

MARCOS ANTONIO DE SOUZA LOURENÇO

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA
COMPUTAÇÃO DE ALTO DESEMPENHO DE
ESCOAMENTOS TRANSIENTES, UTILIZANDO
ESTRUTURA OCTONÁRIA**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA

2012

MARCOS ANTONIO DE SOUZA LOURENÇO

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA
COMPUTAÇÃO DE ALTO DESEMPENHO DE ESCOAMENTOS
TRANSIENTES, UTILIZANDO ESTRUTURA OCTONÁRIA**

Tese apresentada ao Programa de Pós-graduação
em Engenharia Mecânica da Universidade Federal de
Uberlândia, como parte dos requisitos para a obtenção
do título de **DOCTOR EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Fluidos.

Orientador: Prof. Dr. Elie Luis Martínez Padilla

Co-orientador: Prof. Dr. Aristeu da Silveira Neto

Uberlândia - MG

2012

"Understanding is more often used to try to alter an outcome than to repeat or perpetuate it"

Jared Diamond, em Guns, Germs and Steel

AGRADECIMENTOS

À Deus;

À toda minha família;

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de realizar este trabalho;

Aos professores Prof. Dr. Elie L. M. Padilla e Prof. Dr. Aristeu da Silveira Neto pela orientação, apoio, interesse, respeito, flexibilidade e principalmente paciência, demonstrados no decorrer de todo o trabalho;

Ao meu amigo Rafael;

Às pesquisadoras Millena e Catalina, pela atenção, apoio e colaboração no presente trabalho;

Aos colegas do Laboratório do MFLab e da FEMEC;

Ao Laboratório de Mecânica dos Fluidos - MFLab - e ao Departamento da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia - FEMEC - por todo o suporte oferecido;

Ao CNPq #140168/2009-5 e à PETROBRAS pelo apoio financeiro;

LOURENÇO, M. A. S., 2012, *Desenvolvimento de uma plataforma para computação de alto desempenho de escoamentos transientes, utilizando estrutura octonária*. 132 f. Tese (Doutorado), — UFU, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2012.

Resumo

Apresenta-se, neste trabalho, o desenvolvimento e implementação de uma estrutura de dados paralela (memória distribuída), octonária, diretamente vinculada a uma malha cartesiana e adaptativa, para solução das equações de Navier-Stokes. As equações foram discretizadas espacialmente utilizando o Método de Volumes Finitos, com os termos difusivos aproximados por diferenças-centradas. A discretização temporal foi realizada por meio do método Adams-Bashforth que, assim como o método de diferenças centradas, possui segunda-ordem de exatidão. A metodologia de fronteira imersa foi utilizada para representação de corpos sólidos no escoamento e o acoplamento pressão-velocidades é feito com o método de passos-fracionados de dois passos. Resultados são apresentados tanto para a malha octonária adaptativa quanto para a uma malha retilínea, ambas em paralelo. Em ambas as implementações foram utilizadas bibliotecas externas para resolver o sistema linear resultante da equação de Poisson para correção da pressão no método de passos-fracionados. Os resultados determinados foram comparados com dados numéricos e experimentais, apresentando boa concordância.

Palavras-chave: Método de volumes finitos; Escoamento de Taylor-Couette; Adaptatividade de malhas; Equações de Navier-Stokes.

LOURENÇO, M. A. S., 2012, *Desenvolvimento de uma plataforma para computação de alto desempenho de escoamentos transientes, utilizando estrutura octonária*. 132 f. Tese (Doutorado), — UFU, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2012.

Abstract

The present work describes the development of a parallel distributed-memory implementation, of an octree data structure, linked to an adaptive Cartesian mesh to solve the Navier-Stokes equations. The Finite Volume Method was used in the spatial discretization where the diffusive terms approximated by the central differences method. The temporal discretization was accomplished using the Adams-Bashforth method. Both temporal and spatial discretizations are of second-order of accuracy. The immersed boundary method was employed to represent immersed solid bodies in the flow and the Velocity-pressure coupling is done using the fractional-step method of two steps. Results, for sequential and parallel simulations, are presented both for the adaptive octree mesh and for a rectilinear mesh. Moreover, all simulated results were obtained using an external solver for the Poisson equation, from the pressure correction, in the fractional-step method. The numerical results are compared with numerical and experimental data, and the agreement between the two is found to be good.

Keywords: Finite volume method; Taylor-Couette flow; Adaptive mesh refinement; Navier-Stokes equations.

Sumário

Lista de Figuras	x
Lista de Tabelas	xix
Lista de Siglas	xx
Lista de símbolos	xxi
1 Introdução	1
1.1 Objetivos	6
2 Malhas computacionais e discretização.....	7
2.1 Discretização do domínio de solução	7
2.2 Balanço de carga	10
2.3 Estrutura de dados paralela <i>Octree</i>	11
2.3.1 Balanceamento de carga	14
2.3.2 Estrutura interna da malha	16
2.4 Método da Fronteira Imersa	20
3 Revisão Bibliográfica.....	22
3.1 Adaptatividade com malhas cartesianas	22
3.2 Paralelismo e malhas cartesianas	24
3.3 Metodologia de Fronteira Imersa	27
4 Metodologia	31

4.1	Modelagem Matemática	31
4.1.1	Método de Fronteira Imersa	31
4.2	Modelagem Numérica	33
4.2.1	Discretização espacial	33
4.2.2	Discretização temporal	35
4.2.3	Metodologia de Fronteira Imersa	37
4.3	Solução do Sistema de Equações Lineares	45
4.3.1	Solução de Poisson por níveis de malha	46
4.3.2	Solução de Poisson global	49
4.4	Restrição e prolongamento em variáveis desencontradas	53
4.5	Paralelização	56
4.5.1	Estrutura do aplicativo	61
4.5.2	Critérios de refinamento	62
5	Resultados.....	64
5.1	Testes de convergência	67
5.2	Cavidade com tampa deslizante	70
5.3	Jato incompressível laminar	82
5.4	Escoamento de Taylor-Couette	86
6	Conclusões e Perspectivas	91
	Referências	93
	Apêndice A – Paralelização do aplicativo com malha retilínea	103
A.1	Discretização espacial	103
A.2	Solução do Sistema de Equações Lineares	104
A.3	Paralelização	104

A.3.1	Balanço de carga com Zoltan	104
-------	-----------------------------------	-----

Apêndice B – Resultados para o caso com malha retilínea	106
--	------------

B.1	Cavidade quadrada com tampa deslizante	106
-----	--	-----

B.2	Escoamento de Taylor-Couette	110
-----	------------------------------------	-----

B.3	Aspectos computacionais	115
-----	-------------------------------	-----

Lista de Figuras

Figura 1.1	Esboços de Leonardo da Vinci mostrando estruturas de difícil visualização à olho nu. (MACCURDY, 1938)	1
Figura 1.2	Malhas adaptativas. (a) Malha não-estruturada composta de elementos triangulares, na simulação do escoamento supersônico ($M_{inf} = 4$) ao redor da versão simplificada do VLS (Veículo Lançador de Satélites). (b) Malha quadtree utilizada para simular um escoamento oceanográfico.	3
Figura 1.3	Cluster de computadores: (a) Cluster Beowulf na NASA, com 128 processadores conectado por uma rede Ethernet de 100Mb/seg e ; (b) cluster com 372 GPUs e desempenho de mais de 114 teraflops. GPUs são conhecidas por gerar mais processamento por potência consumida e com menor sobreaquecimento do sistema.	4
Figura 1.4	Riser de produção em configuração de catenária simples. D representa o diâmetro externo do riser e h , muito maior que D , é a distância entre a plataforma e o contato com o solo ou TDP (<i>Touch Down Point</i>).	5
Figura 2.1	Exemplos de discretização. (a) utilizando malha estruturada e parametrizada e (b) não-estruturada com elementos retangulares e triangulares.	8
Figura 2.2	Malha cartesiana refinada apenas em uma pequena porção do domínio. (a) Malha cartesiana clássica com adição de linhas próxima à região de interesse; (b) baseada em blocos sobrepostos; e (c) com estrutura quadtree.	9
Figura 2.3	Exemplos de curvas de preenchimento de (a) Morton (Lebesgue) e de (b) Hil-	

bert.	12
Figura 2.4 Exemplo de uma malha quadtree e sua representação na forma de uma árvore.	13
Figura 2.5 Distribuição de volumes entre dois processos utilizando a codificação de Morton.	14
Figura 2.6 Domínio distribuído entre 3 processos utilizando curvas de preenchimento de (a) Morton e (b) Hilbert.	16
Figura 2.7 Malha 3x3 refinada dentro da estrutura quadtree.	16
Figura 2.8 Distribuição das variáveis na malha. (a) mapeamento e (b) posicionamento das células fantasmas: (<i>azul</i>) condições de contorno; (<i>vermelho</i>) fantasmas da malha fina; (<i>verde</i>) fantasmas da malha grossa.	18
Figura 2.9 Restrição para o refinamento de malha.	18
Figura 2.10 Esquema ilustrando as malhas euleriana e os pontos lagrangianos no método de fronteira imersa.	21
Figura 4.1 Distribuição da força lagrangiana para os volumes adjacentes (sombreados).	32
Figura 4.2 Volume de controle elementar ilustrando (a) os fluxos e (b) as posições definidas com letras minúsculas para os valores que se encontram nas faces e maiúsculas para aqueles no centro dos volumes.	34
Figura 4.3 Funções de distribuição. (a) Função distribuição do tipo gaussiana proposta por Juric e Tryggvason (1996) e (b) função triangular, também conhecida como função chapéu.	40

Figura 4.4	Diagrama esquemático para ilustrar o método MLS.	41
Figura 4.5	Localização dos pontos e do estêncil das interpolações para cálculo do Laplaciano de ϕ , na interface fina-grossa, em duas dimensões.	49
Figura 4.6	Configurações da malha para variáveis centradas. Quando não se dispõe do estêncil padrão, é necessário um conjunto diferente de constantes de interpolação para células que necessitam de uma interpolação diferenciada. Dois casos são ilustrados no exemplo: (a) quadrática deslocada e (b) linear.	52
Figura 4.7	Componentes de velocidade u , na interface fina-grossa.	53
Figura 4.8	Disposição das componentes de velocidade em uma malha bidimensional. As setas amarelas denotam as variáveis que precisam ser resolvidas simultaneamente.	55
Figura 4.9	Atualização feita por simples injeção de células fantasmas, destacadas na figura, para a decomposição da malha cartesiana em dois domínios (ou processadores) $P1$ e $P2$	57
Figura 4.10	Numeração no mapa local e operação de importação e exportação de dados entre dois processos. Um mapa é utilizado para ambas as operações no mapa local, que envolve todas as células	58
Figura 4.11	Casca esférica não-estruturada, composta por elementos triangulares e particionada em dois domínios. A linha tracejada situa-se na fronteira entre os domínios.	59
Figura 4.12	Malha com um nível de refinamento, mapeada em dois processadores.	60

Figura 4.13 Diagrama simplificado das ligações entre as principais funções do aplicativo. <i>costruct.so</i> é a biblioteca de malha cartesiana estruturada por uma octree. <i>Solver</i> é a extensão para resolver a equação de Poisson. <i>Trilinos</i> é a biblioteca contendo os pacotes utilizados no aplicativo. Aplicação pode ser um arquivo do tipo <i>main.c</i> que gerencia a simulação.	62
Figura 5.1 Corte bidimensional na posição $x = 0,75$, mostrando os resultados para a Eq. (5.1). Além do nível base, a malha possui mais 3 níveis de refinamento. Nota-se que os valores são da ordem do ϵ para os decimais de dupla precisão utilizados.	65
Figura 5.2 Duas maneiras de ordenação para os índices globais do mapa para o sistema de equações lineares: (a) Mapeamento por agrupamentos e; (b) por blocos 2×2	65
Figura 5.3 Diagrama apontando os termos não nulos na matriz: (a) utilizando ordenamento baseado nos agrupamentos de blocos e; (b) folha a folha, seguindo a curva de preenchimento de Morton.	66
Figura 5.4 Resultados para a Eq. (5.3): (a) Resolução dos contornos para os diferentes níveis de malha e (b) Distribuição do erro absoluto.	68
Figura 5.5 Testes na malha adaptativa: (a) Distribuição dos blocos em 11 processadores, para uma configuração genérica de refinamento. Detalhe da solução da pressão, referente à Eq (5.7), mostrada em uma das diferentes configurações testadas no trabalho.	69
Figura 5.6 Diagrama esquemático para a cavidade quadrada com tampa deslizante.	70
Figura 5.7 Problema da cavidade com tampa deslizante a $Re = 1000$ com soluções por <i>AMG</i> e <i>BiCGStab</i> : (a) Ganho em processamento (<i>speedup</i>) em função do número de processadores e curva ideal de ganho para até 32 processadores e; (b) Tempo gasto em 25 iterações de solução.	72

Figura 5.8	Comparação dos perfis das componentes de velocidade determinados no presente trabalho, com aqueles de Ku, Hirsh e Taylor (1987), Deshpande e Milton (1998) a $Re = 1000$: (a) Componente u sobre a linha vertical em y e passando pelo ponto $(x/L = 0,5; z/L = 0,5)$ e; (b) Componente v sobre a linha horizontal na direção x , cruzando o ponto $(y/L = 0,5; z/L = 0,5)$	73
Figura 5.9	Malha computacional empregada na solução do problema de cavidade quadrada com tampa deslizante a $Re = 10^3$: (a) Visão externa mostrando detalhes das regiões refinadas e; (b) corte no plano $x - y$, em $(z/L = 0,5)$	74
Figura 5.10	Perfis com a variação das componentes de velocidade u e v ao longo das linhas de centro vertical e horizontal, respectivamente: (a) e (b) $t = 6s$, (c) e (d) $t = 8s$ e (e) e (f) $t = 12s$	75
Figura 5.11	Resultados para cavidade já em regime permanente, no instante $t = 30s$, $\Gamma_{ldc} = 2$ e $Re = 10^3$: (a) Contornos de pressão em um corte no plano $x - y$, em $(z/L = 0,5)$ e; (b) corte mostrando detalhes da malha juntamente com os contornos do módulo da velocidade $ \mathbf{U} $, nos planos centrais nas três direções coordenadas.	76
Figura 5.12	Configurações de malha após refinamento adaptativo para os instantes de $t = 2s$, $t = 4s$, $t = 6s$, $t = 8s$, $t = 10s$ e $t = 12s$ no plano $x - y$, em $z = 1$. Os contornos são relativos ao módulo da velocidade para os valores $ \mathbf{U} = 0,1m/s$ e $ \mathbf{U} = 0,2m/s$	77
Figura 5.13	Cortes no plano $x - y$, em $z = 1$ para o problema da cavidade a $Re = 10^3$ e $\Gamma_{ldc} = 2$ para os instantes de tempo $t = 2s$, $t = 4s$, $t = 6s$, $t = 8s$, $t = 10s$ e $t = 12s$. Na coluna direita são mostrados os resultados experimentais de Migeon, Texier e Pineau (2000) e as linhas de corrente determinadas no presente trabalho são mostradas na coluna da esquerda.	79
Figura 5.14	Posição do centro do vórtice primário para os instantes de tempo de $t = 2s$ à $t = 12s$	81

- Figura 5.15 Corte no plano correspondente a $\frac{y}{L} = 0,5$ no escoamento em cavidade com $Re = 10^3$ e $\Gamma = 2$ evidenciando os vórtices de canto e a fase inicial de manifestações de estruturas do tipo *Taylor* no instante $t = 12s$: (a) Linhas de corrente determinadas no presente trabalho e; (b) Dedução do escoamento feita com base em dados experimentais de Migeon, Pineau e Texier (2003). 81
- Figura 5.16 Corte no plano correspondente a $y/L = 0,75$ no escoamento em cavidade com $Re = 10^3$ e $\Gamma = 2$ no instante $t = 8s$: (a) Linhas de corrente determinadas no presente trabalho e; (b) Dedução da configuração do escoamento feita com base em dados experimentais de Migeon, Pineau e Texier (2003). 82
- Figura 5.17 Diagrama esquemático para o problema de jato incompressível laminar. 83
- Figura 5.18 Curva de decaimento para a componente vertical de velocidades v na linha de centro na direção y 84
- Figura 5.19 Configurações de malha após refinamento adaptativo para os instantes de $t = 0,05s$, $t = 1,04s$, $t = 2,64s$ e $t = 7,5s$ no plano $x - y$, em $z = 0,08$. Os valores são para a componente de velocidade v . O nível mais fino não é mostrado para uma maior clareza dos resultados. 85
- Figura 5.20 Configuração dos blocos de malha relativos aos processadores de número 8 à 16, nos instantes (a) $t = 0,05s$ e (b) $t = 8,01s$. As cores para cada bloco são as mesmas nas duas figuras. 86
- Figura 5.21 Representação esquemática do sistema de cilindros concêntricos para o escoamento de Taylor-Couette. O cilindro interno, de raio R_i possui velocidade angular ω_i 87
- Figura 5.22 Escoamento de Taylor-Couette na malha adaptativa: (a) Contornos da componente $w = -0,1m/s$ e $w = 0,1m/s$, calculados utilizando 4 processadores e; (b)

Corte mostrando detalhes das malhas lagrangiana (linhas horizontais tracejadas) e euleriana (somente o mais fino de 3 níveis).	88
Figura 5.23 Detalhes e influência da malha adaptativa no cálculo dos contornos da componente de velocidade $w = \pm 0,1$.	89
Figura 5.24 Variação de velocidade axial adimensional $w/(R_i\omega)$ na direção radial para $Ta = 100$. Os valores são referentes à linha passando pelo núcleo de um dos vórtices.	90
Figura 5.25 Componente de velocidade radial adimensional $u_r/(R_i\omega)$ sobre a linha axial passando pelo centro dos vórtices para $Ta = 100$. Os valores são tomados no intervalo entre dois vórtices contrarotativos.	90
Figura A.1 Esquema para discretização ponderada dos termos advectivos por diferenças centradas.	103
Figura A.2 Exemplo de particionamento de uma malha cartesiana bidimensional. Os pesos são definidos para cada volume para os casos de (a) balanceamento não ponderado e (b) ponderado.	105
Figura B.1 Perfis da componente de velocidade u retirado do centro do cubo, passando pelo ponto $(x/L = 0,5; z/L = 0,5)$. As comparações são feitas para $Re = 100$ e $Re = 400$.	107
Figura B.2 Perfis da componente de velocidade v retirado do centro do cubo, passando pelo ponto $(y/L = 0,5; z/L = 0,5)$. As comparações são feitas para $Re = 100$ e $Re = 400$.	108
Figura B.3 Comparação dos perfis das componentes de velocidade determinados no presente trabalho, com aqueles de Ku, Hirsh e Taylor (1987), Deshpande e Milton (1998) a $Re = 1000$: (a) Componente u sobre a linha vertical em y e passando	

pelo ponto $(x/L = 0,5; z/L = 0,5)$ e; (b) Componente v sobre a linha horizontal na direção x , cruzando o ponto $(y/L = 0,5; z/L = 0,5)$ 109

Figura B.4 Partição do domínio lagrangiano: (a) Casca cilíndrica imersa na malha euleriana de modo particionado em 4 processadores e; (b) Corte mostrando os contornos da componente w entre os blocos distribuídos entre 12 processadores. 110

Figura B.5 Variação de velocidade axial adimensional $w/(R_i\omega)$ na direção radial para $Ta = 100$. Os valores são referentes à linha passando pelo núcleo de um dos vórtices. 111

Figura B.6 Componente de velocidade radial adimensional $u_r/(R_i\omega)$ sobre a linha axial passando pelo centro dos vórtices para $Ta = 100$. Os valores são tomados no intervalo entre dois vórtices contrarotativos. 112

Figura B.7 Escoamento de Taylor-Couette a $Ta = 100$ e $\Gamma = 36$: (a) particionamento em 16 processadores utilizando o método *RCB* e; (b) Visão aproximada, mostrando detalhes da malha lagrangiana, de elementos triangulares e com nós não coincidentes com os volumes eulerianos. 113

Figura B.8 Formação dos vórtices de Taylor-Couette para $Ta = 100$ e razão de aspecto $\Gamma = 36$. A assimetria na evolução dos vórtices na simulação sequencial, é igualmente reproduzida em paralelo. 114

Figura B.9 Vetores de velocidade em um corte transversal na direção azimutal θ para o escoamento a $Ta = 100$ completamente desenvolvido. São mostradas também as isolinhas da componente de velocidade azimutal: (a) corte realizado no plano $r - z$ do código em coordenadas cilíndricas e no plano $y - z$ da malha euleriana e fronteira imersa com os métodos (b) MFV e (c) MLS. 115

Figura B.10 Balanço de carga na solução paralela do problema de cavidade com tampa deslizante a $Re = 400$. Observa-se, nos valores normalizados entre 0 e 1, que o

desbalanceamento máximo é de 5% entre as 64 partições. 117

Figura B.11 Ganho em processamento (*speedup*) em função do número de processadores e curva ideal de ganho, para o problema de cavidade com tampa deslizante a $Re = 1000$ 118

Lista de Tabelas

Tabela 2.3.1Tabela de ordenação e de orientação para converter da curva de Morton para a curva de Hilbert em uma malha bidimensional.	15
Tabela 5.1.1Erros calculados com base na norma $L2$ para a equação de Poisson, com termo fonte a partir da equação manufaturada Eq. (5.3).	68
Tabela 5.1.2Convergência dos erros, baseados na norma $L2$, para as equações de Navier-Stokes.	69
Tabela B.3.1Erros calculados com base na norma $L2$ para a equação de Poisson, com termo fonte a partir da equação manufaturada Eq. (5.3). Para as quatro resoluções da malha octree, com um nível de refinamento, os valores são apresentados para os casos sequencial e paralelo.	116

Lista de Siglas

CFD	Dinâmica dos fluidos computacional (<i>Computational Fluid Dynamics</i>)
FDM	Método de diferenças finitas (<i>Finite Difference Method</i>)
FEM	Método de elementos finitos (<i>Finite Element Method</i>)
FVM	Método de volumes finitos (<i>Finite Volume Method</i>)
MPI	Padrão de troca paralela de mensagens (<i>Message Passing Interface</i>)
GPU	Unidade de processamento gráfico (<i>Graphical Processing Units</i>)
VIV	Vibração induzida por vórtices (<i>Vortex Induced Vibration</i>)
AMR	Refinamento adaptativo de malha (<i>Adaptive Mesh Refinement</i>)
SFC	Curva de preenchimento espacial (<i>Space Filling Curve</i>)
IB	Métodologia de Fronteira Imersa (<i>Immersed Boundary method</i>)
MLS	Método de mínimos quadrados móveis (<i>Moving Least Squares Method</i>)
MFV	Modelo Físico Virtual
MG	Método de multigrid
CFL	Critério de estabilidade de <i>Courant-Friedrichs-Lewy</i>
SIP	Strongly Implicit Procedure
PIV	Método de velocimetria por imagem de partículas (<i>Particle image velocimetry</i>)
AMG	Multigrid algébrico (<i>Algebraic Multi-Grid</i>)

Lista de símbolos

∂	Derivada parcial
\mathbf{G}^T	Matriz ou vetor \mathbf{G} tranposto
\mathbf{x}_k	Coordenadas de um ponto lagrangiano k
E	Espaçamento anular (Escoamento de Taylor Couette)
h	Comprimento característico dos elementos da malha lagrangiana
L_z	Comprimento axial dos cilindros (Escoamento de Taylor Couette)
p	Pressão
p'	Correção de pressão
R	Raio dos cilindros (Escoamento de Taylor Couette)
Re	Número de Reynolds
t	Tempo
Ta	Número de Taylor
H	Domínio de influência para um ponto lagrangiano no método <i>MLS</i>
RHS	Termo do lado direito (<i>Right Hand Side</i>)
δ	Função Delta de Dirac
δx	Espaçamento entre os centros de dois volumes na direção x
η	Razão entre raios
ν	Viscosidade cinemática
ω	Velocidade angular
ρ	Massa específica
Γ	Razão de aspecto
ΔV	Volume de controle elementar
Δt	Passo de tempo discreto
Δx	Espaçamento da malha computacional na direção x
Δy	Espaçamento da malha computacional na direção y
Δz	Espaçamento da malha computacional na direção z
V_k	Volume de um elemento na malha lagrangiana

1 Introdução

Os fenômenos físicos envolvendo escoamentos de fluidos sempre foram de fundamental importância para manutenção e desenvolvimento das atividades cotidianas do ser humano. Modelos simples de sistemas de irrigação já eram desenvolvidos na antiguidade e atualmente a tecnologia nesta área já permite façanhas como as precisas e sofisticadas previsões climáticas, otimizações aerodinâmica e hidrodinâmica de veículos e dos mais diversos processos industriais, entre outras. Estudos já eram feitos pelos matemáticos gregos como Archimedes (287-212 AC) e mais tarde por Leonardo da Vinci (1452-1519). Leonardo criou figuras muito representativas de fenômenos físicos em escoamentos que seriam visualizadas somente séculos depois por técnicas experimentais avançadas. Esses esboços podem ser encontradas em (MACCURITY, 1938), como mostrado na Fig. 1.1.



