1:3:nx,1:3:ny,zcut),UX(1:3:nx,1:3:ny,zcut));
colorbar
xlabel('y');
ylabel('x');
title(['Flow field at z=',num2str(zcut),' after ',num2str(ts),'\deltat']);

```

Aplicação da Lei de Darcy para solução numérica

```

u=mean(mean(mean(UX{BOUND==0}))*100;
Niu=((omega)-0.5)/3;
Rho=DENSITY((nx/2)+0.5,(nx/2)+0.5,(nx/2)+0.5);
Miu=Rho*Niu;
a=r*0.001e-2;
L=nx*0.001e-2;
deltaU_out_in_=(mean(mean(UX(nx,,:))-mean(mean(UX(1,,:))))*100;
GradP=(Rho*(deltaU_out_in_)/((L^2-(2.5*pi*a^2))));
K=(u*Miu)/GradP;

```

ANEXO VI - Código para Representação Gráfica da Permeabilidade

Estrutura Cúbica Siples (CS) - Solução Analítica

```
L=51*0.0001;  
a_CS=[24 22 20 16 14 10].*0.0001;  
cmax_CS=pi/6;  
c_CS=((4/3).*pi.*a_CS.^3)./L.^3;  
X_CS=(c_CS./cmax_CS).^(1/3);
```

```
B_CS=1;  
alfan_CS=[1  
1.418649  
2.012564  
2.331523  
2.564809  
2.584787  
2.873609  
3.340163  
3.536763  
3.504092  
3.253622  
2.689757  
2.037769  
1.809341  
1.877347  
1.534685  
0.9034708  
0.2857896  
-0.5512626  
-1.278724  
1.01335  
5.492491  
4.615388  
-0.5736023  
-2.865924  
-4.709215  
-6.870076  
0.1455304  
12.51891  
9.742811  
-5.566269  
]';
```

```

n=0:30;

Cd_CS1=sum(alfan_CS.*X_CS(1,1).^n);
Cd_CS2=sum(alfan_CS.*X_CS(1,2).^n);
Cd_CS3=sum(alfan_CS.*X_CS(1,3).^n);
Cd_CS4=sum(alfan_CS.*X_CS(1,4).^n);
Cd_CS5=sum(alfan_CS.*X_CS(1,5).^n);
Cd_CS6=sum(alfan_CS.*X_CS(1,6).^n);

Cd_CS=[Cd_CS1 Cd_CS2 Cd_CS3 Cd_CS4 Cd_CS5 Cd_CS6];
K_a_CS=L^3./(6.*B_CS.*pi.*a_CS.*Cd_CS);

```

Gráfico de ambas as soluções - CS

```

figure;
Po_CS=1-((4/3.*pi.*(a_CS.^3))./(L.^3));
K_num_CS=[1.8e-007 2.9e-007 4.6e-7 1.0e-6 1.5e-006 3.1016e-006];
semilogy(Po_CS,K_num_CS,'*r',Po_CS,K_a_CS,'-r');
axis([0.2 1 10^-9 10^-5]);

```

Estrutura Cúbica de Corpo Centrado (CCC) - Solução Analítica

```

L=51*0.0001;

a_CCC=[20 18 16 14 12 10 8].*0.0001;
cmax=(sqrt(3)*pi)/8;
c_CCC=((8/3).*pi.*a_CCC.^3)./L.^3;
X_CCC=(c_CCC./cmax).^(1/3);
B_CCC=2;

alfan_CCC=[1 1.575834 2.483254 3.233022 4.022864 4.65032 5.281412 5.826374 6.258376
6.544504 6.878396 7.190839 7.268068 7.304025 7.301217 7.23641 7.298014 7.369849 7.109497
6.228418 5.235796 4.476874 3.541982 2.939353 3.935484 5.179097 3.959872 2.227627 3.39339
4.491369 2.200686];
n=0:30;

Cd_CCC1=sum(alfan_CCC.*X_CCC(1,1).^n);
Cd_CCC2=sum(alfan_CCC.*X_CCC(1,2).^n);
Cd_CCC3=sum(alfan_CCC.*X_CCC(1,3).^n);
Cd_CCC4=sum(alfan_CCC.*X_CCC(1,4).^n);
Cd_CCC5=sum(alfan_CCC.*X_CCC(1,5).^n);
Cd_CCC6=sum(alfan_CCC.*X_CCC(1,6).^n);
Cd_CCC7=sum(alfan_CCC.*X_CCC(1,7).^n);

Cd_CCC=[Cd_CCC1 Cd_CCC2 Cd_CCC3 Cd_CCC4 Cd_CCC5 Cd_CCC6 Cd_CCC7];
K_a_CCC=L^3./(6.*B_CCC.*pi.*a_CCC.*Cd_CCC);

```

Gráfico de ambas as soluções - CCC

```
figure;  
Po_CCC=1-((8/3.*pi.*(a_CCC.^3))./(L.^3));  
K_num_CCC=[3.89e-8 1.05e-7 2.19e-7 4.61e-7 7.07e-7 1.15e-6 1.9e-6];  
semilogy(Po_CCC,K_num_CCC,'ob',Po_CCC,K_a_CCC,'-b');  
axis([0.2 1 10^-9 10^-5]);
```

Estrutura Cúbica de Faces Centradas (CFC) - Solução Analítica

L=51*0.0001;

```
a_CFC=[17 16 15 14 12 10 8 6].*0.0001;  
cmax=(sqrt(2)*pi)/6;  
c_CFC=((16/3).*pi.*a_CFC.^3)./L.^3;  
X_CFC=(c_CFC./cmax).^(1/3);  
B_CFC=4;
```