Figura 1.1: Esboços de Leonardo da Vinci mostrando estruturas de difícil visualização à olho nu. (MACCURITY, 1938)

Somente muito tempo depois, um grande número de pesquisadores estudaram as relações entre a reação do fluido em resposta às forças aplicadas neste, dando origem às equações

de Navier-Stokes (1822 e 1845). Este conjunto de equações em sua forma completa se apresenta de forma complexa, de tal forma que os primeiros resultados foram determinados somente a partir de simplificações e de aproximações para casos específicos.

Com o advento dos computadores digitais, resultados obtidos a partir de aproximações numéricas de um modelo matemático contribuíram muito para a solução das equações que descrevem os fenômenos físicos, o que acabou se tornando de extrema importância para a pesquisa científica e para a indústria moderna. As equações de Navier-Stokes agora podiam ser resolvidas com um melhor grau de fidelidade, o que também contribuiu para o desenvolvimento de novos modelos matemáticos. Todo esse conjunto de métodos numéricos aplicados a um modelo matemático representando um fenômeno físico em mecânica dos fluidos ficou internacionalmente conhecido por Dinâmica dos Fluidos Computacional (CFD).

Numericamente, um problema contínuo pode ser discretizado de tal maneira que as suas variáveis dependentes são consideradas somente em pontos discretos ou em regiões fixas no espaço. O conjunto espacial de pontos ou volumes que preenchem o domínio analisado constituem a *malha computacional*. Dessa forma, um problema envolvendo cálculos em um determinado domínio físico se torna um problema algébrico calculado em regiões discretas (embora contínuas por partes) de uma malha. Por exemplo, derivadas em uma equação diferencial parcial podem ser aproximadas por diferenças simples, uma representação algébrica calculada entre pontos próximos na malha.

Três métodos de discretização espacial muito utilizados podem ser citados: o método de diferenças finitas (FDM), de elementos finitos (FEM) e de volumes finitos (FVM). Uma comparação entre as diferenças, semelhanças e as principais características desses três métodos podem ser encontradas em (BREZINSKI, 2005). Ao utilizar um ou outro método algumas questões devem ser levadas em conta. Entre elas estão a complexidade geométrica do problema, aplicação das condições de contorno, exatidão do método, conservabilidade e facilidade de paralelização.

Qualquer que seja o método de discretização utilizado, a acurácia alcançada na simulação de determinado problema é, muitas vezes, diretamente dependente da qualidade da malha. Essa qualidade, pode ser interpretada tanto como o tipo e a ordem de seus elementos em uma solução por FEM, como a capacidade de se obter alta densidade de volumes ou nós em locais estratégicos nos métodos de FDM e FVM. O papel desempenhado pela malha na solução numérica de determinado problema é tão importante, que nos códigos modernos o refinamento e/ou melhoramento da malha não é aplicada em todo o domínio e devem ser adaptados de acordo com as necessidades do problema a ser tratado (LÖHNER, 1987). Exemplos de malhas

numéricas com refinamento adaptativo são apresentados na Fig. 1.2.

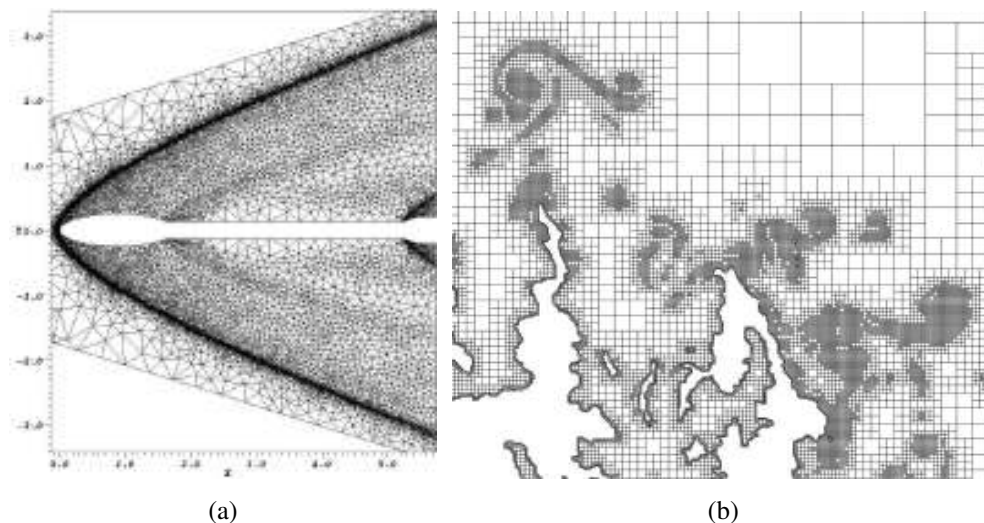


Figura 1.2: Malhas adaptativas. (a) Malha não-estruturada composta de elementos triangulares, na simulação do escoamento supersônico ($M_{inf} = 4$) ao redor da versão simplificada do VLS (Veículo Lançador de Satélites). (b) Malha quadtree utilizada para simular um escoamento oceanográfico.

Fontes: (LOURENÇO, 2007) e (POPINET, 2003) respectivamente.

Isto resulta em economia de armazenamento em memória e de processamento computacional que, aliada ao desenvolvimento de algoritmos de alto desempenho, tem contribuído para a solução de problemas cada vez mais complexos. Essa nova disponibilidade de recursos tem elevado o nível de dificuldade e tamanho característico dos problemas atuais ao ponto que, mesmo com a evolução quase exponencial dos processadores de acordo com a Lei de Moore (PARHAMI, 2002), a utilização de *clusters* de computadores seja cada vez mais necessária para os cálculos de grande porte a serem empregados nas simulações (SOUZA; SOUZA; SILVEIRA NETO, 2006). Atualmente, o paradigma comum de implementação de programação paralela através da passagem de mensagens é o padrão MPI (ou *Message Passing Interface*), utilizado em Campregher (2005). Clusters modernos também já dispõem do processamento adicional de placas do tipo GPU, que são centrais de processamento que antigamente eram somente utilizadas para processamento gráfico. Um exemplo desse tipo de arquitetura é o *CUDA*, ou *Computer Unified Device Architecture*, da NVIDIA, com bons resultados demonstrados na literatura, como em (TöLKE; KRAFCZYK, 2008).

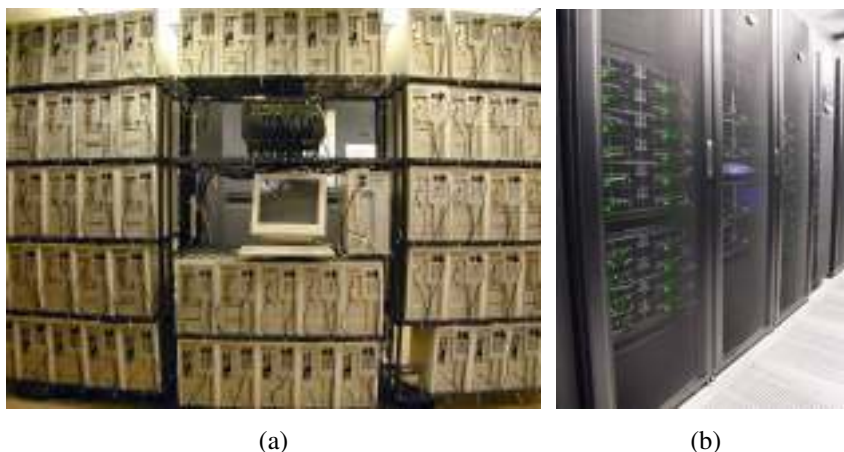


Figura 1.3: Cluster de computadores: (a) Cluster Beowulf na NASA, com 128 processadores conectado por uma rede Ethernet de 100Mb/seg e ; (b) cluster com 372 GPUs e desempenho de mais de 114 teraflops. GPUs são conhecidas por gerar mais processamento por potência consumida e com menor sobreaquecimento do sistema.

Fontes: Nvidia (<http://gfxspeak.com>) e <http://www.cse.mtu.edu> respectivamente.

No final da década de 1960 o Brasil descobriu que a maior parte de suas reservas petrolíferas estava localizada no mar, e não em terra, como acontecia nos demais países. A exploração do mineral se dava em bacias territoriais e era baseada na tecnologia *onshore*, desenvolvida pelos EUA (FREEMAN; SOETE, 1997). Mas a profundidade média dos poços brasileiros eram superiores às dos poços americanos. Diante da ausência de tecnologia internacional e da importância estratégica dos recursos naturais do país, decidiu-se criar uma ciência própria de extração de petróleo em alto mar.

Quase meio século depois, a descoberta do pré-sal coloca o Brasil em uma nova posição no cenário da indústria mundial de petróleo. Todo esse crescimento deve-se principalmente à pesquisa de exploração em campos de águas profundas e ultraprofundas (superiores a 1000m). Um dos desafios impostos para a instalação de plataformas petrolíferas em grandes profundidades é a utilização econômica e segura de seus componentes estruturais, como por exemplo os dutos para perfuração e produção, mais tecnicamente chamados *risers*. Neste trabalho a expressão *riser* será utilizada, devido à disseminação do termo na literatura, e denotará os tubos amplamente utilizados para o transporte de fluidos entre o fundo do mar e as unidades de produção.

Devido às severas condições ambientais e às condições dinâmicas, os *risers* podem ter seu comportamento afetado pelo grande número de solicitações variáveis à que estão sujeitos e, nesse caso, são considerados como uma parte crítica em um sistema de perfuração ou produção *offshore* (PINHO, 2001). Em ambas as formas, rígida ou flexível, os *risers* são componentes

de alto custo, não só de aquisição e manutenção, como também para recuperação do ambiente marinho caso ocorra alguma falha no sistema, o que também poderia levar à um desastre ecológico. Uma das causas que poderia acarretar em um colapso da tubulação seria a amplificação dinâmica da resposta em termos dos deslocamentos e variações de tensões, provocada quando as frequências naturais de vibração do riser são muito próximas daquelas presentes no ambiente circundante.

A previsão de fenômenos característicos relacionados a esse tipo de interação fluido-estrutura é de grande importância para a indústria de petróleo e envolve diferentes áreas da engenharia mecânica. Tecnicamente conhecida por VIV¹, a vibração induzida pelo desprendimento de vórtices é uma das causas mais frequentes de falhas em estruturas imersas em um meio fluido (FALTSCHART et al., 2000). As soluções para minimizar as vibrações em risers para exploração de petróleo são, em sua maioria, extremamente onerosas com uma alta relação custo benefício, chegando à cifra de 4 milhões de dólares por riser instalado (ÁSSI, 2005). Neste caso, analisar a natureza desses fenômenos e propor soluções viáveis para aumentar a confiabilidade e durabilidade do conjunto de estruturas em questão, é fundamental sob o ponto de vista econômico. A Fig. 1.4 ilustra a configuração simplificada de um riser de extração de petróleo. O riser é conectado à plataforma e sua outra extremidade na saída do fluido do poço (geralmente de um componente chamado *árvore de natal*).

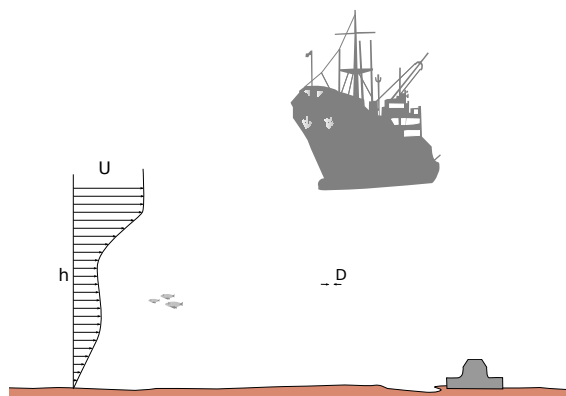


Figura 1.4: Riser de produção em configuração de catenária simples. D representa o diâmetro externo do riser e h , muito maior que D , é a distância entre a plataforma e o contato com o solo ou *TDP* (*Touch Down Point*).

A Fig. 1.4 é meramente ilustrativa e não representa bem a real razão de aspecto $\Gamma \approx \frac{h}{D}$ que, para uma lâmina d'água de $h = 1000m$ é da ordem de $\Gamma = 2500$. Para este problema em especial, a dificuldade na determinação de resultados se deve ao forte acoplamento fluido es-

¹Do inglês *Vortex Induced Vibration*.

trutural, às grandes razões de aspectos encontradas e à escassa fonte de dados experimentais, conforme Lopes (2006). Soluções numéricas existem, embora simplificadas devido à grande razão de aspecto ou ainda, são aproximadas na forma de um conjunto de soluções de fatias bidimensionais compondo o problema tridimensional (WILLDEN; GRAHAM, 2004; FLATS-CHART; MENEGHINI, 2004).

A parte externa de um riser já oferece um grande desafio para ser modelada em engenharia. Entretanto, na parte interna, a dinâmica do escoamento também é complexa, devido principalmente aos fluxos de entrada de fluido refrigerante-lubrificante e da saída de fluido com detritos quando em um riser de perfuração. Esse sistema envolve o escoamento de entrada no interior de uma tubulação e de saída ao redor desta tubulação mas dentro de outro cilindro maior, em um arranjo de cilindros concêntricos. Toda a dinâmica do cilindro interno envolve uma interação do tipo fluido-estrutura, que são de interesse da engenharia que trata de problemas de colunas de perfuração.

1.1 Objetivos

O presente trabalho possui como objetivo apresentar a construção de uma ferramenta numérica para simular, de forma paralela, escoamentos viscosos e incompressíveis, utilizando as equações de Navier-Stokes. Será empregado um paralelismo do tipo por partição de domínio, com a distribuição e gerenciamento de tarefas através do paradigma *MPI*. Para o particionamento será empregada uma Curva de Preenchimento Espacial (*SFC*), aplicada em uma estrutura de dados *octree*. Essa estrutura será utilizada para manipular e realizar as operações matemáticas nas células de uma malha semi-estruturada e com possibilidade de refinamento adaptativo, a ser utilizada com uma discretização espacial por volumes finitos. A equação de Poisson, cuja solução se apresenta como o maior gargalo em relação ao tempo total de simulação, será resolvida utilizando uma biblioteca externa. Para isso, a plataforma desenvolvida deve possuir os recursos de mapeamento e de interpolação necessários, de forma que qualquer *solver* genérico possa ser empregado para resolver o problema em questão, o que constitui a maior contribuição da tese. Além disso pretende-se utilizar a metodologia de fronteira imersa. Uma vez que o refinamento adaptativo é baseado em blocos de malha cartesiana, essa talvez seja uma das melhores maneiras para representar corpos de geometria complexa imersos no escoamento.

2 Malhas computacionais e discretização

Grande parte dos fenômenos físicos abordados na área de mecânica dos fluidos, caracteriza-se pela presença de múltiplas escalas no espaço e no tempo. Muitas vezes as pequenas escalas se localizam apenas em regiões específicas e de exígua extensão, quando comparadas ao domínio total de solução. Esse é o caso, por exemplo, do escoamento nas proximidades de uma parede, onde os efeitos viscosos são, muitas vezes, mais importantes do que aqueles que ocorrem no restante do domínio. Uma das maneiras mais econômicas e utilizadas para a determinação de resultados em tais problemas envolve a realização de um refinamento local nos pontos de interesse, realizando um refinamento adaptativo de malha, mais tecnicamente conhecido pela sigla em língua inglesa de *Adaptive Mesh Refinement* (AMR). Atualmente, existem arquiteturas computacionais com vários núcleos de processamento e memória distribuída, o que se agrega ao refinamento de malha para simular problemas de grande porte. É claro que o desempenho em tais sistemas é altamente dependente da distribuição dos nós/volumes para que diferentes processadores possam realizar os cálculos de maneira quase simultânea. Dessa forma, uma vez que os tipos de malha e de discretização influenciam diretamente no tempo e qualidade da simulação, é apresentada nesse capítulo a metodologia de construção da estrutura da malha computacional utilizada neste trabalho.

2.1 Discretização do domínio de solução

Embora exista atualmente um grande número de esquemas que não fazem uso de uma malha computacional (LIU, 2002), para muitos métodos numéricos ela serve de alicerce para a discretização das equações no domínio de solução, através da qual são acessados dados como a posição geométrica dos nós ou volumes ou sua conectividade entre blocos vizinhos. Também é através dela que decomposições em blocos menores podem ser feitas, permitindo paralelizar o processo de solução. As malhas são classificadas geralmente por seu tipo de conectividade, podendo ser *estruturadas*, *não-estruturadas* e até mesmo *híbridas*. A escolha de um ou outro tipo de malha depende principalmente de questões como o método utilizado na discretização

das equações e a complexidade do problema e/ou a técnica de refinamento. Por exemplo, grande parte dos trabalhos com o método de elementos finitos utiliza malha não-estruturada, enquanto que malhas estruturadas são largamente empregadas em conjunto com o método de diferenças finitas e volumes finitos.

Em uma malha não-estruturada, os elementos utilizados geralmente permitem um melhor ajuste ao se trabalhar com geometrias complexas. Por exemplo, uma malha composta por triângulos (2D) ou tetraedros (3D) pode representar uma superfície esférica imersa no escoamento com um ajuste melhor do que o obtido com uma malha constituída apenas por hexaedros. Os refinamentos isotrópico (MELTON, 1996; LOURENÇO, 2007; NAKAHASHI, 1989; PERAIRE; PEIRO; MORGAN, 1992) e anisotrópico (CONNELL; HOLMES, 1993; KALLINDERIS; BARON, 1987) também podem ser realizados de maneira mais natural do que no caso estruturado. Entretanto, existe a necessidade do cálculo das conectividades a cada remalhamento e do armazenamento desses dados, cujo espaço utilizado e acesso são caros computacionalmente, quando comparados com malhas estruturadas. Em problemas de grandes proporções, isso muitas vezes impossibilita alcançar uma determinada ordem de convergência espacial (dependendo da exatidão do método de discretização). Esta é a principal justificativa da utilização de uma malha estruturada no presente trabalho. Exemplos de malhas dos tipos estruturada e não-estruturada são respectivamente apresentados na Fig. 2.1.

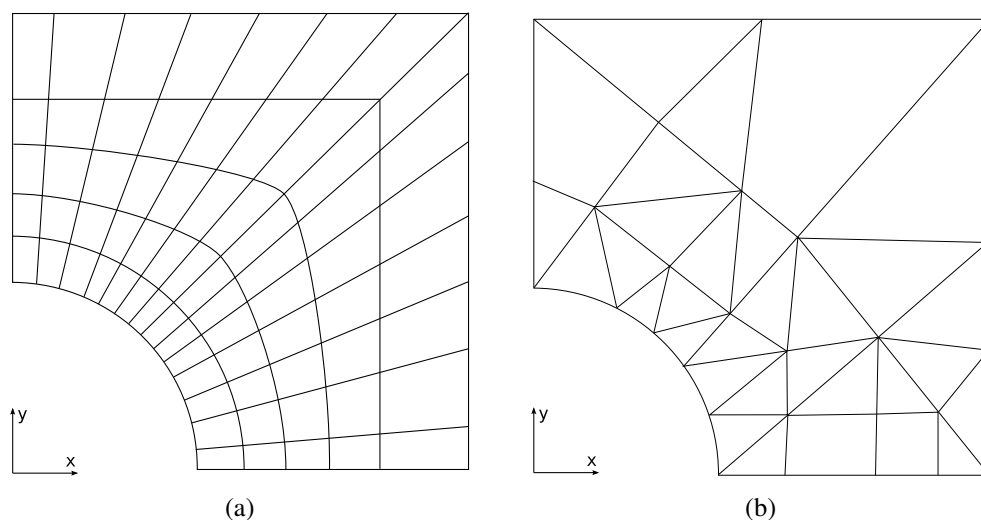


Figura 2.1: Exemplos de discretização. (a) utilizando malha estruturada e parametrizada e (b) não-estruturada com elementos retangulares e triangulares.

Malhas estruturadas do tipo cartesianas podem ser uniformes, quando constituídas por quadrados (2D) ou cubos (3D) e retilíneas, quando utilizam retângulos (2D) ou paralelepípedos (3D). Sendo que, nesse último caso, os elementos não precisam ser necessariamente congru-

entes. Sua conectividade é indicial (ou geométrica), o que permite acessar células vizinhas de forma muito mais simples do que quando se trabalha com uma malha não-estruturada. Por outro lado, embora a discretização cartesiana uniforme tradicional seja de implementação mais fácil, sua resolução constante deixa muito a desejar em situações envolvendo domínios grandes e/ou na representação de geometrias complexas.

Para contornar esse problema, muitas alternativas foram propostas. Entre as aproximações frequentemente adotadas estão: recortar as células, ou volumes eulerianos, que são interseccionados pelas fronteiras imersas (geralmente uma malha lagrangiana), como nas pesquisas de Karman Jr. (1995), Melton (1996) e Charlton e Powell (1997) e uma segunda abordagem, não menos utilizada, que consiste em representar a geometria imersa na forma de uma interface ou campo de força, como nos trabalhos de Pember et al. (1991), Roma, Peskin e Berger (1999), Popinet (2003) e Villar (2007). Entre os problemas que surgem ao se realizar aproximações por meio de células recortadas ou de células contendo uma determinada interface, estão a baixa exatidão, por estarem desacopladas do resto do escoamento, além do problema de determinação do passo de tempo global da simulação.

Mesmo com essas desvantagens, a relação custo/benefício da utilização de malhas cartesianas ou semi-cartesianas para simulações de grande porte é muito boa e justifica sua utilização no presente trabalho. A simulação de um escoamento envolvendo corpos de média a grande razões de aspecto sem a utilização de refinamento adaptativo e/ou paralelismo seria quase que impraticável atualmente, mesmo com a alta tecnologia em processamento computacional. A Fig. 2.2 ilustra uma malha cartesiana refinada em uma região de interesse por três técnicas diferentes.

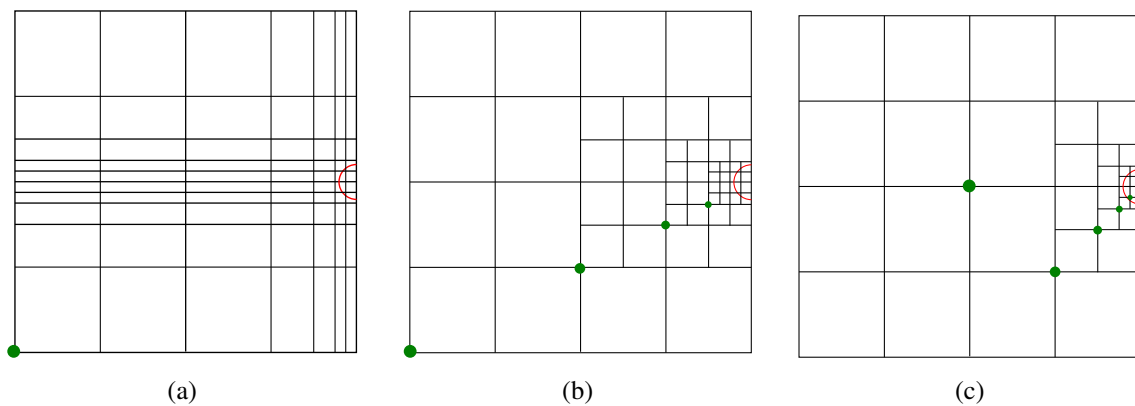


Figura 2.2: Malha cartesiana refinada apenas em uma pequena porção do domínio. (a) Malha cartesiana clássica com adição de linhas próxima à região de interesse; (b) baseada em blocos sobrepostos; e (c) com estrutura quadtree.

Geralmente, essa região de interesse pode ser a presença de uma determinada interface (ROMA; PESKIN; BERGER, 1999; AFTOSMIS; BERGER; ADOMAVICIUS, 2000) ou uma estimativa do erro calculada em todo o domínio, como discutido no trabalho de Le Moigne (2004), de forma a ajustar continuamente a resolução de acordo com as propriedades da solução (FRYXELL et al., 2000). Esse método é chamado de *refinamento adaptativo da malha* (AMR). Os pontos indicam os pontos de acesso à estrutura da malha. Na Fig. 2.2a, as linhas que formam a malha são deslocadas em direção à posição onde se deseja maior resolução. As variáveis podem ser armazenadas de forma contígua na memória e as coordenadas não precisam ser armazenadas, podendo ser determinadas a partir do ponto verde e da equação utilizada para o refinamento (ou *stretching*). É de implementação simples mas não barata computacionalmente, pois adiciona volumes desnecessários, além do problema de não convergência para células com alta razão de aspecto em métodos iterativos, como mencionado nos testes de Botta e Wubs (1993).

Malhas bloco-estruturadas são baseadas na inserção de blocos de malha mais fina nas regiões onde se deseja maior resolução. Na Fig. 2.2b é mostrada uma malha bloco-estruturada com razão 2:1, significando que o nível refinado é duas vezes menor que seu nível de base. A razão 2:1 é a mais utilizada, embora outras razões também sejam encontradas na literatura. É necessário o armazenamento dos índices de cada bloco para se ter acesso aos elementos em todos os níveis da malha. Geralmente o índice indica o nível do bloco e a posição geométrica do início da malha, como mostrado nos pontos verdes da Fig. 2.2b. Os índices são muito baratos para armazenamento e a iteração nas células pode ser feita como em uma malha cartesiana simples, otimizando o acesso à memória.

Na Fig. 2.2c, é exibido o corte bidimensional de uma malha octonária ou *octree*. O acesso às células de maior nível (mais refinadas) se dá por meio de uma hierarquia na razão 2:1, como mostrado nos pontos verdes na figura. Como será discutido na próxima seção, essa estrutura foi escolhida na execução desse trabalho pela simplicidade de paralelização e pelos resultados apresentados na literatura sobre o balanceamento de carga e implementação da técnica de multigrid.

2.2 Balanço de carga

Com a crescente utilização de computadores de múltiplos processadores e de *clusters* de computadores, o balanceamento de carga dinâmico se tornou algo de elevada importância na simulação numérica. Existem aplicativos, livres e comerciais, apenas com o objetivo de re-

alizer a distribuição dos nós/volumes de um determinado problema e realizar o balanceamento de carga. Esse é o caso de Zoltan (CATALYUREK et al., 2007), uma biblioteca para particionamento de domínios, utilizado no presente trabalho para realização do balanço de carga das malhas euleriana uniforme e lagrangiana na representação da fronteira imersa.

A biblioteca Zoltan é distribuída de maneira livre, possui código fonte aberto, além de possuir um conjunto de serviços de gerenciamento de dados para aplicações paralelas, não-estruturadas e dinamicamente adaptativas. Isso permite otimizar a comunicação entre diferentes processos, para evitar o desbalanceamento de carga na simulação, utilizando uma interface simples. No presente trabalho, sua utilização possui a finalidade de separar o domínio discretizado em partições, de acordo com o número de processadores disponíveis para a simulação. Essas partições devem, não somente possuir um número próximo de uma constante de células para cada um dos processadores, como também o posicionamento das interfaces desses domínios devem ser otimizadas para minimizar a comunicação de dados durante a simulação.

2.3 Estrutura de dados paralela *Octree*

O balanceamento de carga pode ser considerado satisfatório se os processadores estiverem todos trabalhando em sua capacidade máxima para executar uma tarefa comum. Se uma porcentagem significativa dos núcleos ficar ociosa, esperando dados de outro processo para continuar os cálculos, então existe um desbalanceamento de carga. Para conseguir que os processadores trabalhem de maneira uniforme, o problema a ser resolvido deve ser repartido igualmente e compartilhando o mínimo de informações com outros processadores.

Uma das formas de se realizar esse particionamento é através da aplicação de uma curva de preenchimento, como as de Sierpiński, de Lebesgue e de Hilbert, discutidas em mais detalhes em (HENDRICKSON; DEVINE, 2000). Matematicamente, uma curva de preenchimento espacial *Space Filling Curve* (SFC) é aquela em cuja extensão preenche um hipercubo de dimensão N . Isto significa que, utilizando uma curva de dimensão 2, é possível mapear um quadrado unitário. O primeiro a apresentar uma SFC bidimensional foi Giuseppe Peano (1858–1932) e, atualmente, devido às suas propriedades de discretização espacial, são utilizadas em muitas aplicações práticas como o *problema do caixeiro viajante* e em algoritmos de decomposição de domínio. Entre as mais conhecidas estão as curvas de Lebesgue (TU et al., 2006) e de Hilbert (HILBERT, 1891) e o mapeamento determinado por ambas possui propriedades semelhantes. Dentre as três curvas, as duas últimas foram testadas no trabalho e são ilustradas na Fig. 2.3.

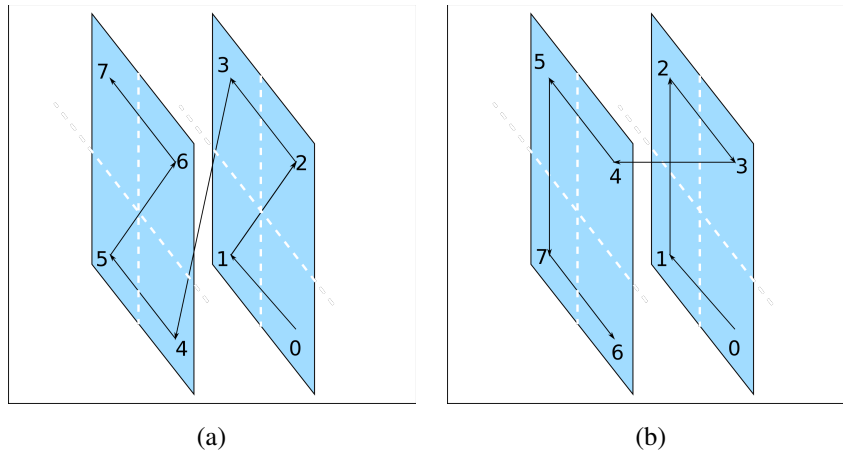


Figura 2.3: Exemplos de curvas de preenchimento de (a) Morton (Lebesgue) e de (b) Hilbert.

Os padrões mostrados na Fig. 2.3 foram aplicados em um cubo de tamanho 2^D , onde D é a dimensão do problema, sendo para esse caso com $D = 3$, tem-se 8 volumes nas figuras mostrada (cada centro é representado por um algarismo arábico). Em problemas práticos, esse cubo é subdividido quantas vezes forem necessárias para discretizar um determinado domínio, produzindo assim um grande número de sub-volumes. Geralmente, em volumes finitos ou diferenças finitas, os volumes na malha são indexados com base nos eixos cartesianos. Entretanto, para trabalhar com o modelo de distribuição espacial da Fig. 2.3 será utilizada uma estrutura na forma de uma árvore, na qual cada célula mostrada na figura possua outras 2^D células em seu interior e assim por diante. No caso bidimensional tem-se uma malha quadtree, como mostrado na Fig. 2.4, e no caso tridimensional uma malha octree. Uma vez que ambas as abordagens são semelhantes, aqui o caso bidimensional será explorado com mais frequência.

Embora esta seja uma técnica de discretização muito utilizada para processamento de imagens e em computação gráfica (ZHANG; WEBBER, 1993), mais recentemente encontrou aplicação na discretização de domínios para solução das equações de Euler e de Navier-Stokes (COIRIER, 1994; CHARLTON; POWELL, 1997; POPINET, 2003). Métodos sem-malha, assim como outras aproximações que trabalham com partículas, como em (LIU, 2002), também fazem uso de técnicas de discretização semelhante, devido principalmente à facilidade de deslocamento e localização de um ponto dentro de um domínio paralelepípedo.

Para apresentar o processo de construção da estrutura octree, será utilizada a Fig. 2.4, que mostra um exemplo de malha bidimensional e sua árvore correspondente. A figura mostra uma célula mãe, demarcada pela letra A, a partir da qual são realizadas 4 subdivisões consecutivas, sempre no canto inferior direito. Isso dá origem às células demarcadas, em ordem crescente de níveis, pelas letras de B à M. As células que são subdivididas, {A,C,G}, são chamadas de

mães e o restante, de folhas $\{B, D, E, F, H, I, J, K, L, M\}$.

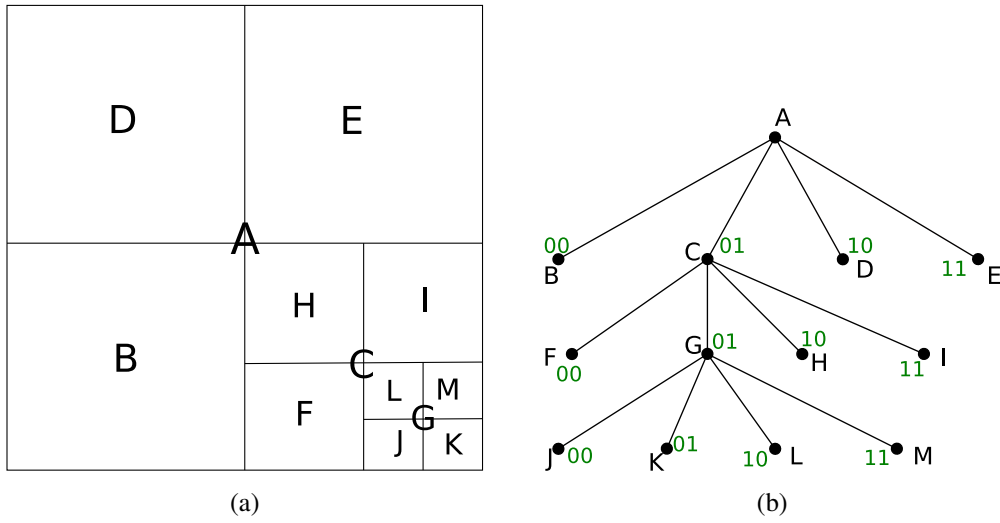
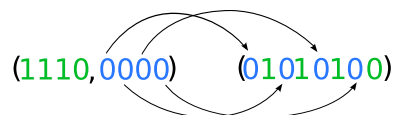


Figura 2.4: Exemplo de uma malha quadtree e sua representação na forma de uma árvore.

Hierarquicamente, essa estrutura pode ser representada pelo diagrama da Fig. 2.4b. Uma vez que o caso em questão envolve uma árvore quadtree, pode-se associar um identificador para cada folha para uma determinada célula mãe e isso pode ser feito de maneira recursiva. Nesse caso, apenas dois dígitos são necessários para identificação, e dentro de cada célula mãe as folhas serão enumeradas de 00 a 11, como mostrado na Fig. 2.4b. Dessa forma, para se acessar a célula K , a partir da célula A , ter-se-ia o caminho $C(01)$, $G(01)$ e $K(01)$ ou, de maneira concatenada, 010101. O principal inconveniente dessa aproximação é que o acesso à célula K depende das células G , C e A .

Um modo para se determinar a forma concatenada a partir da posição geométrica da célula, é empregando a codificação de Morton (TU et al., 2006). Para ilustrar esse método será utilizada novamente a célula K . A partir de um eixo coordenado em cada uma das direções, enumeram-se as folhas como em uma malha cartesiana. Nesse caso, supõe-se que a célula K tenha coordenadas $(14,0)$, do tipo inteiro, e está no nível mais refinado. Em um próximo passo, esses inteiros, representando as coordenadas, são escritos em notação binária e seus bits são intercalados um por um. Para a folha K , de coordenadas $(14,0)$ ou, escrita de modo binário, $(1110,0000)$, os bits poderiam ser intercalados como:



formando o mesmo identificador apresentado anteriormente, 10101000. Pode-se notar que exis-

tem dois zeros adicionais e também que, em representação decimal se torna o inteiro 42. Esse identificador pôde ser determinado sem o conhecimento prévio das células *A*, *C* e *G*. No trabalho de Dawes et al. (2009), essa operação é discutida com mais detalhes. Empregando-se esse procedimento, toda a estrutura pode ser armazenada na forma de uma sequência de inteiros.

2.3.1 Balanceamento de carga

Outra propriedade interessante da codificação de Morton é a possibilidade de realizar o balanceamento de carga para a estrutura criada. Ao se organizar, em ordem crescente, a sequência de inteiros resultante do procedimento apresentado anteriormente, tem-se as células da Fig. 2.4 na seguinte ordem: $\{B, F, J, K, L, M, H, I, D, E\}$. Essa lista de células pode ser facilmente distribuída de maneira balanceada entre os processos. A Fig. 2.5 ilustra a distribuição da estrutura apresentada entre dois processadores, no qual as células azuis pertencem ao primeiro processo e as verdes ao segundo.

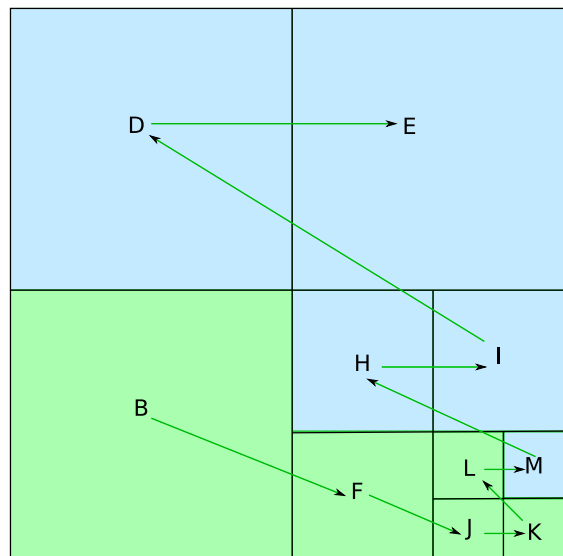


Figura 2.5: Distribuição de volumes entre dois processos utilizando a codificação de Morton.

Devido às conexões apresentadas na curva de Morton (Fig. 2.3a), muitas vezes são obtidos blocos completamente separados um do outro no mesmo processo. Isso ocorre com menos frequência quando se emprega a curva de Hilbert, pois, como pode ser observado no padrão da Fig. 2.3b, sempre a célula anterior da sequência compartilha ao menos um lado (ou face em 3D) com a próxima célula. Entretanto, o processo de criação da curva de Hilbert é mais complexo que a codificação de Morton. Nesse caso, é possível utilizar uma tabela, de maneira a mapear os índices da estrutura de Morton para aquela equivalente de Hilbert (CAMPBELL et

al., 2003). A Tabela 2.3.1 pode ser utilizada com esse propósito, para uma malha bidimensional.

No caso de uma malha quadtree, a linha de orientação i define as novas ordenações das células que se encontram dentro do bloco i . Assim, para uma malha contendo apenas um bloco com quatro células de ordem inicial Morton $\{0, 1, 2, 3\}$, fica na curva de Hilbert, $\{0, 1, 3, 2\}$. E se a célula 0 no bloco fosse refinada, a ordem de suas quatro células na curva de Hilbert seria $\{0, 2, 3, 1\}$. O caso tridimensional funciona de maneira análoga, embora que, nesse caso, existam 4 combinações diferentes para cada face do cubo, ou seja, a tabela de conversão terá 24 linhas e 8 colunas. Na implementação desse trabalho, o custo computacional para mapear de uma curva para outra é pequeno, uma vez que apenas os ponteiros (ou endereços) são trocados.

Tabela 2.3.1: Tabela de ordenação e de orientação para converter da curva de Morton para a curva de Hilbert em uma malha bidimensional.

Ordenação				Orientação			
0	1	3	2	1	0	0	2
0	2	3	1	0	1	1	3
3	1	0	2	3	2	2	0
3	2	0	1	2	3	3	1

Observando-se a Figura 2.6, pode-se comparar a qualidade de particionamento tridimensional para as duas curvas em determinado domínio quando dividido em 3 processos. Mesmo para esse domínio simples tem-se dois blocos de malha completamente independentes dentro do processo 2, com o uso da curva de Morton. A curva mapeada para Hilbert gerou partições bem mais definidas, justificando sua utilização.

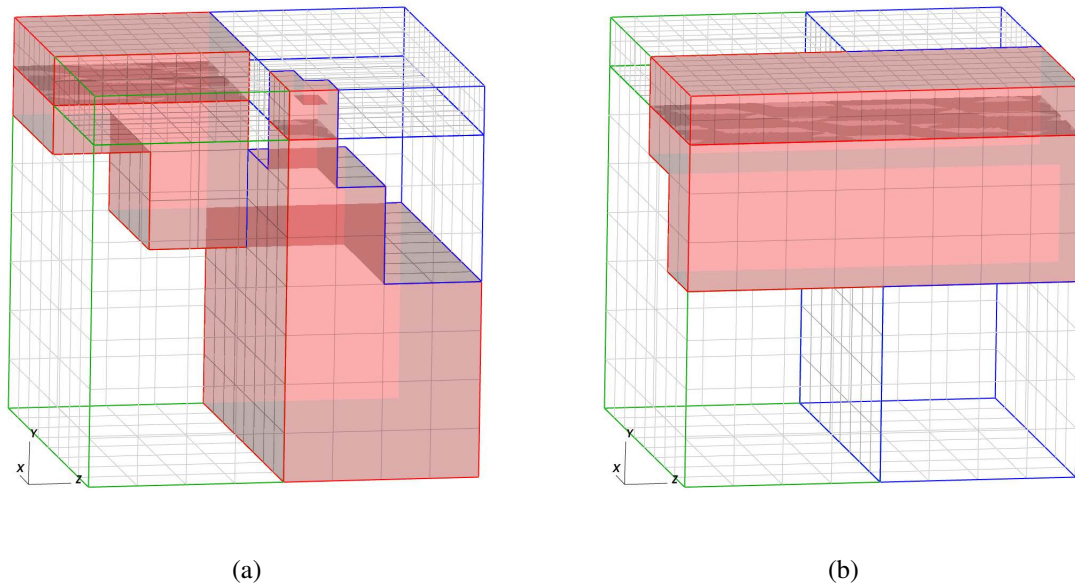


Figura 2.6: Domínio distribuído entre 3 processos utilizando curvas de preenchimento de (a) Morton e (b) Hilbert.

2.3.2 Estrutura interna da malha

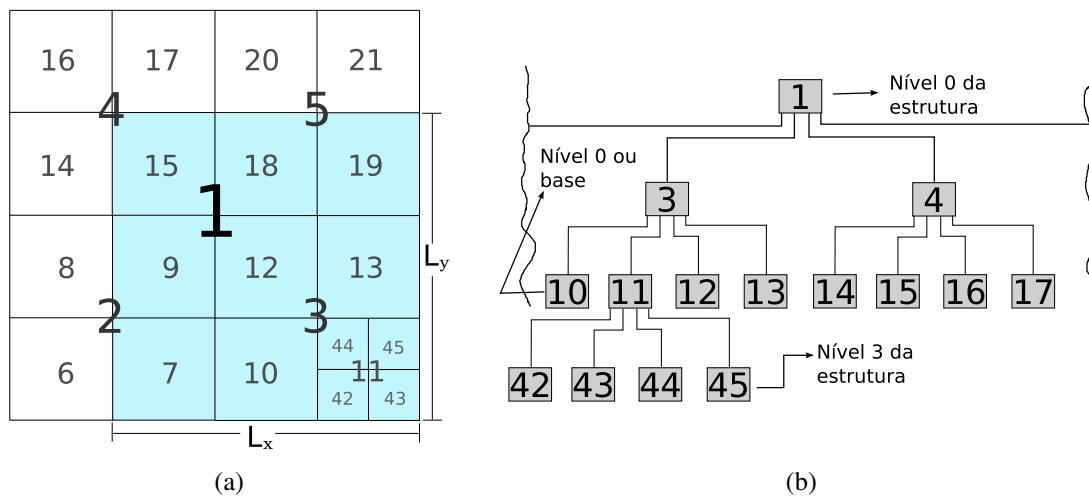


Figura 2.7: Malha 3x3 refinada dentro da estrutura quadtree.

A Figura 2.7 ilustra o exemplo de discretização de um domínio de dimensões (L_x, L_y) em uma malha com nível base 3x3 e com uma célula refinada utilizando uma estrutura quadtree. Os identificadores ou *ids* da malha são $\{7, 9, 10, 12, 13, 15, 18, 19, 42, 43, 44, 45\}$, enquanto que os *ids* restantes, $\{6, 8, 14, 16, 17, 20, 21\}$, existem apenas virtualmente na estrutura octonária e não representam células reais. Numerando-se a malha dessa maneira, é possível, utilizando-se

as relações 2.1, iterar através da malha, acessar células vizinhas, células mães e células que compõem o refinamento de determinada célula.

$$\begin{aligned}
 \text{mãe}(k) &= (k + 2^D - 2)/2^D \\
 \text{primeira filha da esquerda}(k) &= 2^D k - 2^D + 2 \\
 \text{última filha da direita}(k) &= 2^D k + 1 \\
 \text{número de blocos até o nível } N_p(l) &= [(2^D)^l - 1]/(2^D - 1)
 \end{aligned} \tag{2.1}$$

Nas relações acima, D é a dimensão da malha. No presente trabalho o armazenamento dos dados de cada célula é feito por uma *hashtable*, em função de seu identificador. Esta funciona como um dicionário: cada objeto pode ser recuperado a partir de uma chave que o identifica. Sobre a utilização de *octrees* com *hashtables*, veja-se (PRESS et al., 2007). Outras implementações também são possíveis, como o emprego de uma árvore de ponteiros, na qual cada mãe possui ponteiros para suas folhas. Entretanto, testes preliminares mostraram que a alternativa corrente apresenta um desempenho melhor.

A partir da utilização da SFC para distribuir a malha entre os diferentes processos, é criado um mapa que identifica cada célula local na malha global (em todos os processadores). Todas as listas de variáveis são construídas a partir desse mapa. De maneira semelhante a matriz de solução também utiliza um mapa semelhante. Uma ilustração de como as variáveis estão distribuídas na malha é dada na Fig. 2.8a. Para cada parênteses na forma (c, b, a) : a é a numeração local dos elementos. O acesso a cada variável das listas utiliza essa numeração; b é a numeração global, utilizada para acessar valores que estão em outro processo e para a criação da matriz de solução da correção de pressão; c define a identificação da célula dentro da árvore, como na Fig. 2.4a.

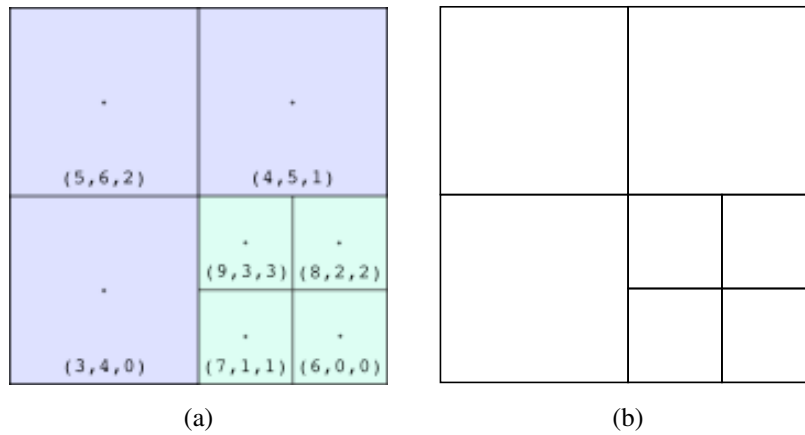


Figura 2.8: Distribuição das variáveis na malha. (a) mapeamento e (b) posicionamento das células fantasmas: (*azul*) condições de contorno; (*vermelho*) fantasmas da malha fina; (*verde*) fantasmas da malha grossa.