```
alfan_CFC=[1  
1.620994  
2.62762  
3.518875  
4.503759  
5.354862  
6.240194  
7.048893  
7.778734  
8.380856  
9.093106  
10.04412  
10.99079  
11.76754  
12.34515  
12.61369  
12.71545  
12.95785  
12.97464  
12.59288  
12.33984  
12.58161  
13.10988  
12.51041  
10.89836  
10.59025  
12.10762  
12.21455  
9.163566  
5.1202
```

```
4.523067
```

```
];
```

```
n=0:30;
```

```
Cd_CFC1=sum(alfan_CFC.*X_CFC(1,1).^n);
```

```
Cd_CFC2=sum(alfan_CFC.*X_CFC(1,2).^n);
```

```
Cd_CFC3=sum(alfan_CFC.*X_CFC(1,3).^n);
```

```
Cd_CFC4=sum(alfan_CFC.*X_CFC(1,4).^n);
```

```
Cd_CFC5=sum(alfan_CFC.*X_CFC(1,5).^n);
```

```
Cd_CFC6=sum(alfan_CFC.*X_CFC(1,6).^n);
```

```
Cd_CFC7=sum(alfan_CFC.*X_CFC(1,7).^n);
```

```
Cd_CFC8=sum(alfan_CFC.*X_CFC(1,8).^n);
```

```
Cd_CFC=[Cd_CFC1 Cd_CFC2 Cd_CFC3 Cd_CFC4 Cd_CFC5 Cd_CFC6 Cd_CFC7 Cd_CFC8];
```

```
K_a_CFC=L^3./(6.*B_CFC.*pi.*a_CFC.*Cd_CFC);
```

Gráfico de ambas as soluções - CFC

```
figure;
```

```
Po_CFC=1-((16/3.*pi.*(a_CFC.^3))./(L.^3));
```

```
K_num_CFC=[6.8766e-009 2.08e-8 4.4e-8 7.77e-8 1.90e-7 3.8e-7 7.19e-7 1.32e-6];
```

```
semilogy(Po_CFC,K_num_CFC,'+r',Po_CFC,K_a_CFC,'-r');
```

```
axis([0.2 1 10^-9 10^-5]);
```

Gráfico de ambas as soluções para as 3 estruturas cúbicas

```
figure
```

```
semilogy(Po_CS,K_num_CS,'*k',Po_CS,K_a_CS,'--b',Po_CCC,K_num_CCC,'ok',Po_CCC,K_a_CCC,'-
```

```
.k',Po_CFC,K_num_CFC,'+k',Po_CFC,K_a_CFC,'-r');
```

```
axis([0.2 1 10^-9 10^-5]);
```

```
xlabel('Porosidade');
```

```
ylabel('Permeabilidade m^2');
```

ANEXO VII - Código para Representação Gráfica da Permeabilidade Normalizada

Dimensão L do domínio

$L=51*0.0001;$

Solução numérica CS

$a_CS=[24\ 22\ 20\ 16\ 14\ 10].*0.0001;$

$K_num_CS=[1.8e-007\ 2.9e-007\ 4.6e-7\ 1.0e-6\ 1.5e-006\ 3.1016e-006];$

$Po_CS=1-((4/3.*pi.*(a_CS.^3))./(L.^3));$

$K_num_Ad_CS=K_num_CS./(a_CS.^2);$

Solução numérica CCC

$a_CCC=[20\ 18\ 16\ 14\ 12\ 10\ 8].*0.0001;$

$K_num_CCC=[3.89e-8\ 1.05e-7\ 2.19e-7\ 4.61e-7\ 7.07e-7\ 1.15e-6\ 1.9e-6];$

$Po_CCC=1-((8/3.*pi.*(a_CCC.^3))./(L.^3));$

$K_num_Ad_CCC=K_num_CCC./(a_CCC.^2);$

Solução numérica CFC

$a_CFC=[17\ 16\ 15\ 14\ 12\ 10\ 8\ 6].*0.0001;$

$K_num_CFC=[6.8766e-009\ 2.08e-8\ 4.4e-8\ 7.77e-8\ 1.90e-7\ 3.8e-7\ 7.19e-7\ 1.32e-6];$

$Po_CFC=1-((16/3.*pi.*(a_CFC.^3))./(L.^3));$

$K_num_Ad_CFC=K_num_CFC./(a_CFC.^2);$

Solução analítica CS

$a3=0.0118;$

$a2=-0.2093;$

$a1=2.2648;$

$a0=-4.6254;$

$Fi_CS=log(Po_CS./(1-Po_CS));$

$K_ad_CS=exp(a3.*Fi_CS.^3+a2.*Fi_CS.^2+a1.*Fi_CS+a0);$

Solução analítica CCC

$Fi_CCC=log(Po_CCC./(1-Po_CCC));$

$K_ad_CCC=exp(a3.*Fi_CCC.^3+a2.*Fi_CCC.^2+a1.*Fi_CCC+a0);$

Solução analítica CFC

$Fi_CFC=log(Po_CFC./(1-Po_CFC));$

$K_ad_CFC=exp(a3.*Fi_CFC.^3+a2.*Fi_CFC.^2+a1.*Fi_CFC+a0);$

Gráfico com as várias soluções

```
figure;  
semilogy(Po_CS,K_num_Ad_CS,'*k',Po_CS,K_ad_CS,'--  
b',Po_CCC,K_num_Ad_CCC,'ok',Po_CCC,K_ad_CCC,'-  
.k',Po_CFC,K_num_Ad_CSC,'+k',Po_CFC,K_ad_CFC,'-r');  
axis([0.2 1 10^-4 10^1]);  
xlabel('Porosidade');  
ylabel('Permeabilidade / a^2');
```