São utilizadas células fantasmas para imposição das condições de contorno e para atualização dos valores dos volumes compartilhados entre processos vizinhos. A Fig. 2.8b ilustra o posicionamento dessas células em uma malha bidimensional. As condições de contorno são definidas apenas nos lados dos quadrados, uma vez que não existem células além daquelas internas à estrutura octonária. Embora não esteja ilustrado na figura, nas células de processadores diferentes que compartilham a mesma face, os valores são injetados empregando-se uma técnica semelhante, com volumes fantasmas de mesmo nível.

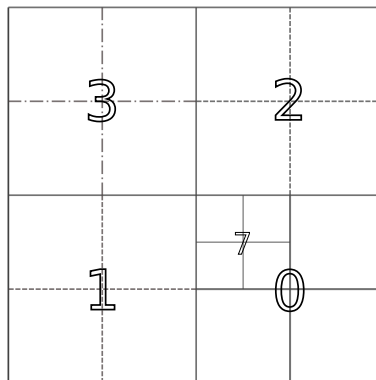


Figura 2.9: Restrição para o refinamento de malha.

No presente trabalho, foi adotado que a diferença entre níveis, ou seja, o número máximo que uma célula pode se dividir em relação à vizinhança, seja sempre na proporção 1 : 2. Isso facilita principalmente a construção das interpolações, utilizadas para transferência

de dados entre os diferentes níveis da malha, e que devem manter a conservação de certas propriedades físicas. Sendo assim, a estrutura da malha sempre terá uma construção como a apresentada na Fig. 2.8b. Tomando-se por exemplo uma malha com nível base de 2×2 , e refinando-se no ponto $p = (0,51L_x, 0,49L_y)$, que se situa na célula 0, com dois níveis, ter-se-á como resultado a malha mostrada na Fig. 2.9 (assumindo-se que L_x e L_y sejam as dimensões do domínio nas direções x e y respectivamente). As células 1 e 2 foram refinadas devido à restrição de cada célula vizinha compartilhar sua face sempre na razão de 1 : 2 e a célula 3 foi refinada para facilitar o cálculo de células fantasmas. No presente trabalho, as células de mesmo nível, em um mesmo processador, são aninhadas em *patches*, de modo que as operações realizadas sobre essas células, ou blocos de células, são baseadas em índices da mesma forma quando se trabalha com malhas retilíneas cartesianas simples.

2.4 Método da Fronteira Imersa

A definição do domínio de análise constitui um passo fundamental para a determinação de resultados para muitos problemas de engenharia. Aqui, considerar-se-á, como exemplo, o campo de mecânica dos fluidos, embora os conceitos discutidos possam ser aplicados em outras áreas. Neste caso, tanto as condições de contorno, como o tipo de aproximação para a modelagem, são adotadas, muitas vezes a partir do domínio ou da discretização deste. Quando uma solução numérica é buscada, essa discretização é feita através de uma malha, como as apresentadas nas seções anteriores e cujo tipo depende da natureza do problema a ser tratado e do esquema utilizado.

A complexidade da representação do problema físico muitas vezes se reflete na necessidade de malhas não somente com um grande número de células, mas também geometricamente complicadas. Esse é o caso, por exemplo, quando se trabalha com simulações envolvendo interações do tipo fluido-estrutura. Neste caso, embora distintas, as modelagens, tanto para o fluido, quanto para a estrutura precisam estar acopladas com referência ao domínio de cálculo.

Malhas não-estruturadas geralmente se adaptam muito bem à geometrias com curvaturas ou outros tipos de complexidade. Entretanto, o processo de geração e armazenamento deste tipo de malha, mesmo com o avanço do poder de processamento computacional nos últimos anos, continua caro. Na verdade, não somente para o caso de malhas não-estruturadas, uma relação satisfatória de *exatidão/custo computacional*, quando a simulação envolve geometrias móveis ou corpos deformáveis, ainda continua sendo um desafio para a área de experimentação numérica.

Uma alternativa viável é a utilização de malhas estruturadas, especialmente cartesianas. Entretanto, estas não possuem a capacidade de ajuste a geometrias complicadas. O Método de Fronteira Imersa, do inglês *Immersed Boundary Method* (IB) oferece a possibilidade de representar essas geometrias com pontos lagrangianos de uma malha não-estruturada e continuar a resolver problema na malha estruturada. Este método foi utilizado por Peskin (1972), na simulação do escoamento de sangue em válvulas cardíacas. Basicamente, o método consiste na imposição de um campo de força numérico que atua sobre o escoamento, representando assim o corpo imerso, com uma malha cartesiana representando o domínio euleriano. A representação da geometria imersa, assim como o posicionamento dos pontos lagrangianos, são dadas a partir de uma malha lagrangiana.

Utilizando esta aproximação, é possível, por exemplo, resolver o campo de velocidades e de pressão para o fluido, com um alto número de graus de liberdade, em uma malha carte-

siana acoplada à solução de uma outra estrutura com um número menor de graus de liberdade representada por uma malha lagrangiana. Quando se considera o refinamento e a adaptação da malha ao corpo a cada passo de tempo, o método IB se torna ainda mais atrativo. Isso porque em uma malha não-estruturada existe a necessidade de armazenamento de conectividades, pontos e outras propriedades que, em variáveis do tipo *ponto flutuante* iriam requerer uma quantidade muitas vezes significativas de memória, tanto para armazenamento quanto para o acesso.

A Figura 2.10 ilustra um exemplo de malha lagrangiana imersa em uma malha cartesiana regular. Trata-se apenas de uma malha fictícia, uma vez que a malha euleriana deveria ser muito mais refinada, principalmente próximo à fronteira imersa. Além da complexidade da malha lagrangiana, também pode ser notada a independência geométrica entre as malhas. Nota-se a complexidade geométrica da malha não-estruturada triangular lagrangiana em contraste com a simplicidade da malha cartesiana que descreve o domínio do problema.

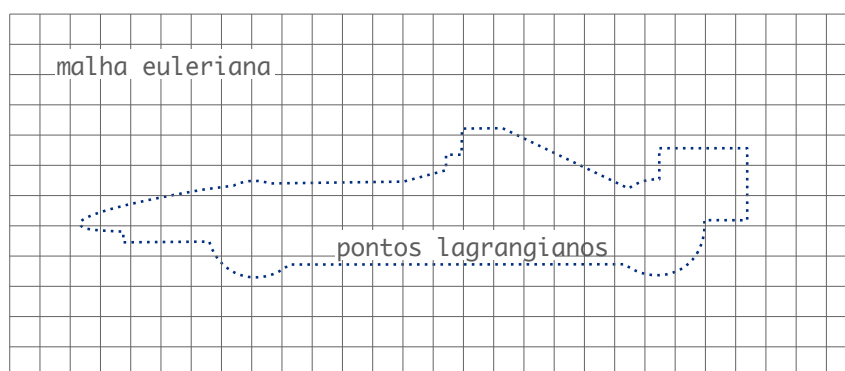


Figura 2.10: Esquema ilustrando as malhas euleriana e os pontos lagrangianos no método de fronteira imersa.

3 Revisão Bibliográfica

A modelagem de problemas em mecânica dos fluidos produz, muitas vezes, um conjunto de equações diferenciais parciais. Como apresentado anteriormente, malhas numéricas são amplamente utilizadas em diversos métodos como, por exemplo, FVM, FDM e FEM, para discretização espacial na solução dessas equações. Atualmente, existe um grande número de aproximações na criação dessas malhas, sejam estas estruturadas, não-estruturadas ou híbridas. Independentemente da aproximação, questões como tempo gasto na construção da malha, tipo de armazenamento e acesso à células da malha, e ajuste a geometrias complexas são constantemente abordadas.

O emprego de técnicas de refinamento adaptativo de malhas não-estruturadas e AMR bloco-estruturadas possibilitou, juntamente com algoritmos de alto desempenho como o método de multigrid, que problemas de solução muito cara computacionalmente, antes considerados de grande porte, pudessem ser resolvidos com muito mais desempenho. Nesta seção, são apresentados, em ordem cronológica, parte desses desenvolvimentos que, de uma forma ou de outra, acabaram por influenciar ou contribuir para os desenvolvimentos do presente trabalho.

3.1 Adaptatividade com malhas cartesianas

Malhas cartesianas já eram utilizadas na solução de problemas em mecânica dos fluidos na década de 70. No trabalho de Purvis e Burkhalter (1979), foi resolvida a equação do escoamento potencial não-linear utilizando o método de volumes finitos. A aproximação de axissimetria foi utilizada para a simulação do escoamento em regime permanente com número de Mach crítico. Axissimetria também foi utilizada em um trabalho mais recente, por Lourenço (2007), no escoamento de um fluido invíscido ao redor de uma esfera a alto número de Mach. No mesmo trabalho, foi realizada a simulação tridimensional do problema, mostrando-se que, embora seja muito mais barata, a suposição de axissimetria muitas vezes não reproduz o resultado tridimensional satisfatoriamente.

Thompson (1982) apresentou muitas relações para transformar malhas cartesianas em malha curvilíneas, assim como os diferentes tipos de configurações possíveis, em sistemas de coordenadas geradas numericamente, para o ajuste, de acordo com determinado objeto ou geometria no escoamento. Outro tipo de ajuste foi realizado no trabalho de Berger e Oliger (1984), na solução de equações diferenciais parciais hiperbólicas utilizando o método de diferenças finitas. Utilizando refinamento adaptativo em uma e duas dimensões, foi possível ajustar a resolução da malha para regiões específicas do domínio computacional. Além de resolução de malha, observou-se que, com o refinamento local baseado em determinados critérios, conseguia-se uma redução no custo computacional para a solução do problema.

O mapeamento de coordenadas, como mostrado em Thompson (1982), permite que problemas em geometrias complexas possam ser resolvidos com o uso de uma malha mais simples, como a cartesiana. Grossman e Whitaker (1986), entre outros, apresentam a solução da equação potencial bidimensional. Utilizando o método de volumes finitos sobre malhas cartesianas foi possível, através de um mapeamento para uma geometria retangular, resolver o escoamento supersônico ao redor de geometrias cônicas. Em uma configuração semelhante, solução tridimensional das equações de Euler em escoamentos transônicos seria dada um ano depois por Gaffney, Hassan e Salas (1987). Também utilizando malha cartesiana, foram apresentados resultados para o escoamento ao redor de uma asa com perfil *ONERA M6* utilizando volumes finitos. Entretanto, foi reportado o grande número de volumes adicionais, necessários para representar a borda de ataque de forma adequada.

O refinamento de malhas cartesianas através da simples subdivisão das células, pela metade em cada direção coordenada, é apresentada em Zeeuw e Powell (1991). A malha é refinada para se ajustar à geometria do corpo imerso ou então, até que aspectos indesejáveis da simulação sejam eliminados pela suavização adquirida devido ao ganho de resolução de malha. A técnica foi utilizada para a solução das equações de Euler na forma bidimensional em regime permanente. Os testes foram executados em escoamentos internos e externos, sobre elementos como aerofólios, um canal e uma elipse dupla. A malha resultante, conhecida como *quadtree*, se adaptou muito bem para capturar choques e outras singularidades características resultantes das simulações. Se um refinamento isotrópico fosse empregado, ao invés do tipo anisotrópico, seria uma técnica muito atrativa para escoamentos viscosos e incompressíveis, a moderados números de Reynolds.

Como uma extensão do trabalho anterior, em Quirk (1992), é resolvido o escoamento transiente ao redor de geometrias não-cartesianas utilizando-se volumes finitos, com *upwind* para discretização espacial dos termos advectivos. A proposta de utilizar o refinamento da ma-

lha cartesiana para se adaptar ao corpo se mostrou muito apropriada para capturar as particularidades do escoamento. Também utilizando volumes finitos e *upwind* para os termos convectivos, o trabalho de Coirier (1994) apresenta soluções das equações de Euler e de Navier-Stokes, em duas dimensões e com malha cartesiana refinada localmente. O refinamento se dá pela divisão recursiva das células nos pontos de interesse. Células que interseccionam determinadas superfícies são recortadas, de forma a se ajustar à geometria, através de algoritmos específicos (*cut-cell*). Por fim, foi mostrado que, com o refinamento automatizado da malha, é possível conseguir bons resultados, envolvendo geometrias complexas em malhas híbridas.

O esquema proposto por Berger e Oliger (1984) foi estendido por Roma, Peskin e Berger (1999), que apresentaram a aplicação do refinamento adaptativo em malhas cartesianas, justamente nas adjacências de corpos imersos no escoamento. No entanto, diferente da aproximação *cut-cell* de Coirier (1994), foi empregado o Método de Fronteira Imersa para representação da superfície do corpo. Com essa aproximação, o nível de exatidão próximo às fronteiras imersas foi melhorado pela adição de uma sequência de blocos, progressivamente refinados, de maneira aninhada sobre a malha *base*. Os resultados determinados utilizando-se a auto-adaptatividade desse método concordaram bem com aqueles calculados com malha uniforme e resolução do nível de malha mais fino, cuja concordância foi igualmente reproduzida no presente trabalho.

3.2 Paralelismo e malhas cartesianas

Grande parte dos códigos numéricos utilizados em simulações de mecânica dos fluidos é computacionalmente onerosa. A busca por resultados cada vez mais fiéis à física do problema, assim como outros desafios, fazem com que sejam exigidas máquinas com capacidade de memória e de poder de processamento cada vez maiores. Se, por um lado, modelos matemáticos, algoritmos e esquemas numéricos não são suficientes para amenizar o problema, processadores e memórias também não se desenvolvem na mesma velocidade que as necessidades da indústria e da pesquisa científica. Nesse contexto, o uso de vários processadores em paralelo para resolver um problema passou, de atividade de pesquisa, para um dos elementos mais fundamentais da computação científica (GROPP; KEYES; TIDRIRI, 1995).

Uma metodologia semelhante à apresentada em Zeeuw e Powell (1991), de malhas cartesianas do tipo *quad/oct-tree*, foi implementada de forma paralela por Popinet (2003). As equações de Euler transientes, para um escoamento incompressível, foram resolvidas utilizando os métodos de projeção para acoplamento pressão-velocidade e de multigrid, na solução da

equação de correção da pressão. É reportado um excelente desempenho computacional para o remalhamento adaptativo, com uma eficiência para o balanceamento de carga próxima de 1. A metodologia é barata, permitindo que a malha seja atualizada a cada passo de tempo, mesmo com a presença de corpos de geometrias complexas. Entretanto, devido principalmente à essa flexibilidade, é relatada uma perda de eficiência relativa ao acesso a dados armazenados de modo não-contíguo na memória.

Quando bem aprimorada, a paralelização numérica possui um grande potencial para realização de simulações de grande porte, como aquelas envolvendo sistemas físicos complexos. Steensland et al. (2000) apresentam um estudo aprofundado sobre a implementação de balanço de carga dinâmico em malhas adaptativas. É destacada não somente a importância de se particionar o domínio corretamente, mas sim de uma estratégia de particionamento dinâmico, com base nas características e estados da simulação durante a solução, e faz comparações com um exemplo estático. Os resultados mostraram uma grande melhoria no tempo de execução computacional utilizando o particionador adaptativo.

O armazenamento de dados em uma estrutura octree permite implementações paralelas extremamente eficientes, como mostrado no trabalho de Warren e Salmon (1993). O acesso aos dados é feito por meio de chaves em uma *hashtable*, ao invés de utilização de ponteiros, comum em armazenamento de árvores de dados, o que contribuiu para o alto desempenho nos resultados. Também utilizando estruturas do tipo octree, Campbell et al. (2003) apresentam estudos sobre a qualidade da escalabilidade e do particionamento dinâmico de malha utilizado quando essa é particionada por meio de uma *Space Filling Curve* (SFC). Os resultados mostraram que a ordenação de Hilbert apresenta partições mais bem definidas do que utilizando Morton ou *Gray code*, o que se encontra de acordo com a Fig. 2.6. Os desenvolvimentos foram inseridos na biblioteca de código aberto de particionamento *Zoltan*, utilizada no presente trabalho para o particionamento da malha uniforme.

Berger et al. (2005) reportam uma melhora significativa no processamento paralelo com memória compartilhada com OpenMP e implementação semelhante à que seria feita para memória distribuída em MPI. A ótima comparação de escalabilidade é justificada pela utilização de paralelismo apenas em nível de laços computacionais. Entretanto, os resultados de aceleração de processamento com até 640 nós mostraram uma diferença significativa quando utilizada malha única e com multigrid, mesmo para memória compartilhada. Códigos cartesianos AMR, baseados em *patches* e/ou blocos, mesmo implementados de modo paralelo apresentam geralmente bom desempenho, devido principalmente ao fato que o acesso a variáveis nos *patches* é feito de maneira contígua.

Uma extensão de (POPINET, 2003) para simulação de escoamentos oceânicos foi feita por Popinet e Rickard (2007). As equações de Navier-Stokes bidimensionais foram discretizadas utilizando o método de volumes finitos em uma malha cartesiana quadtree. Ambos os trabalhos apresentam a maneira de trabalhar com estruturas de dados na forma de uma árvore como sendo a principal responsável pela flexibilidade de adaptação e refinamento da malha. É reportado em Popinet e Rickard (2007) uma redução do número de pontos proporcional a duas ordens de magnitude, utilizando malha adaptativa quadtree. Em comparação com malhas completamente não estruturadas que oferecem as mesmas possibilidades, a construção da malha quadtree se mostra mais acessível, tanto em termos de memória quanto de cálculos computacionais.

Ricker (2008) apresenta as modificações necessárias na solução de Poisson proposta por Greengard e Huang (2002). Em ambos foram utilizadas malhas octree adaptativas, implementadas no código em Fortran *FLASH* (FRYXELL et al., 2001). Entre as modificações, está a solução direta da equação de Poisson, localmente por blocos, com condições de contorno de Dirichlet para cada bloco e em cada um dos níveis. A alternativa proposta foi paralelizada utilizando o padrão de comunicação paralela *MPI*. Aqui o particionamento da lista de blocos também foi realizado empregando uma SFC (TU et al., 2006; HILBERT, 1891). Foi reportada uma melhora substancial no tempo de solução paralela do multigrid, quando comparada à solução multigrid, no mesmo código, apresentada por (FRYXELL et al., 2001). Diferente das outras aproximações apresentadas, a solução multigrid proposta envolve a aplicação de uma solução direta para cada um dos blocos, nível a nível, o que, segundo os autores, contribuiu para a melhora em relação ao código de Fryxell et al. (2001), que utiliza o multigrid clássico, baseado em iterações de relaxação entre os níveis.

Analisando o apanhado da literatura à respeito da utilização de malhas computacionais na simulação de problemas em mecânica dos fluidos, nota-se que a aplicação de refinamento adaptativo e de recursos de paralelismo, como a partição de domínios, se tornou uma tarefa comum e necessária para vencer os desafios nessa área da engenharia. A malha adaptativa utilizada na determinação dos resultados no presente trabalho também é baseada em uma estrutura de dados octree e particionada com uma SFC. Entretanto, a solução da equação de Poisson pode ser resolvida de duas maneiras diferentes. A primeira proposta utiliza o esquema proposto por Ricker (2008), mas ao invés de uma solução por blocos, é realizada uma solução por níveis. Uma outra maneira, que se mostra mais promissora, é a incorporação das expressões para interpolação de variáveis na interface fina-grossa em um sistema de equações lineares que poderia, a grosso modo, ser resolvido tanto por métodos iterativos quanto diretos, ou até mesmo por uma

biblioteca de solução de sistemas lineares do tipo *caixa-preta*.

3.3 Metodologia de Fronteira Imersa

O conceito recente do método de fronteira imersa foi introduzido por Peskin (1972), em seu trabalho com a simulação do escoamento de sangue em um coração. Essa metodologia também consta nos trabalhos de pesquisadores da antiga União Soviética, sob o nome de *Fictitious Domains* (SMAGULOV, 1979). Devido principalmente à capacidade de lidar com geometrias complexas e flexíveis, o método foi difundido de forma expressiva entre a comunidade científica. Contudo, o método inicial apresentava problemas em termos da condição de estabilidade imposta pelo CFL. Peskin (1977) estendeu seu trabalho, agora com o movimento e interação da fronteira imersa com o fluido.

Unverdi e Tryggvason (1992) apresentaram uma técnica para simulação de escoamentos multifásicos transientes a partir das equações de Navier-Stokes, na qual uma interface separando dois fluidos foi imposta usando o método da fronteira imersa. O escoamento foi discretizado espacialmente com um esquema de segunda ordem utilizando FDM, e a interface foi explicitamente representada por uma malha lagrangiana não estruturada capaz de se mover no interior do escoamento. Um esquema de primeira ordem (Euler) foi utilizado na discretização temporal. A interface deformava-se continuamente, devido aos efeitos da tensão superficial e cuja localização era determinada por uma função indicadora. Resultados foram apresentados para a simulação do movimento de bolhas discretas para os casos bi e tridimensional.

Com base no método de fronteira imersa proposto por Peskin (1977), Goldstein, Adachi e Sakata (1993) propuseram um modelo para determinação do campo de força de densidade para escoamentos com a presença de corpos imersos utilizando o FDM e malha lagrangiana com pontos coincidentes àqueles da malha euleriana. O campo de velocidade é utilizado iterativamente em um esquema de realimentação (*feedback forcing method*) para calcular o termo de força. Duas constantes a serem ajustadas de acordo com o campo de velocidade (daí o nome *realimentação*) são utilizadas nesse cálculo. Uma produz uma frequência de oscilação natural e a outra amortece as oscilações de resposta. Na discretização espacial, foi utilizada uma expansão de Fourier nas direções periódicas e expansão de Chebyshev nas direções normais às paredes.

Este método de realimentação (*feedback forcing method*) para imposição da fronteira imersa foi utilizado posteriormente por Saiki e Biringen (1996). Foi empregado FDM de alta ordem para eliminação de oscilações artificiais. Foram simulados escoamentos ao redor de cilindros estacionários, rotativos e oscilatórios para números de Reynolds menores que 400.

Com o objetivo de eliminar a necessidade de ajustar constantes, Mohd-Yusof (1997) propôs que a força lagrangiana fosse determinada à partir da equação da quantidade de movimento do fluido na interface. Os valores das propriedades das partículas de fluido adjacentes foram interpolados utilizando *B-Splines* e algoritmos para localizar a geometria no interior do domínio que constituem pontos fracos no método. Uma discussão mais detalhada do método é apresentada em Mohd-Yusof (1998).

Ye et al. (1999) apresentaram um modelo de interpolação para imposição da fronteira imersa associado a um método de volumes finitos baseado em um esquema de diferenças centradas de segunda ordem de precisão. O objetivo foi simular o escoamento com segunda ordem de precisão, ao redor de corpos complexos, utilizando malha cartesiana simples. Um esquema de passo fracionado foi utilizado na discretização temporal, e as equações da quantidade de movimento discretizadas foram resolvidas de modo iterativo em cada passo de tempo. Contudo, foi utilizado um pré-condicionador para acelerar o processo de convergência da solução, que foi determinada apenas para baixos valores do número de Reynolds.

Os métodos propostos por Goldstein, Adachi e Sakata (1993) e Mohd-Yusof (1998) foram comparados em Fadlun et al. (2000), sendo que o segundo leva grande vantagem sobre o primeiro, especialmente devido ao baixo custo computacional para lidar com domínios tridimensionais. Mesmo a versão alterada do primeiro modelo para economia de tempo ainda era mais cara que o modelo de Mohd-Yusof (1998). Alguns testes foram feitos, empregando um algoritmo *preditor-corretor* com um número grande iterações por passo, visando melhorar o campo de escoamento ao redor dos obstáculos, mas isto se mostrou desnecessário.

Kim, Kim e Choi (2001) adicionaram termos de fonte/sumidouro à equação da continuidade no método de Mohd-Yusof (1997) com o objetivo de melhorar a precisão do método e determinar resultados mais consistentes. Empregaram FVM e os termos de fonte/sumidouro foram adicionados somente na superfície ou no interior do objeto imerso no escoamento. O esquema de interpolação utilizado foi apresentado como sendo numericamente mais estável, no que diz respeito à posição relativa entre fronteira imersa e o domínio discretizado, garantindo uma maior independência entre o número de pontos utilizados e suas distâncias relativas aos volumes de fluido circundantes.

Analisando os trabalhos de Fadlun et al. (2000) e Kim, Kim e Choi (2001), Uhlmann (2005) evidenciou intensas oscilações da força, ao simular a interação do escoamento de fluido com partículas rígidas. Visando atenuar essas oscilações, os autores adotaram a função delta regularizada de Peskin (1972), mantendo o esquema direto de forçagem, evitando assim restrições adicionais para o passo de tempo.

Existem na literatura muitas tentativas para tratar as oscilações providas de fronteiras móveis, envolvendo desde o aumento do passo de tempo (PAN; SHEN, 2009), a introdução de um termo adicional de força na interface (LIAO et al., 2010) ou a modificação do esquema temporal na equação de quantidade de movimento (MITTAL et al., 2008). Utilizando o trabalho de Uhlmann (2005) como base, Vanella e Balaras (2009) propuseram implementar um procedimento de reconstrução da função por mínimos quadrados móveis (MLS) tanto para a interpolação quanto para a distribuição da força. Demonstraram também que o método mantém segunda ordem em seu solver por diferenças finitas.

Baseados no trabalho de Mohd-Yusof (1997), que empregou o balanço da quantidade de movimento para calcular o termo de força para uma célula vizinha da superfície do corpo imerso, Lima e Silva, Silveira Neto e Damasceno (2003) propuseram um balanço em uma partícula de fluido na superfície do corpo. Devido a essa particularidade e ao fato da condição de não-deslizamento ser modelada de forma direta, foi dado ao método o nome de Modelo Físico Virtual (*Virtual Physical Model* - MFV). Uma outra vantagem do método diz respeito à utilização de funções de interpolação simplificadas, reduzindo seu custo computacional.

Um grande número de trabalhos deram sequência ao trabalho de Lima e Silva, Silveira Neto e Damasceno (2003). A técnica foi utilizada para determinação do escoamento ao redor de cilindros variáveis por Oliveira et al. (2004a) e escoamentos ao redor de obstáculos a altos números de Reynolds por Oliveira et al. (2004b). O escoamento forçado no interior de um duto-cavidade, com movimento no fundo da cavidade, é estudado em Arruda (2004). Campregher (2005) empregou o MFV em simulações da interação fluido-estrutura. Seu estudo sobre o escoamento ao redor de uma esfera ancorada por molas demonstrou as potencialidades do método. A metodologia foi estendida para simulação do movimento e deformação de corpos imersos no escoamento por Oliveira (2006). Simulações de escoamentos ao redor de geometrias altamente complexas utilizando MFV são apresentadas por Oliveira (2007) e por Vedovoto (2007). Escoamentos transicionais em canais cilíndricos-anulares com excentricidade variável foram simulados por ??).

A partir da revisão sobre o método da fronteira imersa pode-se concluir que, embora se trate de uma metodologia ainda em desenvolvimento, possui uma base bem fundamentada e validada tanto para corpos rígidos como móveis. A metodologia apresenta características atrativas, comprovadas pelo crescente número e qualidade dos trabalhos. Muito ainda pode ser obtido da técnica que, comparada com métodos atuais similares, apresenta capacidade de trabalhar flexivelmente com interfaces imersas, possuindo grandes potencialidades no estudo da mecânica dos fluidos computacional. No presente trabalho, foi implementado o método MLS e

são apresentados os resultados determinados com a técnica para a simulação do escoamento de Taylor-Couette.

4 Metodologia

4.1 Modelagem Matemática

Neste trabalho, é considerado um sistema constituído por escoamentos viscosos e incompressíveis. Assumindo que o fluido possui propriedades físicas constantes, como a massa específica ρ e a viscosidade cinemática ν . Para esse sistema, as equações de Navier-Stokes (balanço de quantidade de movimento), Eq. (4.1), e de conservação de massa, Eq. (4.2), são escritas aqui como:

$$\frac{\partial u_i}{\partial t} + \frac{\partial(u_i u_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \quad (4.1)$$

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (4.2)$$

Basicamente, essas são as equações utilizadas para solução do escoamento de fluidos no domínio euleriano. Estas também são explícita ou implicitamente utilizadas nos métodos de fronteira imersa e nas interpolações conservativas, empregadas na malha cartesiana adaptativa. A implementação e a modelagem utilizadas neste trabalho foram fortemente influenciadas pelo trabalho de Padilla e Silveira Neto (2005), do qual provém também o código sequencial e em malha retilínea utilizado nos desenvolvimentos iniciais deste trabalho.

4.1.1 Método de Fronteira Imersa

Nesta metodologia, a imposição da interface no fluido é feita através da utilização um termo fonte na equação de quantidade de movimento. Este termo é calculado na malha lagrangiana e então distribuído nos pontos da malha euleriana vizinhos à interface, possibilitando que o fluido sinta a presença do corpo imerso no escoamento. Para apresentar o método, as equações de Navier-Stokes na forma indicial para escoamentos incompressíveis e isotérmicos é utilizada

na forma das equações Eq. (4.1) e Eq. (4.2), nas quais o termo de força de corpo f_i , cujo valor é nulo para pontos do domínio euleriano não coincidentes com a interface imersa, é utilizado para representar a fronteira imersa. Matematicamente, o campo de força pode ser representado como:

$$\vec{f}(\vec{x}, t) = \int_{\Gamma} \vec{F}(\vec{x}_k, t) \delta(\vec{x} - \vec{x}_k) d\vec{x}_k \quad (4.3)$$

na qual δ é a função *Delta de Dirac*, k denota uma variável lagrangiana e $\vec{F}(\vec{x}_k, t)$ é a força lagrangiana, calculada sobre os pontos da interface.

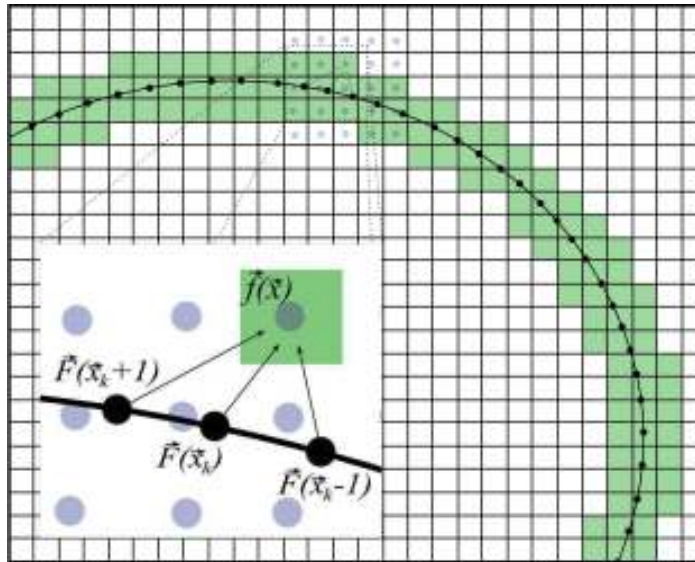


Figura 4.1: Distribuição da força lagrangiana para os volumes adjacentes (sombreados).

Uma forma de calcular a densidade de força lagrangiana é através da avaliação dinâmica da força que o fluido exerce sobre a superfície imersa no escoamento. Este é o procedimento, ilustrado bidimensionalmente na Figura 4.1, proposto pelo Modelo Físico Virtual (MFV), no qual a força lagrangiana é avaliada através de um balanço da quantidade de movimento sobre a partícula de fluido junto à interface, ou seja:

$$\vec{F}_i(\vec{x}_k, t) = \vec{F}_a(\vec{x}_k, t) + \vec{F}_i(\vec{x}_k, t) + \vec{F}_v(\vec{x}_k, t) + \vec{F}_p(\vec{x}_k, t) \quad (4.4)$$

na qual os termos de força de aceleração F_a , inercial F_i , viscosa F_v e de pressão são escritos respectivamente como:

$$\vec{F}_a = \rho \frac{\partial u_{ki}}{\partial t} \quad (4.5)$$

$$\vec{F}_i = \rho \frac{\partial u_{ki} u_{kj}}{\partial x_{kj}} \quad (4.6)$$

$$\vec{F}_v = -\frac{\partial}{\partial x_{kj}} \left[v \left(\frac{\partial u_{ki}}{\partial x_{kj}} + \frac{\partial u_{kj}}{\partial x_{ki}} \right) \right] \quad (4.7)$$

$$\vec{F}_p = \frac{\partial p_{kj}}{\partial x_{kj}} \quad (4.8)$$

Calculando a força necessária para impor a condição de contorno, o domínio lagrangiano pode ser acoplado ao euleriano, a partir da imposição da força da interface em pontos apropriados da malha euleriana. Um desenvolvimento mais completo da metodologia utilizando o MFV pode ser encontrada em Lima e Silva, Silveira Neto e Damasceno (2003), Oliveira et al. (2004a).

4.2 Modelagem Numérica

A solução das equações na forma contínua para o escoamento no domínio euleriano, Eqs. (4.1) e (4.2), é aproximada pela discretização das equações em uma malha cartesiana, que preenche o domínio físico representado por um paralelepípedo. Reescrevemos aqui a Eq. (4.1) na forma:

$$\int_V \frac{\partial u_i}{\partial t} dV = A_i + D_i - P_i + S_i \quad (4.9)$$

na qual as variáveis condensadas, A_i , D_i , P_i e S_i , representam as parcelas de advecção, difusão, pressão e de um termo fonte, respectivamente, e V é o volume mostrado na Fig. 4.2a.

4.2.1 Discretização espacial

Integrando cada termo da Eq. 4.9 no volume de controle elementar $\Delta V = \Delta x \cdot \Delta y \cdot \Delta z$, mostrado na Fig. 4.2, tem-se:

$$\int_V \frac{\partial u_i}{\partial t} dV = \frac{\partial u_i}{\partial t} \Delta x \cdot \Delta y \cdot \Delta z \quad (4.10)$$

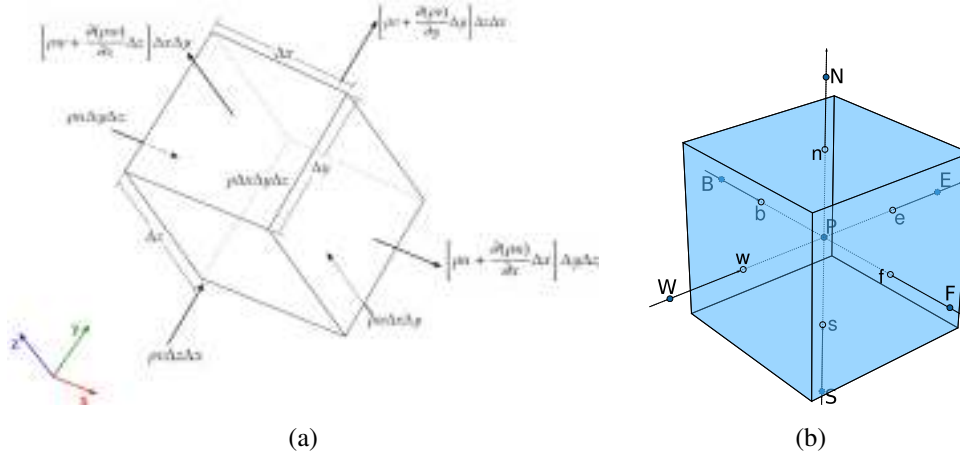


Figura 4.2: Volume de controle elementar ilustrando (a) os fluxos e (b) as posições definidas com letras minúsculas para os valores que se encontram nas faces e maiúsculas para aqueles no centro dos volumes.

para o termo de variação temporal,

$$\begin{aligned} A_i &= \int_V \left[\frac{\partial uu_i}{\partial x} + \frac{\partial vu_i}{\partial y} + \frac{\partial wu_i}{\partial z} \right] dV \\ &= [(uu_i)_e - (uu_i)_w] \cdot \Delta y \cdot \Delta z \\ &\quad + [(vu_i)_n - (vu_i)_s] \cdot \Delta z \cdot \Delta x \\ &\quad + [(wu_i)_f - (wu_i)_b] \cdot \Delta x \cdot \Delta y \end{aligned} \quad (4.11)$$

para os termos advectivos e,

$$\begin{aligned} D_i &= \int_V \left[\frac{\partial}{\partial x} \left(v \frac{\partial u_i}{\partial x} \right) + \frac{\partial}{\partial y} \left(v \frac{\partial u_i}{\partial y} \right) + \frac{\partial}{\partial z} \left(v \frac{\partial u_i}{\partial z} \right) \right] dV \\ &= v \left[\left(\frac{\partial u_i}{\partial x} \right)_e - \left(\frac{\partial u_i}{\partial x} \right)_w \right] \Delta y \Delta z + \\ &\quad v \left[\left(\frac{\partial u_i}{\partial y} \right)_n - \left(\frac{\partial u_i}{\partial y} \right)_s \right] \Delta z \Delta x + \\ &\quad v \left[\left(\frac{\partial u_i}{\partial z} \right)_f - \left(\frac{\partial u_i}{\partial z} \right)_b \right] \Delta x \Delta y \end{aligned} \quad (4.12)$$

para os termos difusivos. Na Eq. 4.12, as derivadas nas interfaces do volume de controle são aproximadas por diferenças centradas. A derivada da componente u_i para a face w pode ser escrita na forma:

$$\left(\frac{\partial u_i}{\partial x_i}\right)_w = \frac{u_{iP} - u_{iW}}{\Delta x_{iw}} \quad (4.13)$$

Assim como as outras parcelas, o termo de pressão, P_i também é integrado no volume, como:

$$P_i = \frac{1}{\rho} \int_V \frac{\partial p}{\partial x_i} dV \quad (4.14)$$

na qual ρ é a massa específica constante adotada. De modo análogo, $S_i = \int_V f_i dV$, na qual f_i representa todas as forças externas atuando no escoamento, na direção i , por unidade de volume.

4.2.2 Discretização temporal

A integração no tempo da Eq. (4.9) é realizada utilizando o esquema de Adams-Bashforth. O método, apresentado por Bashforth e Adams (1883) e foi modificado e desenvolvido por outros autores, como Lilly (1965), é uma maneira conveniente de extrapolar explicitamente os termos da Eq. (4.9) avaliados no tempo anterior, evitando a solução de um sistema de equações lineares adicional para o campo de velocidades. Assim, a integração da Eq. (4.9), por unidade de volume, fica:

$$\frac{u_i^{t+1} - u_i^t}{\Delta t} = \frac{3}{2} [-A_i + D_i + S_i]^t - \frac{1}{2} [-A_i + D_i + S_i]^{t-1} - P_i^{t+1} \quad (4.15)$$

As componentes de velocidade, u_i^{t+1} na Eq. (4.15) necessitam do termo de pressão P_i^{t+1} , avaliado no tempo $t + 1$. Esse acoplamento precisa ser feito de forma que o campo de velocidades satisfaça a equação da continuidade. Aqui, foi utilizado o método clássico de *passos fracionados* (KIM; MOIN, 1985) no qual, ao invés de utilizar um método iterativo, o acoplamento é feito de maneira segregada, na qual as soluções para a quantidade de movimento e pressão são calculadas somente uma vez a cada passo de tempo. Foi utilizada a versão do método com dois passos, no qual a Eq. (4.15) é dividida em duas partes denominadas de passos preditor e corretor. Esses passos são apresentados respectivamente nas Eqs. (4.16) e (4.17).

$$\frac{u_i^* - u_i^t}{\Delta t} = \frac{3}{2} [-A_i + D_i + S_i]^t - \frac{1}{2} [-A_i + D_i + S_i]^{t-1} - P_i^t \quad (4.16)$$

$$\frac{u_i^{t+1} - u_i^*}{\Delta t} = -P'_i \quad (4.17)$$

Na Eq. (4.17), P' representa o gradiente¹ do campo de correção da pressão p' . No primeiro passo estima-se o valor do campo de velocidades u_i^* utilizando a Eq. (4.16) com o valor da pressão no tempo t . No segundo passo, u_i^{t+1} e p são corrigidos pela aplicação de uma projeção aproximada de u_i^* , gerando P' (FERZIGER; PERIC, 2002). Derivando a Eq. (4.17), tem-se:

$$\frac{\partial P'}{\partial x_i} = \frac{1}{\Delta t} \frac{\partial u_i^{t+1}}{\partial x_i} - \frac{1}{\Delta t} \frac{\partial u_i^*}{\partial x_i} \quad (4.18)$$

na qual, substituindo a Eq. (4.2), o primeiro termo do lado direito pode ser retirado e a equação de correção da pressão pode ser escrita como:

$$\frac{\delta}{\delta x_i} \left(\frac{\delta p'}{\delta x_i} \right) = \frac{\rho}{\Delta t} \frac{\delta u_i^*}{\delta x_i} \quad (4.19)$$

cuja discretização resulta em um sistema de equações lineares esparsas, no qual a solução calculada é a correção de pressão, e o termo fonte do lado direito é proporcional à Eq. (4.2), avaliada no tempo t . Uma vez resolvida, as velocidades são corrigidas:

$$u_i^{t+1} = u_i^{*t} - P'_i \Delta t \quad (4.20)$$

logo, a partir das equações Eq. (4.16) e Eq. (4.17), a pressão também é atualizada para o tempo $t + 1$:

$$p^{t+1} = p^t + p' \quad (4.21)$$

¹ P'_i é a derivada da correção de pressão (ou a integral do gradiente) na direção i .

Critério para o cálculo do Δt

Geralmente, uma simulação numérica irá apresentar estabilidade e convergência dependendo de certos fatores, que no presente trabalho, são dados pelos tamanhos da malha e do passo de tempo. Aqui, foi adotado o critério *CFL* (*Courant-Friedrichs-Lewy*), que relaciona o passo tempo em função da velocidade e do comprimento característico de uma célula da malha, ou seja, calcula o intervalo temporal necessário para que uma onda com determinada velocidade cruze uma das células da malha. As condições de advecção e difusão para o cálculo do passo de tempo são:

$$\Delta t = \frac{1}{\frac{1}{\Delta t_{adv}} + \frac{1}{\Delta t_{dif}}} \quad (4.22)$$

na qual as contribuições advectiva e difusiva são dadas por:

$$\begin{aligned} \Delta t_{adv} &= \frac{1}{\frac{u_i}{\Delta x_i}} \\ \Delta t_{dif} &= \frac{1}{2 \cdot (1 + \nu) \cdot \frac{1}{(\Delta x_i)(\Delta x_i)}} \end{aligned}$$

Para certos problemas ou condições de análise, nem sempre o Δt calculado a partir da Eq. (4.22) garante a convergência e/ou a estabilidade no decorrer da simulação. No presente trabalho, o valor de Δt é então multiplicado por valores de *CFL* que variam de $CFL = 0,25$ a $CFL = 1,0$.

4.2.3 Metodologia de Fronteira Imersa

O Método de Multi-Forçagem Direta

Nesta seção será discutido o método de fronteira imersa utilizado no presente trabalho, uma aproximação denominada de *Multi Direct Forcing (MDF)*, apresentada por Wang, Fan e Luo (2008). Nesse método a imposição da força se dá através de um processo iterativo para se calcular a força lagrangiana, $\vec{F}(\vec{x}_k, t)$, semelhante ao método de aplicação direta da força, ou *Direct Forcing (DF)* proposto por Uhlmann (2005). Outros autores apresentam variações do método, como em Shu, Liu e Chew (2007), Su, Lai e Lin (2007).

Isolando o termo de força f_i na equação de quantidade de movimento Eq. 4.1 tem-se:

$$\frac{f_i}{\rho} = \frac{\partial u_i}{\partial t} + \frac{\partial(u_j u_i)}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j} \left[v \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \quad (4.23)$$

e,

$$\frac{F_{ki}}{\rho} = \frac{\partial u_{ki}}{\partial t} + \frac{\partial(u_{kj} u_{ki})}{\partial x_{kj}} + \frac{1}{\rho} \frac{\partial p}{\partial x_{ki}} - \frac{\partial}{\partial x_{kj}} \left[v \left(\frac{\partial u_{ki}}{\partial x_{kj}} + \frac{\partial u_{kj}}{\partial x_{ki}} \right) \right] \quad (4.24)$$

nas quais as variáveis lagrangianas são denotadas com a letra k e $F_{ki}(x_k, t)$ é a força lagrangiana, definida uma vez que o domínio lagrangiano está contido no euleriano e que a Eq. 4.23 foi deduzida utilizando a hipótese do contínuo. Discretizando explicitamente a derivada temporal da Eq. 4.24 utilizando Euler, tem-se:

$$\frac{F_{ki}}{\rho} = \frac{u_{ki}^{t+\Delta t} - u_{ki}^t}{\Delta t} + RHS_{ki}^t \quad (4.25)$$

na qual Δt é o intervalo discreto da análise e $u_{ki}^t = \frac{\partial(u_{kj} u_{ki})}{\partial x_{kj}} + \frac{1}{\rho} \frac{\partial p}{\partial x_{ki}} - \frac{\partial}{\partial x_{kj}} \left[v \left(\frac{\partial u_{ki}}{\partial x_{kj}} + \frac{\partial u_{kj}}{\partial x_{ki}} \right) \right]$. Utilizando um parâmetro temporário u_{ki}^* é possível escrever a Eq. 4.25 como:

$$\frac{F_{ki}}{\rho} = \frac{u_{ki}^{t+\Delta t} + u_{ki}^* - u_{ki}^* - u_{ki}^t}{\Delta t} + RHS_{ki}^t \quad (4.26)$$

A Eq. 4.26 pode ser decomposta em duas equações complementares:

$$\frac{u_{ki}^* - u_{ki}^t}{\Delta t} + RHS_{ki}^t = 0 \quad (4.27)$$

$$\frac{F_{ki}}{\rho} = \frac{u_{ki}^{t+\Delta t} - u_{ki}^*}{\Delta t} \quad (4.28)$$

A Eq. 4.27 pode ser resolvida no domínio euleriano para determinar u_i^* como:

$$u_i^* = u_i^t - RHS_i^t \Delta t \quad (4.29)$$

Para calcular a força lagrangiana é necessário avaliar u_{ki}^* na Eq. 4.28. Esse valor pode ser interpolado a partir do valor de u_i^* calculado no campo euleriano através de uma equação da forma:

$$u_{ki}^* = \sum_p u_i^* D_h(x_{ki} - x_i) h^3 \quad (4.30)$$

na qual os pontos p do somatório dependem da *função de distribuição* D_h adotada. Um exemplo de tal função, proposta por Juric e Tryggvason (1996) é dado na Eq. 4.31.

$$D_h(\vec{x}_k - \vec{x}) = \frac{1}{h^3} [W(r_x)W(r_y)W(r_z)] \quad (4.31)$$

com

$$W(r_x) = \begin{cases} \frac{3-2|r_x| + \sqrt{1+4|r_x|-4|r_x|^2}}{8} & \text{se } 0 \leq |r_x| \leq 1 \\ \frac{5-2|r_x| - \sqrt{-7+12|r_x|-4|r_x|^2}}{8} & \text{se } 1 < |r_x| \leq 2 \\ 0 & \text{se } |r_x| > 2 \end{cases} \quad (4.32)$$

na qual $r_x = \frac{x_{ki}-x_i}{h}$ e h é o espaçamento de discretização do domínio euleriano. A função de distribuição depende muito da física e da ordem de precisão requeridas na aplicação (MARIANO, 2007). Uma expressão mais simples, mas comumente utilizada é a função *Chapéu*:

$$W(r_x) = \begin{cases} 1 - |r_x| & \text{se } |r_x| < 1 \\ 0 & \text{se } |r_x| \geq 1 \end{cases} \quad (4.33)$$

Ambas as funções possuem integrais unitárias no intervalo $[-\inf, \inf]$, representadas respectivamente pelas áreas preenchidas em verde e azul na Fig. 4.3.

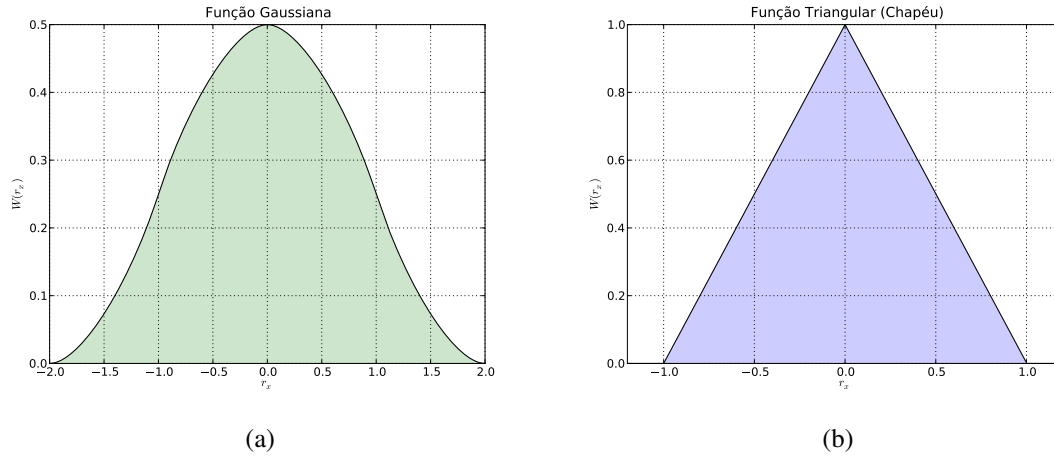


Figura 4.3: Funções de distribuição. (a) Função distribuição do tipo gaussiana proposta por Juric e Tryggvason (1996) e (b) função triangular, também conhecida como função chapéu.

Reconstrução por Mínimos Quadrados Móveis (MLS)

O método MDF, descrito previamente, foi utilizado por Borges (2009) na simulação de um problema com interação fluido-estrutura. Foi reportado um ganho de até 16 vezes no tempo para simular até 5 segundos de relógio físicos em comparação com o método MFV. Essa diferença se deu principalmente pelo fato dos métodos de forçagem direta não terem restrições tão severas quanto ao valor do passo de tempo.

No desenvolvimento do presente trabalho, entretanto, se notou que uma restrição no método MDF era justamente sua forma de forçagem, ou de múltiplas iterações de forçagem. Não foi possível estabelecer um critério confiável para a determinação do número máximo ou mínimo de iterações, de tal maneira que a fronteira pudesse ser bem representada. Dessa forma, testes foram feitos em outro método de forçagem direta, no qual a interpolação e distribuição de valores entre as malhas euleriana e lagrangiana são feitas utilizando-se de uma *Reconstrução por Mínimos Quadrados Móveis* (MLS), apresentada, também para solução de problemas de fluido-estrutura, no trabalho de Vanella e Balaras (2009).

Assim como no método MDF, a força representativa para o corpo é calculada nos pontos lagrangianos e então distribuída para os pontos eulerianos. Entretanto, aqui a interpolação é feita utilizando-se funções de forma de suporte compacto para MLS (LIU, 2002). Este processo de interpolação será descrito com base na Fig 4.4. Embora seja possível utilizar volumes de tamanhos diferentes na malha, supõe-se que na Fig. 4.4, as dimensões das células sejam, respectivamente, de h_x e h_y nas direções x e y .

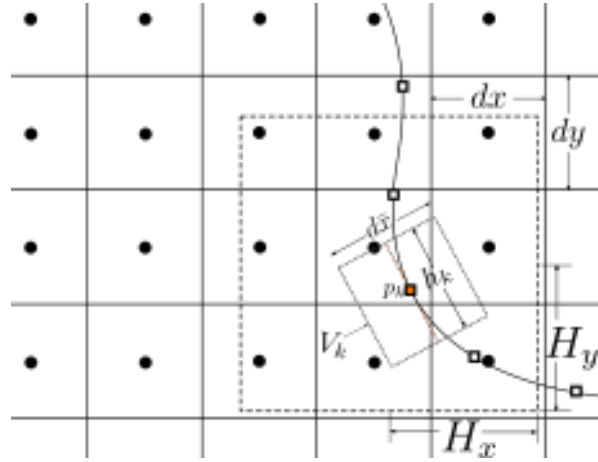


Figura 4.4: Diagrama esquemático para ilustrar o método MLS.

Nota: Recriada com base no trabalho de Vanella e Balaras (2009).

A cada ponto lagrangiano, associa-se uma área proporcional ao produto do tamanho característico dos lados dos volumes pelo tamanho dos segmentos da malha não estruturada, para o caso bidimensional ilustrado na Fig. 4.4. Em três dimensões, adotou-se o volume associado como sendo o produto da área do elemento lagrangiano pelo comprimento médio dos lados do volume euleriano. Essas aproximações para a área e para o volume do ponto lagrangiano estão de acordo com aquelas utilizadas em Uhlmann (2005). O domínio de influência para um ponto lagrangiano é representado pelo retângulo de dimensões $2H_x \times 2H_y$ centralizado nesse ponto. Os valores de H_x , H_y dependem do posicionamento do ponto lagrangiano e da resolução da malha euleriana próxima ao ponto.

A partir da equação discretizada para o acoplamento pressão-velocidades Eq. 4.16,

$$u_i^* = u_i^t + \Delta t \left[\frac{3}{2} (-A_i + D_i + S_i)^t - \frac{1}{2} (-A_i + D_i + S_i)^{t-1} - P_i^t \right] \quad (4.34)$$

é possível calcular a velocidade no ponto lagrangiano u_{ki} em função da componente aproximada na malha cartesiana \tilde{u}_i . Fazendo uso do método MLS, uma aproximação para u_{ki} em termos dos valores dentro de seu domínio de influência pode ser dada como:

$$\tilde{u}_{ki}(\mathbf{x}_k) = \sum_{j=1}^m p_j(\mathbf{x}_k) a_j(\mathbf{x}_k) = \mathbf{p}^T(\mathbf{x}_k) \mathbf{a}(\mathbf{x}_k) \quad (4.35)$$

na qual T significa que o vetor ou matriz foi transposto, \mathbf{x}_k são as coordenadas do ponto lagrangiano, $\mathbf{a}(\mathbf{x}_k)$ é o vetor de coeficientes e $\mathbf{p}(\mathbf{x}_k)$ o vetor com as funções de base, que aqui será adotado como sendo formado por uma base linear. Essas funções de base necessitam apenas de serem funções conhecidas de \mathbf{x} . Devido ao estêncil de malha utilizado neste trabalho e da ordem de exatidão da discretização utilizada, a base adotada será a mesma utilizada nos trabalhos de Vanella e Balaras (2009).

$$\mathbf{a}^T(\mathbf{x}_k) = [a_0(x_k) \ a_1(x_k) \ a_2(x_k) \ \dots \ a_m(x_k)] \quad (4.36)$$

$$\mathbf{p}^T(\mathbf{x}_k) = [1 \ x_k \ y_k \ z_k] \quad (4.37)$$

embora monômios de mais alta ordem possam ser utilizados, como por exemplo, para um caso bidimensional e de segunda ordem, teria-se:

$$\mathbf{p}^T(x_k, y_k) = [1 \ x_k \ y_k \ x_k^2 \ x_k y_k \ y_k^2] \quad (4.38)$$

Para determinar os valores dos coeficientes $a_j(\mathbf{x}_k)$ a partir das funções de base Eq. (4.37), admite-se que exista uma função peso $W(\mathbf{x}_k - \mathbf{x})$, de tal maneira que o funcional de resíduos ponderados J seja construído como:

$$J = \sum_{k=1}^{np} W(\mathbf{x}_k - \mathbf{x}) [\mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}_k) - \tilde{u}_i]^2 \quad (4.39)$$

onde \mathbf{x} é o vetor de posição do ponto euleriano, np o número de pontos no estêncil de interpolação e \tilde{u}_i a componente i de velocidade aproximada da Eq. (4.34). O resíduo ponderado é minimizado por $\mathbf{a}(\mathbf{x})$ para qualquer ponto arbitrário \mathbf{x} , ou seja, é necessário que $\frac{\partial J}{\partial \mathbf{a}} = 0$. Minimizando J para determinar $\mathbf{a}(\mathbf{x})$, chega-se ao seguinte conjunto de equações:

$$\begin{aligned} \mathbf{A}(\mathbf{x}_k) \mathbf{a}(\mathbf{x}_k) &= \mathbf{B}(\mathbf{x}_k) \tilde{\mathbf{u}}_i \\ \mathbf{A}(\mathbf{x}_k) &= \sum_{k=1}^{np} W(\mathbf{x}_k - \mathbf{x}) \mathbf{p}(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) \\ \mathbf{B}(\mathbf{x}_k) &= [W(\mathbf{x} - \mathbf{x}^1) \mathbf{p}(\mathbf{x}^1) \ \dots \ W(\mathbf{x} - \mathbf{x}^{np}) \mathbf{p}(\mathbf{x}^{np})] \end{aligned} \quad (4.40)$$

no qual $\mathbf{A}(\mathbf{x}_k)$ é a *matriz de momentos*. As dimensões de \mathbf{A} e \mathbf{B} dependem do vetor de funções

de base adotado que, para o vetor \mathbf{p} linear adotado neste trabalho, são respectivamente dadas como 4×4 e $4 \times np$. Admitindo que a matriz \mathbf{A} seja não singular, tem-se:

$$\mathbf{a}(\mathbf{x}_k) = \mathbf{A}^{-1}(\mathbf{x}_k) \mathbf{B}(\mathbf{x}_k) \tilde{\mathbf{u}}_i \quad (4.41)$$

que, substituindo na Eq. (4.35), dá:

$$\tilde{u}_{ki}(\mathbf{x}_k) = \sum_I^n \sum_{j=1}^m p_j(\mathbf{x}_k) \left[\mathbf{A}^{-1}(\mathbf{x}_k) \mathbf{B}(\mathbf{x}_k) \right]_{jI} \tilde{\mathbf{u}}_I \quad (4.42)$$

e uma vez que:

$$\tilde{u}_{ki}(\mathbf{x}_k) = \sum_{k=1}^{np} \phi(\mathbf{x}) \tilde{u}_i^k \quad (4.43)$$

chega-se às funções de forma $\phi(x)$ como:

$$\begin{aligned} \Phi_I(\mathbf{x}) &= \sum_{j=1}^m p_j(\mathbf{x}_k) \left[\mathbf{A}^{-1}(\mathbf{x}_k) \mathbf{B}(\mathbf{x}_k) \right]_{jI} \\ &= \mathbf{p}^T \mathbf{A}^{-1} \mathbf{B}_I \end{aligned} \quad (4.44)$$

A função peso atuará de maneira semelhante às funções de distribuição da Fig 4.3, devendo variar suavemente e decrescer monotonicamente no domínio de influência. Adotou-se *splines* cúbicas para os testes realizados neste trabalho como descrito na seguinte equação:

$$W(\mathbf{x}_k - \mathbf{x}) = \begin{cases} 2/3 - 4\bar{r}_k^2 + 4\bar{r}_k^3, & \text{se } \bar{r}_k \leq 0,5 \\ 4/3 - 4\bar{r}_k + 4\bar{r}_k^2 - 4/3\bar{r}_k^3, & \text{se } 0,5 \geq \bar{r}_k \geq 1 \\ 0, & \text{se } \bar{r}_k > 1 \end{cases}$$

na qual $\bar{r}_I = |\mathbf{x}_k - \mathbf{x}|/H_i$. Assim, a força volumétrica F_{ki} sentida por cada um dos pontos lagrangianos pode ser calculada substituindo a expressão na Eq. (4.43) na Eq. (4.28). Para o procedimento inverso, ou seja, impor a contribuição das forças lagrangianas nodais na malha

euleriana, as mesmas funções de forma são utilizadas, desde que ponderadas por um fator e_f , deduzido com base nos volumes médios entre as duas malhas, dados nas seguintes equações,

$$f_i = \sum_{k=1}^{nk} e_{fk} \phi F_{ki} \quad e_{fk} = \frac{V_k}{\bar{V}} \quad (4.45)$$

na qual f_i é a força volumétrica sentida no volume euleriano devido às contribuições dos nk pontos lagrangianos associados. A ponderação é realizada devido ao fato dos volumes lagrangianos e eulerianos não serem iguais. O fator de escala para transferir a força pode ser deduzido usando que a força sentida pelo fluido deve ser a mesma nos dois domínios:

$$\sum_{I=1}^n f_i \Delta V = \sum_{k=1}^{nk} F_{ki} \Delta V_k \quad (4.46)$$

na qual n é o número de volumes associados à algum ponto lagrangiano e os volumes das células euleriana e lagrangiana são dados respectivamente como:

$$V = dx \cdot dy \cdot dz \quad V_k = A_k \cdot h_k$$

$$h_k = \frac{1}{3} \sum_{I=1}^n \phi_I(dx \cdot dy \cdot dz)$$

Combinando as equações Eq. (4.46) e Eq. (4.45), tem-se que o fator deve ser necessariamente aquele apresentado na Eq. (4.45), na qual \bar{V} é o valor médio dos volumes eulerianos associados ao ponto lagrangiano x_k . Uma vez calculada a força devida à interface imersa no escoamento, pode-se atualizar a componente de velocidade estimada, agora com a influência do corpo imerso como:

$$u_i^* = \tilde{u}_i + f_i \Delta t \quad (4.47)$$

e então o campo pode ser recorrigido de maneira a satisfazer a conservação da massa,

$$u_i^{n+1} = u_i^* - \Delta t \frac{\partial}{\partial x_i}(\delta p) \quad (4.48)$$

4.3 Solução do Sistema de Equações Lineares

Com o aumento da complexidade e do número de graus de liberdade dos problemas que tem sido resolvidos numericamente nos últimos anos, o acoplamento pressão-velocidades, que dá origem à Eq. (4.19), se tornou o foco de um intenso esforço dedicado à criação de métodos e algoritmos para acelerar o processo de solução, não somente em sistemas com acoplamento, mas também segregados (Van Doormaal; RAITHBY, 1985). No presente trabalho, a discretização da Eq. (4.19) leva a um sistema de equações lineares para determinar a correção de pressão ϕ . Em malhas AMR, a solução para a equação de Poisson é geralmente realizada empregando-se algum processo iterativo, aliado a técnicas de multigrid (MG).

O método de multigrid demonstrou ser uma das melhores ferramentas desenvolvidas nas últimas décadas para aceleração da solução da equação de Poisson, tanto a versão geométrica (VILLAR, 2007; CENICEROS et al., 2010) como a algébrica (AMG) (STÜBEN, 2001; HAASE et al., 2001). No AMG, são utilizados mecanismos semelhantes àqueles do MG, mas montados em uma estrutura totalmente algébrica. Isso permite que o método possa ser empregado na solução de sistemas lineares de forma muito eficiente e robusta, combinando algoritmos iterativos para amortecer erros de alta frequência enquanto transfere os erros de baixa frequência na malha fina para erros de alta frequência na malha grossa.

Devido aos tipos de problemas abordados neste trabalho, para solução da equação de Poisson será utilizado o AMG, atuando como preconditionador para os métodos de *Gradiente Bi-Conjugado Estabilizado* (BiCGStab) e *Resíduo Mínimo Generalizado* (GMRES), (SAAD; SCHULTZ, 1986). Esses métodos serão utilizados à partir dos pacotes para preconditionamento e solução paralela de sistemas lineares *ML* e *AztecOO*, contidos na biblioteca de programação científica *Trilinos* (HEROUX et al., 2003). Além de possuir código aberto e orientação a objetos, esta biblioteca contém uma coleção de pacotes para simulação de problemas multifísicos em larga-escala além de recursos como pré-condicionadores pré-definidos e/ou personalizados pelo usuário. Como exemplo, no balanço de carga para a malha retilínea foi utilizado outro pacote do Trilinos, chamado *Zoltan*, apresentado na Seção 2.2.

Os solvers apresentados serão utilizados para solução paralela da equação de Poisson no aplicativo desenvolvido. Para a malha adaptativa, dois algoritmos foram utilizados e serão discutidos nas seções a seguir. No primeiro, a solução será nível a nível, enquanto que no segundo será montada uma matriz global para o sistema.

Nas equações relativas à malha octree, a seguinte nomenclatura é utilizada: $\varphi_{ijk}^{c,l}$ significa a variável φ na célula c , nível l e na posição indicial ijk , respectivamente nas três direções

coordenadas xyz . $\mathcal{M}(c)$ denota uma célula que é um nível mais grossa que c e na qual c está contida. $Q(c, i)$ faz referência à filha da célula c de numeração i , com i variando de 0 à $(2^D - 1)$, na qual D é a dimensão da malha. Assim, a equação de Poisson (Eq. 4.19) a ser resolvida pode ser reescrita na forma,

$$\begin{aligned}
 \mathcal{L}_l \tilde{\phi}_{ijk}^{cl} &= Q_{ul}^c \\
 \mathcal{L}_l \tilde{\phi}_{ijk}^{cl} &\equiv \frac{1}{\Delta x_l^2} (\tilde{\phi}_{i+1,jk}^{cl} - 2\tilde{\phi}_{i,jk}^{cl} + \tilde{\phi}_{i-1,jk}^{cl}) + \\
 &\quad \frac{1}{\Delta y_l^2} (\tilde{\phi}_{i,j+1,k}^{cl} - 2\tilde{\phi}_{i,j,k}^{cl} + \tilde{\phi}_{i,j-1,k}^{cl}) + \\
 &\quad \frac{1}{\Delta z_l^2} (\tilde{\phi}_{ij,k+1}^{cl} - 2\tilde{\phi}_{ij,k}^{cl} + \tilde{\phi}_{ij,k-1}^{cl}) \\
 Q_{ul}^c &\equiv \frac{\rho}{\Delta t} \left[\frac{(u_{i+1,jk} - u_{ijk})}{\Delta x} + \frac{(v_{i,j+1,k} - v_{ijk})}{\Delta y} + \frac{(w_{ij,k+1} - w_{ijk})}{\Delta z} \right]^{cl}
 \end{aligned} \tag{4.49}$$

na qual $\tilde{\phi}$ é a forma discretizada da correção de pressão ϕ .

4.3.1 Solução de Poisson por níveis de malha

O método de solução empregado nesta seção é apresentado em (FRYXELL et al., 2001) que, por sua vez, é baseado no algoritmo de Huang e Greengard (2000). Comparado ao multigrid tradicional, é aplicada uma solução direta em cada bloco (ou *patch*) no nível e o resíduo global é minimizado à medida em que as informações são trocadas entre os diferentes níveis de malha.

Como está sendo utilizado o método de volumes finitos, o operador de restrição \mathcal{R}_l , para uma variável co-localizada φ , pode ser definido diretamente como

$$(\mathcal{R}_l \varphi)_{IJK}^{\mathcal{M}(c),l} \equiv \frac{1}{2^D} \sum_{ijk} \varphi_{ijk}^{c,l+1} \tag{4.50}$$

na qual ijk são índices de células que estão contidas em IJK . O operador restrição é aplicado em folhas que são filhas de uma fantasma com nível $l - 1$, como ilustrado na Fig. 2.8b. Na mesma figura, para a operação inversa, ou seja, calcular o valor de uma fantasma no nível $l + 1$ filha de uma folha em l , é utilizado o operador prolongamento:

$$(\mathcal{P}_l \varphi)_{ijk}^{c,l+1} \equiv \sum_{p,q,r=-1}^1 \alpha_{IJK,pqr} \varphi_{I+p,J+q,K+r}^{\mathcal{M}(c),l} \quad (4.51)$$

na qual α é o coeficiente de interpolação. Por exemplo, para a face $-1/2i$ temos

$$\begin{aligned} \alpha_{(-\frac{1}{2}I,J,K,pqr)} &= \beta_p \cdot \gamma_{Jq} \cdot \gamma_{Kr} \\ \beta_p &= \left(\frac{60}{32}, -\frac{40}{32}, \frac{12}{32} \right) \\ \gamma_{q^*J^*} &= \begin{cases} \left(\frac{21}{32}, \frac{14}{32}, -\frac{3}{32} \right), & \text{se } J^* \text{ é ímpar} \\ \left(\frac{45}{32}, -\frac{18}{32}, \frac{5}{32} \right), & \text{ou se } J^* \text{ é par} \end{cases} \end{aligned} \quad (4.52)$$

Uma aproximação para o erro na solução do sistema linear pode ser dada pelo resíduo a partir da Eq. (4.49) como:

$$R_{ijk}^{cl} = Q_{ul}^c - \mathcal{L}_l \tilde{\phi}_{ijk}^{cl} \quad (4.53)$$

Uma vez que a equação de Poisson é linear, podemos definir C_{ijk}^{cl} como sendo a correção que precisa ser adicionada à $\tilde{\phi}_{ijk}^{cl}$ para determinar a solução corrigida ϕ_{ijk}^{cl} , de tal forma que

$$\mathcal{L}_l C_{ijk}^{cl} = R_{ijk}^{cl} \quad (4.54)$$

cujas estrutura é semelhante àquela da Eq. 4.49.

A solução consiste primeiro em se calcular o termo do lado direito da Eq. 4.49 para todas as células que são folhas da árvore, ou seja, aquelas que não possuem nenhuma célula de nível maior embutidas em seu interior. Esse termo é restringido para todos os níveis de malha que existirem, até o nível base, abaixo do qual ele foi calculado. A partir do nível base, a equação é resolvida para cada nível, e calculado o resíduo, sendo que o resultado é interpolado utilizando interpolação conservativa quadrática para as fronteiras do próximo nível em refinamento, para que esse possa ser resolvido na sequência. O resíduo é então restringido para todas as células de nível menor que o da célula que o contém, até o nível base.

Nesta fase, o resíduo deve ter um valor considerável apenas nas regiões próximas às fronteiras de malha fina-grossa. Esse resíduo é utilizado como termo fonte para resolver a equação de correção e então corrigir a malha no nível l . Os valores corrigidos são então interpolados como condições de contorno de Dirichlet e utilizados para resolver a equação de correção em $l + 1$. O processo continua até que o resíduo atinja um valor dentro de uma tolerância especificada. O algoritmo é discutido em mais detalhes em (RICKER, 2008) e aqui será apresentado apenas um resumo dos passos de implementação:

1. *Restrição* - Primeiro restrinja o termo fonte Q_{ul}^c em todos os níveis, de tal forma que ele esteja definido em todos os blocos.
2. *Interpolação* - Para cada nível l , de 0 até o número máximo de níveis menos 1:
 - (a) Resolva a Eq. 4.49 para o nível l .
 - (b) Calcule o resíduo (Eq. 4.53) para o nível l . Ao impor as condições de contorno ignore os níveis $> l$ e interpole das células vizinhas com nível $< l$.
 - (c) Para cada célula c que contenha um nível mais refinado em seu interior, interpole os valores para as condições de contorno desse nível.
3. *Resíduo* - Restrinja o resíduo em todos os níveis de tal forma que R_{ijk}^{cl} seja um valor calculado na célula folha ou então um valor restringido de um nível maior.
4. *Correção* - Calcule a norma discreta L2 do resíduo em todas as células folhas e divida esta pela norma discreta L2 do termo fonte sobre as mesmas células. Se o valor for maior que uma tolerância pré-determinada, então é necessário um passo de correção, no qual, para cada nível l , de 0 até o número máximo de níveis menos 1:
 - (a) Resolva a Eq. 4.54 para o nível l .
 - (b) Substitua o valor de R_{ijk}^{cl} com o novo resíduo $R_{ijk}^{cl} - \mathcal{L}_l C_{ijk}^{cl}$ no nível l . Ao impor as condições de contorno ignore os níveis $> l$ e interpole das células vizinhas com nível $< l$.
 - (c) Corrija a solução em todas as células folhas no nível l com $\tilde{\phi}_{ijk}^{cl} = \tilde{\phi}_{ijk}^{cl} + C_{ijk}^{cl}$.
 - (d) Para cada célula c que contenha um nível mais refinado em seu interior, interpole C_{ijk}^{cl} constituir as condições de contorno desse nível.
5. Retorne ao passo 3.

No código de Ricker (2008) a solução direta das equações era aplicada em cada um dos blocos da malha para todos os níveis. No presente trabalho, é aplicado uma solução iterativa independente para cada nível de malha. A solução nos diferentes níveis, em toda a malha, é obtida através da utilização de uma camada de volumes fantasmas na periferia de um bloco de malha com nível n . As células fantasmas são preenchidas com valores interpolados a partir dos blocos de malhas adjacentes.

4.3.2 Solução de Poisson global

Uma outra aproximação utilizada para resolver a equação de Poisson na malha AMR consiste em reescrever as equações na interface fina-grossa em função das interpolações para as células fantasmas nessas regiões. Entretanto, aqui as interpolações serão diferentes daquelas apresentadas na seção anterior, pois a conservação dos fluxos de ϕ se torna crítica para solução do sistema linear final. Além disso, o estêncil das interpolações afetará diretamente, não somente a esparsidade, mas também a estrutura da matriz. Os procedimentos adotados para essa aproximação serão ilustrados com o auxílio da Fig. 4.5. A área sombreada em azul compreende o estêncil de interpolação para $\phi_{I-1,J}^l$, enquanto que as linhas tracejadas, verde e vermelha, estão no estêncil para a interpolação do tipo fina-grossa.

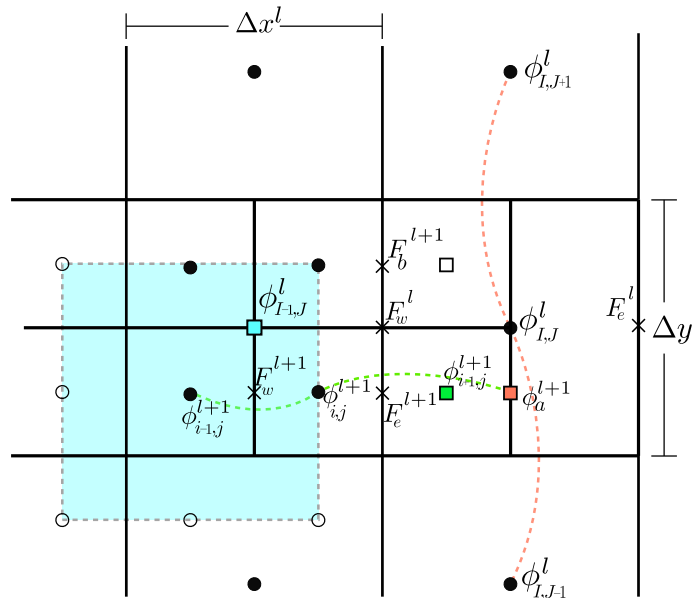


Figura 4.5: Localização dos pontos e do estêncil das interpolações para cálculo do Laplaciano de ϕ , na interface fina-grossa, em duas dimensões.

Por simplicidade, será omitido o parentesco entre células, c , e a Eq. (4.49) será rescrita

para o caso bidimensional como,

$$\mathcal{L}_l \tilde{\phi}_{ij}^l = Q_u^l$$

Utilizando a Fig. 4.5, essa equação pode ser discretizada em função dos fluxos de ϕ nas faces. Assumindo que os lados da malha são uniformes para um dado nível l e iguais a Δx_l e Δy_l nas direções x e y , respectivamente, tem-se:

$$\mathcal{L}_l \tilde{\phi}_{ij}^l = \left[\frac{(F_e - F_w)}{\Delta x_l} \right] + \left[\frac{(F_n - F_s)}{\Delta y_l} \right] \quad (4.55)$$

na qual as derivadas nas faces F_e , F_w , F_n e F_s , são calculadas por diferenças centradas. Porém, fluxos como F_e^{l+1} , na face e da célula com valor $\phi_{i,j}^{l+1}$, dependem de uma fantasma, nesse caso do valor de $\phi_{i+1,j}^{l+1}$,

$$F_e^{l+1} = \frac{(\phi_{i+1,j}^{l+1} - \phi_{i,j}^{l+1})}{\Delta x^{l+1}} \quad (4.56)$$

O valor de $\phi_{i+1,j}^{l+1}$ é calculado ajustando-se uma função quadrática passando por ϕ_a^{l+1} , calculado no estêncil da malha no nível l , como mostrado na Fig 4.5 e os valores das folhas em $l+1$, $\phi_{i,j}^{l+1}$ e $\phi_{i-1,j}^{l+1}$. Para calcular ϕ_a^{l+1} , utiliza-se a seguinte expressão quadrática,

$$\phi(x) = \beta_2 x^2 + \beta_1 x + \beta_0 \quad (4.57)$$

cujos coeficientes β podem ser determinados utilizando os valores conhecidos na malha *grossa* (nível l). Adotando (x_0, y_0) , como as coordenadas de $\phi_{I,J}^l$ pode se escrever as expressões para cada um dos pontos como,

$$\begin{aligned} \beta_0 + \beta_1(y_0 - \Delta y) + \beta_2(y_0 - \Delta y)^2 &= \phi_{I,J-1}^l \\ \beta_0 + \beta_1 y_0 + \beta_2 y_0^2 &= \phi_{I,J}^l \\ \beta_0 + \beta_1(y_0 + \Delta y) + \beta_2(y_0 + \Delta y)^2 &= \phi_{I,J+1}^l \end{aligned} \quad (4.58)$$

Uma vez que ϕ é a correção de pressão, para que suas derivadas sejam compatíveis com o operador restrição dado pela Eq. (4.50), é necessário que o fluxo de ϕ na interface, sentido pela malha grossa, seja o mesmo fluxo percebido pela fina, ou seja,

$$F_w^l = \frac{(F_e^{l+1} + F_b^{l+1})}{2} \quad (4.59)$$

O conjunto de equações (4.58) pode ser resolvido para encontrar os coeficientes da Eq. (4.57), a partir da solução do seguinte sistema linear,

$$\begin{bmatrix} 1 & y_0 - \Delta y & (y_0 - \Delta y)^2 \\ 1 & y_0 & y_0^2 \\ 1 & y_0 + \Delta y & (y_0 + \Delta y)^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \phi_{I,J-1}^l \\ \phi_{I,J}^l \\ \phi_{I,J+1}^l \end{bmatrix} \quad (4.60)$$

Com o valor de ϕ_a^{l+1} , calculado à partir da malha grossa, é possível interpolar $\phi_{i+1,j}^{l+1}$ novamente ajustando uma parábola passando pelos pontos ϕ_a^{l+1} , $\phi_{i,j}^{l+1}$ e $\phi_{i-1,j}^{l+1}$ como:

$$\phi_{i+1,j}^{l+1} = (8\phi_a^{l+1} + 10\phi_{i,j}^{l+1} - 3\phi_{i-1,j}^{l+1})/15 \quad (4.61)$$

Assumindo que os fluxos na direção y, F_s e F_n já tenham sido adequadamente calculados em ambas as malhas, a forma discretizada do laplaciano para ϕ_{ij}^{l+1} fica,

$$\mathcal{L}_l \phi_{ij}^{l+1} = \left[\frac{(F_e^{l+1} - F_w^{l+1})}{\Delta x_l^{l+1}} \right] + \left[\frac{(F_n^{l+1} - F_s^{l+1})}{\Delta y_l^{l+1}} \right] \quad (4.62)$$

enquanto que para a variável na malha grossa, ϕ_{ij}^l é escrita como,

$$\begin{aligned}
\mathcal{L}_l \phi_{ij}^l &= \left[\frac{(F_e^l - F_w^l)}{\Delta x_l^l} \right] + \left[\frac{(F_n^l - F_s^l)}{\Delta y_l^l} \right] \\
&= \left[\frac{\left(F_e^l - \frac{(F_e^{l+1} - F_b^{l+1})}{2} \right)}{\Delta x_l^l} \right] + \left[\frac{(F_n^l - F_s^l)}{\Delta y_l^l} \right]
\end{aligned} \tag{4.63}$$

Quando essas equações forem escritas na forma de um sistema de equações lineares, as expressões (4.57) e (4.61) devem ser incluídas integralmente, de forma que a própria solução, seja esta iterativa ou direta, possa atualizar os valores fantasmas. A forma final das equações para solução de Poisson, na forma matricial, pode ser dada como:

$$\mathbf{A}\Phi = \mathbf{b} \tag{4.64}$$

permitindo que qualquer *solver* possa ser utilizado, até mesmo na forma de uma caixa-preta. Para certas configurações, como as apresentadas na Fig. 4.6, o estêncil precisa ser tomado de células adjacentes ou então uma outra ordem de interpolação deve ser utilizada, como a linear apresentada na Fig. 4.6b.

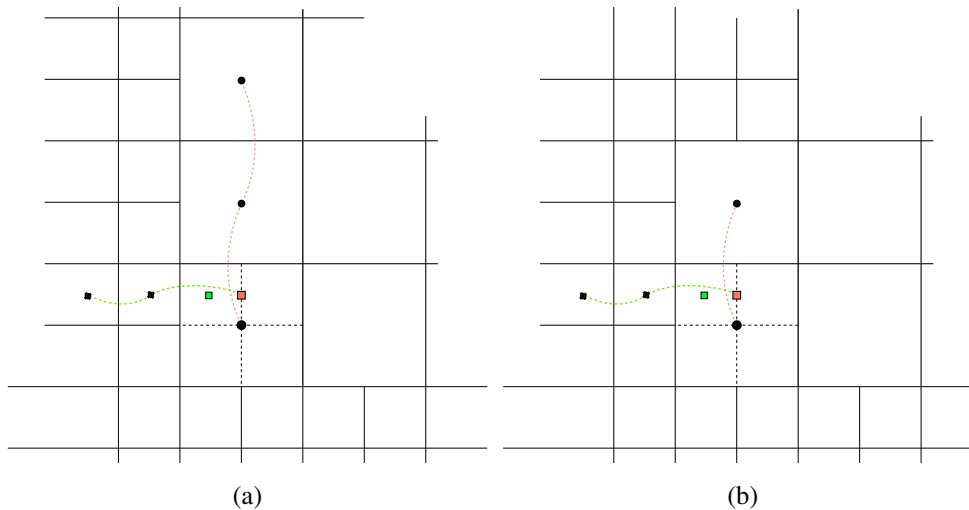


Figura 4.6: Configurações da malha para variáveis centradas. Quando não se dispõe do estêncil padrão, é necessário um conjunto diferente de constantes de interpolação para células que necessitam de uma interpolação diferenciada. Dois casos são ilustrados no exemplo: (a) quadrática deslocada e (b) linear.

4.4 Restrição e prolongamento em variáveis desencontradas

No presente trabalho, variáveis vetoriais representadas pelas componentes do campo de velocidades, u , v e w , respectivamente nas direções coordenadas x , y e z , são posicionadas nas faces dos volumes, enquanto os escalares são centrados nos centros dos volumes, como a pressão. Uma preocupação frequente quando se lida com escoamentos de fluidos incompressíveis em malhas AMR, é garantir a conservação da massa, Eq. (4.2).

A forma de construir as interpolações, apresentada na seção anterior, para variáveis colocalizadas, já é apropriada para que, no momento da correção da velocidade (Eq. 4.20), seja garantida a conservação da massa. Entretanto, para calcular os valores das velocidades nas células fantasmas, é necessário que os operadores de restrição e de prolongamento mantenham essa conservação, previamente imposta pela correção de pressão.

Para variáveis desencontradas, definindo a operação de restrição, a média entre os fluxos em $l + 1$ de determinada face em l se mostrou suficientemente eficaz para assegurar a conservação de massa na interface. Para o caso bidimensional e com razão de refinamento de $1 : 2$, mostrado na Fig 4.7, o valor da velocidade u_{ij}^l na malha grossa será simplesmente,

$$u_{ij}^l = \frac{\Delta y^{l+1}}{\Delta y^l} (u_{ij-1}^{l+1} + u_{ij+1}^{l+1}) = \frac{u_{ij-1}^l + u_{ij+1}^l}{2} \quad (4.65)$$

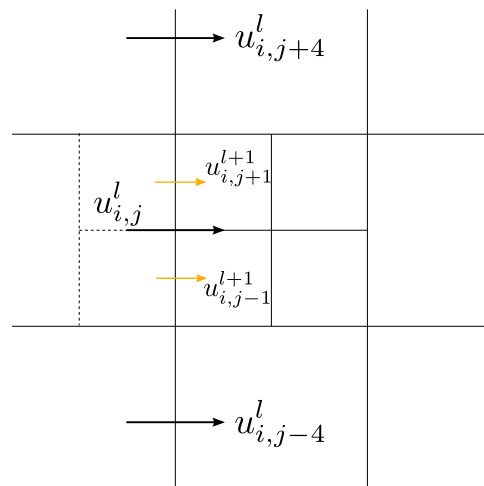


Figura 4.7: Componentes de velocidade u , na interface fina-grossa.

A operação de prolongamento para as velocidades, em $l + 1$, nas faces de uma célula mãe em l é implementada de maneira semelhante àquela para variáveis colocalizadas. Para as velocidades do interior do bloco, ilustradas em amarelo na Fig. 4.8, é necessária uma atenção

especial. Para o primeiro caso, é assumido que as velocidades na malha fina possam ser descritas a partir de uma função quadrática, diga-se,

$$u(y) = \alpha_3 y^2 + \alpha_2 y + \alpha_1 \quad (4.66)$$

de tal forma, que as velocidades $u_{i,j-1}^{l+1}$ e $u_{i,j+1}^{l+1}$, na Fig 4.7, possam ser escritas em função de $u_{i,j-4}^l$, $u_{i,j}^l$ e $u_{i,j+4}^l$. Assumindo que $h = \frac{\Delta y^{l+1}}{2}$, as seguintes expressões podem ser escritas para u na malha grossa:

$$\begin{aligned} u_{i,j-4}^l &= \alpha_1 + \alpha_2 2h + \alpha_3 4h^2 \\ u_{i,j}^l &= \alpha_1 + \alpha_2 6h + \alpha_3 37h^2 \\ u_{i,j+4}^l &= \alpha_1 + \alpha_2 10h + \alpha_3 100h^2 \end{aligned} \quad (4.67)$$

onde, de modo a impor conservatividade na interpolação, Eq. (4.66), na segunda equação é aplicada a operação de restrição fina-grossa,

$$u_{i,j}^l = \frac{u_{i,j-1}^{l+1} + u_{i,j+1}^{l+1}}{2} \quad (4.68)$$

Os coeficientes α_1 , α_2 e α_3 são encontrados resolvendo um sistema parecido com o da Eq. (4.60), construído a partir da Eq. (4.68). Para determinação das velocidades no interior do bloco, emprega-se as velocidades já calculadas nas faces e a equação da continuidade em sua forma discreta escrita, em um nível l qualquer, como:

$$\mathcal{L}(\mathbf{u})^l = \frac{u_{i,j}^l - u_{i-1,j}^l}{\Delta x^l} + \frac{v_{i,j}^l - v_{i,j-1}^l}{\Delta y^l} \quad (4.69)$$

A Eq. (4.69) é avaliada para as quatro folhas no bloco, dando origem à um sistema de quatro equações a quatro incógnitas, na forma $\mathbf{Ax} = \mathbf{b}$ da seguinte equação:

$$\begin{bmatrix} 1 & 0 & \delta & 0 \\ -1 & 0 & 0 & \delta \\ 0 & 1 & -\delta & 0 \\ 0 & -1 & 0 & -\delta \end{bmatrix} \begin{bmatrix} u_{i,j-1}^{l+1} \\ u_{i,j+1}^{l+1} \\ v_{i-1,j}^{l+1} \\ v_{i+1,j}^{l+1} \end{bmatrix} = \begin{bmatrix} u_{i-1,j-1}^{l+1} + \delta v_{i-1,j-1}^{l+1} \\ -u_{i+1,j-1}^{l+1} + \delta v_{i+1,j-1}^{l+1} \\ u_{i-1,j+1}^{l+1} - \delta v_{i-1,j+1}^{l+1} \\ -u_{i+1,j+1}^{l+1} - \delta v_{i+1,j+1}^{l+1} \end{bmatrix} \quad (4.70)$$

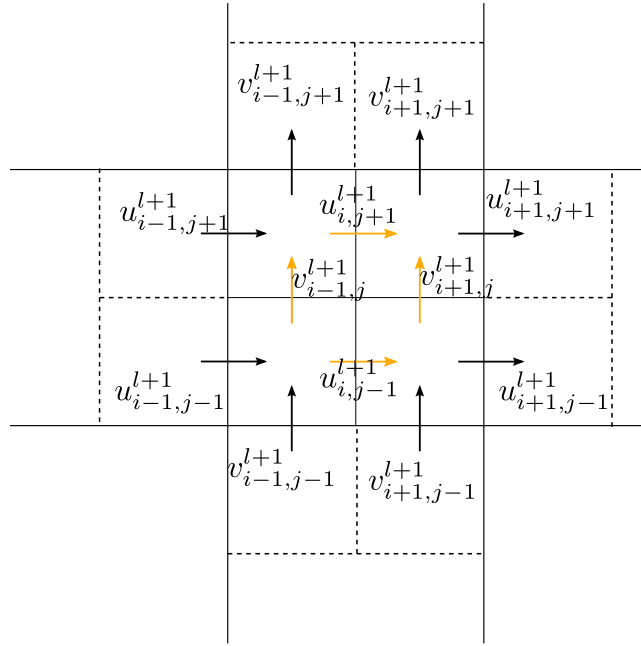


Figura 4.8: Disposição das componentes de velocidade em uma malha bidimensional. As setas amarelas denotam as variáveis que precisam ser resolvidas simultaneamente.

Contudo, uma vez que cada uma das 4 velocidades depende de seu próprio valor, existe mais de uma solução possível, tornando a matriz \mathbf{A} singular. Para determinar \mathbf{x} , uma equação adicional independente pode ser adicionada ao sistema, em função de qualquer um dos termos procurados. Por exemplo, para uma das componentes u (VANELLA; RABENOLD; BALARRAS, 2010),

$$u_{i,j-1}^{l+1} = \frac{1}{8} (6u_{i-1,j-1}^{l+1} + 3u_{i+1,j-1}^{l+1} - u_{i-3,j-1}^{l+1}) \quad (4.71)$$

O sistema pode ser reescrito como:

$$\begin{bmatrix} 1 & 0 & \delta & 0 \\ -1 & 0 & 0 & \delta \\ 0 & 1 & -\delta & 0 \\ 0 & -1 & 0 & -\delta \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{i,j-1}^{l+1} \\ u_{i,j+1}^{l+1} \\ v_{i-1,j}^{l+1} \\ v_{i+1,j}^{l+1} \end{bmatrix} = \begin{bmatrix} u_{i-1,j-1}^{l+1} + \delta v_{i-1,j-1}^{l+1} \\ -u_{i+1,j-1}^{l+1} + \delta v_{i+1,j-1}^{l+1} \\ u_{i-1,j+1}^{l+1} - \delta v_{i-1,j+1}^{l+1} \\ -u_{i+1,j+1}^{l+1} - \delta v_{i+1,j+1}^{l+1} \\ \frac{1}{8} (6u_{i-1,j-1}^{l+1} + 3u_{i+1,j-1}^{l+1} - u_{i-3,j-1}^{l+1}) \end{bmatrix} \quad (4.72)$$

que é um sistema de cinco equações a quatro incógnitas que pode ser resolvido na forma,

$$\mathbf{x} = \mathbf{A}_m \mathbf{b} \quad (4.73)$$

na qual,

$$\mathbf{A}_m = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (4.74)$$

é a inversa de *Moore-Penrose* do sistema 4×5 . A extensão dos procedimentos acima para o caso tridimensional, implementado no presente trabalho, pode ser feita de maneira direta. Entretanto, as equações acima foram determinadas utilizando o pacote de linguagem simbólica da linguagem Python, chamado *sympy* e as expressões resultantes, mesmo simplificadas são muito grandes, o que prejudica testes preliminares e garimpagem por erros no programa.

Aqui foram utilizadas as simplificações apresentadas no trabalho de Tóth e Roe (2002), no qual essas expressões para as velocidades internas no bloco são escritas em função das derivadas de primeira, segunda e terceira ordens, tornando a implementação mais compacta e simples para otimizar. Por exemplo, utilizando essa aproximação, a velocidade interpolada $u_{i,j-1,k}^{l+1}$ pode ser escrita na forma:

$$u_{i,j-1,k}^{l+1} = \frac{1}{2} (u_{i-1,j-1,k}^{l+1} + u_{i+1,j-1,k}^{l+1}) + 4 (U_{xx} + k\Delta z^2 V_{xyz} + j\Delta y^2 W_{xyz}) \quad (4.75)$$

na qual U_{xx} , a derivada segunda de u^l na direção x , e os termos de terceira ordem na forma U_{xyz} , dependem do posicionamento e do estêncil das células vizinhas de mesmo nível. Pode se notar que mesmo para o caso tridimensional a equação resultante é uma expressão simples de implementar. Essa forma compacta pode ser obtida substituindo na expressão simbólica, retornada pela Eq. (4.73), as expressões para as derivadas segunda e terceira e fazendo algumas simplificações.

4.5 Paralelização

Paralelizar algoritmos numéricos visando a solução de problemas em mecânica dos fluidos se tornou uma implementação quase mandatória em problemas de médio e de grande porte. Isso acontece porque a solução das equações de Navier-Stokes, mesmo para um problema

simplificado, pode se tornar uma tarefa computacionalmente intensiva para certas aplicações. Embora atualmente existam técnicas sofisticadas para esse tipo de prática, a paralelização realizada no presente trabalho pode ser caracterizada como um simples paralelismo geométrico, no qual o conjunto de dados a ser processado é dividido em tarefas que, por sua vez, são executadas por cópias do código que estão em diferentes processadores. Nesta seção serão apresentadas as aproximações empregadas para paralelizar o método de fronteira imersa e a malha com refinamento adaptativo.

Nos blocos, formados por células da árvore, que se comportam como um conjunto de malhas cartesianas, as células fantasmas simplesmente copiam os valores quando possuem suas correspondentes em um outro processador, como apresentado na Fig 4.9. Para isto, são criados dois mapas, um local e um global. No mapa local os volumes são enumerados de 0 a Nl_{ip} , onde Nl_{ip} é o número de células que pertencem ao processador ip . O mapa global é composto de inteiros que variam de 0 a Ng_t , com Ng_g sendo a soma das células em todos os processadores. Na contagem das células, em ambos os mapas, as células fantasmas não são inseridas, uma vez que compartilham a mesma posição geométrica de outras células. Todavia, as células fantasmas nos contornos do domínio, utilizadas para imposição de condições de contorno e, portanto, não partilhando dados com outro processador, são contadas no mapa local.

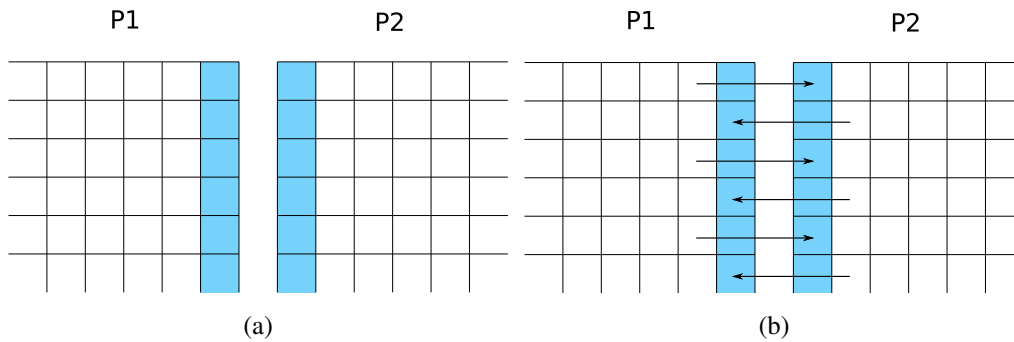


Figura 4.9: Atualização feita por simples injeção de células fantasmas, destacadas na figura, para a decomposição da malha cartesiana em dois domínios (ou processadores) $P1$ e $P2$.

A matriz, montada a partir da Eq. (4.49), é enumerada segundo o mapa global, enquanto que as listas contendo a pressão e o campo de velocidades, são especificadas de acordo com o mapa local. Isso é feito porque o mapa local, diferentemente do global, possui células fantasmas. No processo de solução da Eq. (4.49), o termo do lado direito é avaliado no mapa local, e uma vez resolvido o sistema linear, a solução ϕ é copiada do mapa global. Assim, exceto pelo sistema linear, todos os cálculos são praticamente realizados somente no mapa local, para ambas as malhas apresentadas nesse trabalho.

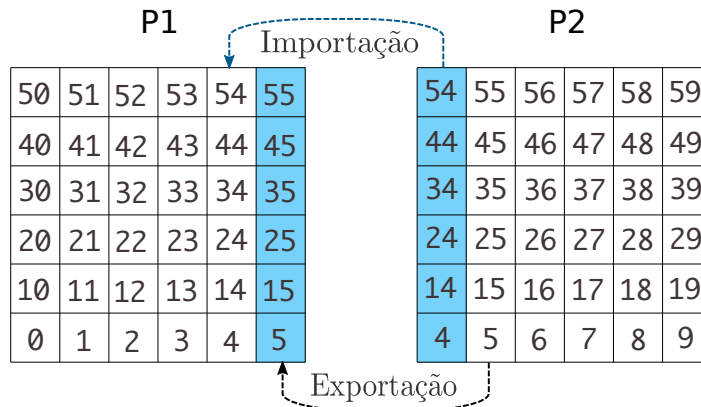


Figura 4.10: Numeração no mapa local e operação de importação e exportação de dados entre dois processos. Um mapa é utilizado para ambas as operações no mapa local, que envolve todas as células

A partir da criação de sub-mapas, contendo apenas células fantasmas ou que precisam de algum tipo de atualização, é possível definir as operações de *Exportação* e de *Importação*. Na primeira, um valor definido no processo $P1$ é enviado para sua fantasma correspondente em um outro processo $P2$. Na importação, um valor definido em uma fantasma no processo $P2$ é atualizado com o valor da célula correspondente no processo $P1$. Um diagrama mostrando as operações de importação e exportação de dados em um mapa local é apresentado na Fig. 4.10.

A operação de importação é especialmente útil na paralelização da metodologia com fronteira imersa. Para os métodos apresentados na Seção 2.4, é necessária uma camada de células fantasmas por bloco. Foi utilizada uma malha lagrangiana composta por triângulos, nos quais as variáveis lagrangianas como força e velocidade, são localizadas nos vértices (os pontos lagrangianos).

No presente trabalho, o particionamento da malha lagrangiana é feito a partir dos limites geométricos dos blocos nos diferentes processadores. Assim, se um elemento triangular possui ao menos um vértice que faça parte do bloco P_{ip} , pertencente ao processo ip , então este elemento é atribuído à malha lagrangiana naquele processo. O resultado é uma malha semelhante àquela representada na Fig. 4.11.

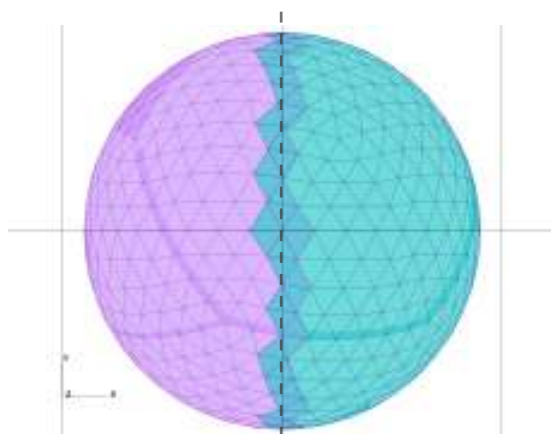


Figura 4.11: Casca esférica não-estruturada, composta por elementos triangulares e particionada em dois domínios. A linha tracejada situa-se na fronteira entre os domínios.

A transferência do elemento, ao invés do nó, para outro processo, permite que os nós sejam organizados em função das conectividades dos triângulos. Isso também contribui para o cálculo do volume lagrangiano, que no presente trabalho é calculado em função dos triângulos vizinhos. Aqui os pontos lagrangianos são tomados principalmente devido à posição das variáveis desconstruídas. Uma vez que as células fantasmas eulerianas estiverem corretas, os valores interpolados lagrangianos não precisarão de atualização entre os processos.

Entretanto, a força distribuída na malha euleriana precisa ser atualizada pela soma das contribuições provenientes das células fantasmas dos diferentes processos envolvidos na simulação. Para isso, utiliza-se agora a operação de importação, que atua como uma operação inversa de exportação na qual, a cada iteração, soma-se os valores das fantasmas àqueles em suas relativas posições geométricas em outro processo. Para esse caso, seria como se as setas na Fig. 4.9b estivessem com sua direção invertida.

Uma vez que estamos trabalhando com refinamento adaptativo, os mapas são geometricamente mais complexos. Contudo, os mesmos operadores de importação e de exportação descritos para um caso simples acima, continuam sendo utilizados sendo que, nesse caso, devido à existência de diferentes níveis de malha e de operações como restrição e prolongamento, essas atualizações precisam ser feitas de maneira sincronizada, como apresentado na Fig 4.12. A enumeração para as células finas se encontram no canto superior direito e na malha grossa no canto inferior esquerdo. As células fantasmas na interface são ilustradas para os dois níveis por linhas tracejadas.

Como mostrado na Fig 4.12, na malha grossa ($l = 0$) a enumeração é feita no canto inferior esquerdo e na malha fina, no canto superior direito, de acordo com a seguinte ordem de

		39	22	23	26	27
10	11	38	20	21	24	25
		37	14	15	18	19
8	9	36	12	13	16	17
		31	32	33	34	35
2	3					
0	1					

Figura 4.12: Malha com um nível de refinamento, mapeada em dois processadores.

prioridade:

- Enumera-se primeiro as células que não são fantasmas;
- A enumeração é iniciada no menor nível ($l = 0$);
- Entre todos os níveis a enumeração é iniciada no processo raiz ($ip = 0$);
- Aplica-se os mesmos procedimentos acima para as células fantasmas.

Além dos mapas global e local, outros três mapas são muito importantes aqui:

Folhas contém as células fantasmas de mesmo nível a serem atualizadas por simples injeção de dados de outro processo. Na Fig 4.12, as células 14, 15, 16, 18, no processo $P2$ são atualizadas como fantasmas das células 12, 13 no processo $P1$;

Grossas contém as componentes fantasmas de células grossas que devem ser calculadas em outro processo antes de serem exportadas, enumeradas como 28, 29, 30 na figura;

Finas contém as fantasmas de células finas que também precisam ser calculadas em outro processo antes de serem exportadas, compostas por 34, 35, 37, 38, 39 e calculadas em $P1$ antes de serem exportadas para $P2$.

Na simulação, atualizam-se primeiro as Folhas. Em seguida, aplica-se a operação de restrição às fantasmas grossas locais. Devido ao estêncil adotado, pode-se então calcular as células que estão no mapa de Grossas, também aplicando uma restrição, mas no processo que possui as células. Exportam-se as Grossas e o operador de prolongamento é aplicado localmente, para

calcular as finas locais. Então, as células fantasmas Finas, que podem depender tanto das Folhas como das Grossas, também são calculadas e exportadas.

A vantagem de trabalhar com essa forma ordenada, é o fato de que somente as células fantasmas serão importadas e exportadas entre diferentes processadores, ao invés de todo o estêncil necessário para seu cálculo, o que aumenta consideravelmente a relação *processamento/transferência de dados* na simulação.

4.5.1 Estrutura do aplicativo

As operações na malha octree foram condensadas em uma biblioteca. O usuário pode incluir o cabeçalho dessa biblioteca e então fazer as chamadas das principais funções de maneira quase transparente. Essas funções, apresentadas na forma de um diagrama na Fig. 4.13, são:

Inicialização : na qual o usuário entra com os dados iniciais;

Mapas : Os mapas retornados após a inicialização, com os quais se criam os vetores e matrizes;

Atualização : Atualiza as malhas fantasmas em operações que podem ser:

- locais, pela imposição de condições de contorno ou por interpolação na interface fina-grossa;
- em outro processo, no qual, além das operações locais, também existe a injeção de valores para um mesmo nível.

Refinamento : Refina a malha com base em determinado critério e salva os dados em disco, atualizando os mapas;

Salvar : Salva um determinado vetor no formato *Silo*;

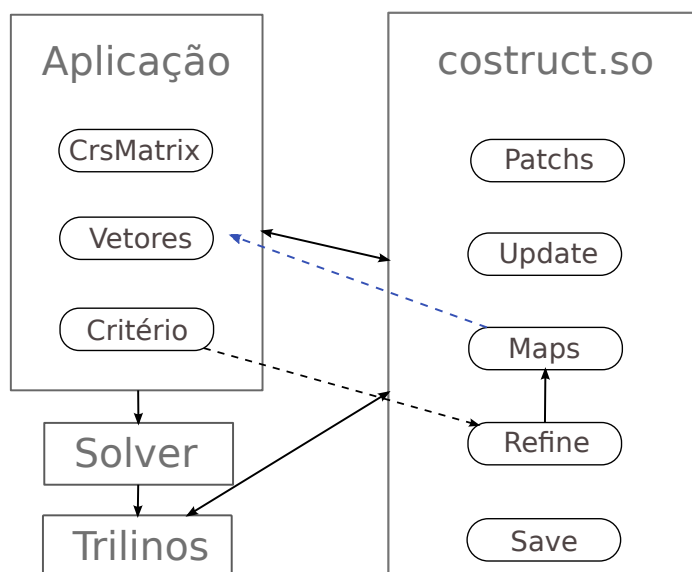


Figura 4.13: Diagrama simplificado das ligações entre as principais funções do aplicativo. *costruct.so* é a biblioteca de malha cartesiana estruturada por uma octree. *Solver* é a extensão para resolver a equação de Poisson. *Trilinos* é a biblioteca contendo os pacotes utilizados no aplicativo. Aplicação pode ser um arquivo do tipo *main.c* que gerencia a simulação.

Nos testes realizados, a partir dos mapas, são criados os vetores e matrizes, já de forma paralela. A iteração e o posicionamento geométrico de determinado valor do mapa são acessados a partir da lista retornada pela função *Patches*. Existe a possibilidade do usuário iterar diretamente nas folhas da octree, se este assim preferir². Sempre que necessário, pode-se chamar a função *Update*, que atualiza todas as células fantasmas para cada *patch*. Se o usuário deseja refinar a malha com base em algum critério, pode chamar a função *Refine* que, a partir de dois vetores como parâmetros de entrada, refina e salva em disco a nova malha, juntamente com os novos valores das variáveis para a nova malha. Estes dois vetores devem conter, respectivamente, as posições onde se deseja refinar, no formato (x,y,z) , e os níveis para o refinamento. Para visualização, é escrito um arquivo de dados paralelo, com base nos vetores passados à função *Save*.

4.5.2 Critérios de refinamento

O critério empregado no refinamento adaptativo em regiões específicas do escoamento possui grande influência no custo computacional e na exatidão da solução determinada. Não somente a densidade e a distribuição das células em diferentes níveis da malha são afetados pelo critério, mas também a localização de onde se deve ou não enriquecer ou diminuir o número

²Esta opção não é muito recomendada em termos de desempenho, uma vez que os mapas são construídos em função dos blocos, e não das folhas.

de nós na malha. No presente trabalho, foram analisadas duas maneiras para se calcular uma estimativa do erro ϵ_r , ambas baseadas nas componentes de velocidade, uma vez que o escoamento é incompressível. Dessa forma, o valor de ϵ_r é calculado com base nos valores da norma euclidiana dos gradientes nas três direções coordenadas, para cada uma das componentes de velocidade:

$$\|\nabla_u\| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2} \quad (4.76)$$

De maneira semelhante, outro critério empregado para calcular ϵ_r é baseado nos valores da norma euclidiana da vorticidade, como:

$$\|\omega\| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \quad (4.77)$$

onde,

$$\begin{aligned} \omega_x &= \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \\ \omega_y &= \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \\ \omega_z &= \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \end{aligned}$$

ϵ_r pode então ser normalizado entre 0 e 1, em função dos valores máximos e mínimos de $\|\nabla_u\|$ ou $\|\omega\|$, dependendo do critério adotado. Dessa forma, o nível de cada célula é mapeado proporcionalmente entre seu valor de ϵ_r e o número máximo de níveis na simulação. Por exemplo, se existem 3 níveis, e determinada célula no nível 1 possui $\epsilon_r = 0,4$, então, essa célula deve ser refinada para o nível 2. Em determinadas simulações, os máximos e mínimos dos valores calculados pelo critério não oferecem uma boa estimativa das regiões a serem refinadas. Nesse caso, utiliza-se um fator de escala, de maneira a aumentar ou diminuir o intervalo de aplicabilidade para determinado nível de refinamento.

5 Resultados

A forma de criação, distribuição entre os processos e manipulação da malha octree foi apresentada na Seção 2.3.2. De maneira geral, os resultados apresentados aqui são referentes a testes, verificações e validações qualitativas e quantitativas com relação a resultados experimentais para o problema da cavidade de seção quadrada com tampa deslizante a $Re = 1000$. Primeiramente, foi realizado um teste de modo a verificar a montagem da matriz global, nos casos sequencial e paralelo. Os resultados apresentados a seguir foram calculados usando a estrutura, com a presença de refinamento localizado. Para isso, a partir da Eq. (4.64) é esperado que as expressões, acoplando os fluxos entre os diferentes níveis da malha, estejam escritas de forma compacta, em um sistema linear da forma:

$$\mathbf{A}\Phi = \mathbf{b} \quad (5.1)$$

Para um dado valor de Φ , calcula-se \mathbf{b} e espera-se que a diferença:

$$\mathbf{A}\Phi - \mathbf{b} = \mathbf{r} \quad (5.2)$$

contenha valores próximos do erro de arredondamento de máquina para o tipo decimal de dupla precisão utilizado para representar as variáveis u , v , w e p , contidos em \mathbf{r} , que aqui representa o vetor de resíduo. Os testes foram feitos para diferentes números de processadores e de níveis de refinamento. Uma das configurações (com 4 níveis de malha) é mostrada na Fig. 5.1 comprovando, em parte, a implementação da Eq. (5.1).

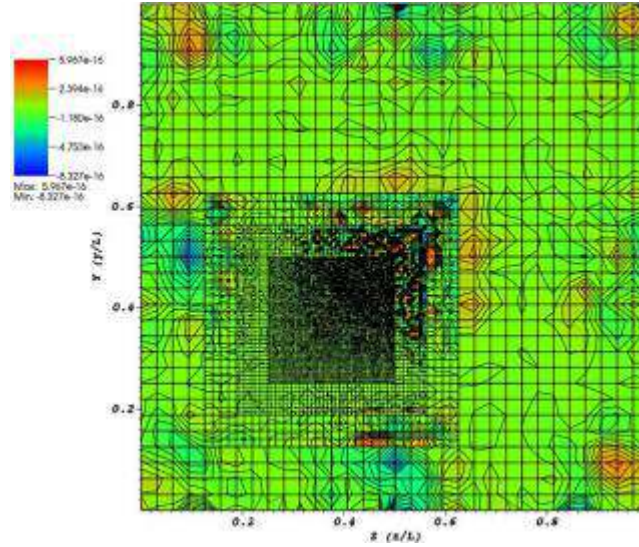


Figura 5.1: Corte bidimensional na posição $x = 0,75$, mostrando os resultados para a Eq. (5.1). Além do nível base, a malha possui mais 3 níveis de refinamento. Nota-se que os valores são da ordem do ϵ para os decimais de dupla precisão utilizados.

É conhecido que o reordenamento dos índices da matriz esparsa, além de gerar um sistema menor e economizar memória, pode melhorar o desempenho na solução direta de sistemas de equações lineares (GEORGE; LIU, 1981). Embora no presente trabalho o sistema linear esparsa seja resolvido utilizando métodos iterativos, a estrutura de dados utilizada facilita a criação de mais de um tipo de ordenação dos termos não nulos na matriz esparsa, o que permitiu fazer um teste utilizando dois modos de ordenação para criar a matriz. O primeiro é baseado na enumeração dos blocos aninhados, ou *patches*, enquanto que o segundo é mapeado a partir da curva de Morton, folha a folha. A Fig. 5.2 ilustra essas duas aproximações.

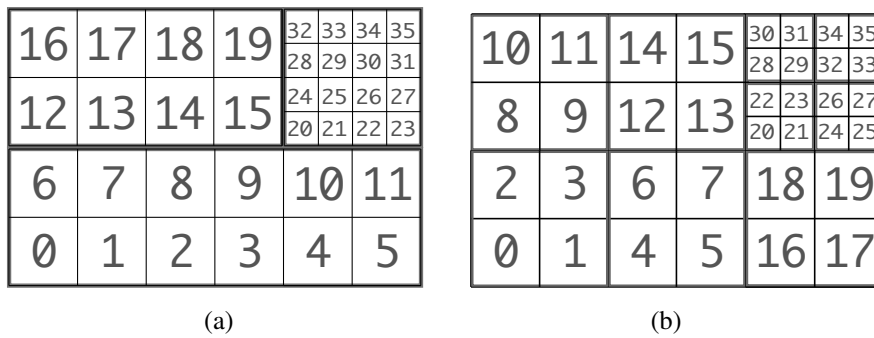


Figura 5.2: Duas maneiras de ordenação para os índices globais do mapa para o sistema de equações lineares: (a) Mapeamento por agrupamentos e; (b) por blocos 2×2 .

Tomando-se como exemplo uma malha uniforme octree, com nível base $16 \times 16 \times 16$

e 2 níveis adicionais de refinamento em um dos oitantes do domínio cúbico utilizado, foram plotados na Fig. 5.3 os gráficos de esparsidade da matriz resultante, respectivamente, para cada uma das ordenações apresentadas na Fig. 5.2.

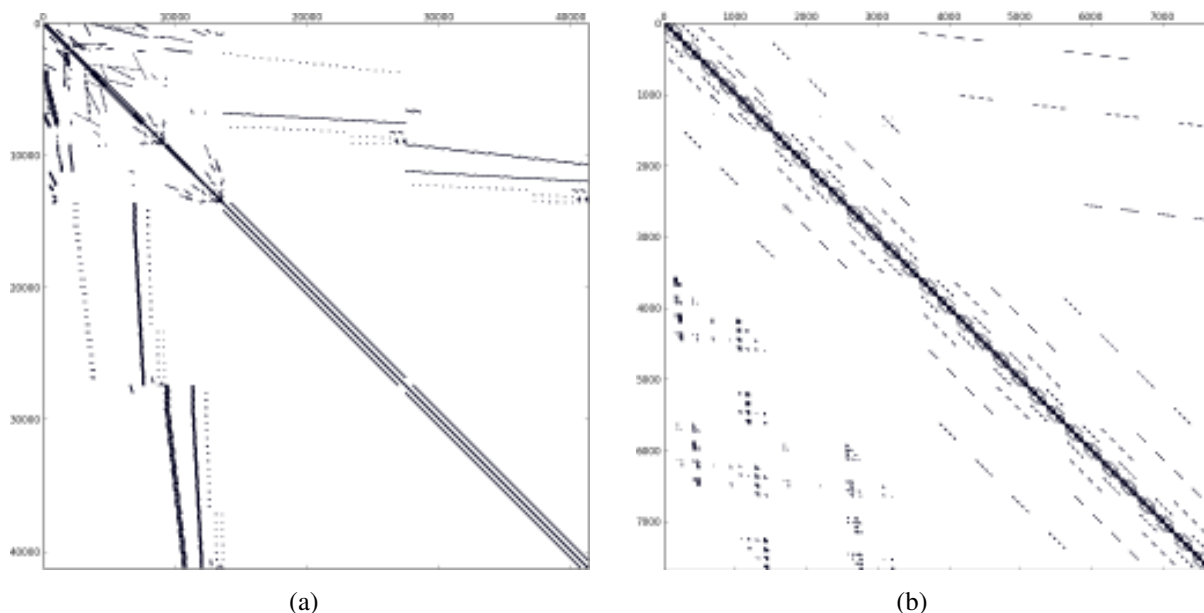


Figura 5.3: Diagrama apontando os termos não nulos na matriz: (a) utilizando ordenamento baseado nos agrupamentos de blocos e; (b) folha a folha, seguindo a curva de preenchimento de Morton.

Pode-se notar que as duas matrizes possuem simetrias características com relação à diagonal principal. Isso é devido ao fato que os termos de interpolação para uma célula fina em determinada interface fina-grossa são copiados como contribuições para a célula grossa que compartilha a mesma interface com a fina. No primeiro caso, com um único nível, a matriz se reduz àquela clássica advinda da discretização por diferenças finitas. Isso pode ser visto nas seções bem comportadas, representando os grandes blocos não refinados, o que não acontece no caso folha à folha. As matrizes são armazenadas no formato de linhas esparsas comprimidas ou da sigla em inglês (CRS). Para os métodos iterativos de solução utilizados no presente trabalho, não houve diferença no número de iterações, nem no resíduo ao final das iterações, ao utilizar uma ou outra forma de ordenação. Uma vez que o segundo método (folha à folha) é mais caro computacionalmente, adotou-se a primeira abordagem para obtenção dos resultados apresentados.

5.1 Testes de convergência

A partir da montagem e verificação da implementação do sistema de equações lineares, foi realizado um teste de convergência. Embora tenha sido utilizada uma equação ordinária para a equação elíptica, dada pela Eq. (5.3), o objetivo principal do teste consiste em verificar os erros do método aplicado nas interpolações e também aqueles relativos à implementação. O termo forçante na Eq. (4.19) é calculado a partir da equação manufaturada:

$$\phi(x, y, z) = \cos(2\pi x) \cdot \cos(2\pi y) \cdot \cos(2\pi z) \quad (5.3)$$

com condições de contorno de derivada imposta (*Neumann*), nas seis faces do domínio cúbico adotado. Na Tabela 5.1.1 são apresentados os erros determinados para as malhas com nível base, variando de 8 a 64 células em cada direção coordenada e com um nível de refinamento em um dos oitantes. Nela, para as quatro resoluções da malha octree, com um nível de refinamento, os valores das relações entre os erros (resíduos) calculados, são apresentados para os casos sequencial e paralelo. Para resolver o sistema, foi utilizado o método *BiCGStab* e a tolerância imposta para o valor máximo do resíduo foi calculada em função das dimensões das células nos diferentes níveis da malha. As diferenças entre os erros apresentados nas soluções sequencial e paralela podem ser devidos ao número diferente de iterações de convergência do método *BiCGStab*, quando executado em paralelo.

Na Fig. 5.4 é mostrada a solução para a equação de Poisson, referente à equação manufaturada, Eq. (5.3). Na Fig. 5.4a é mostrado o comportamento da solução nas faces externas do domínio cúbico utilizado. Nota-se que nas áreas mais refinadas (2 dos oitantes) os contornos são muito mais suaves que na região de malha mais grossa. Em uma outra configuração, mostrada na Fig. 5.4b, onde somente uma pequena porção do domínio foi refinada com dois níveis adicionais, é mostrada a distribuição do erro absoluto, proveniente da comparação entre a solução numérica e a Eq. (5.3). Naturalmente, o erro aumenta nas regiões com menor nível de resolução de malha.

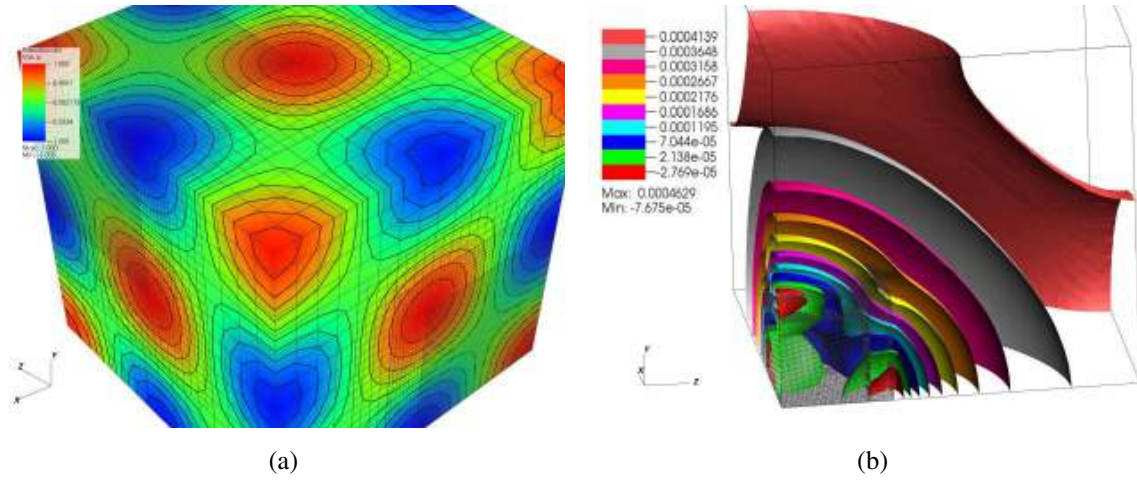


Figura 5.4: Resultados para a Eq. (5.3): (a) Resolução dos contornos para os diferentes níveis de malha e (b) Distribuição do erro absoluto.

Tabela 5.1.1: Erros calculados com base na norma $L2$ para a equação de Poisson, com termo fonte a partir da equação manufaturada Eq. (5.3).

Malha	Sequencial		Paralelo	
	$\ p_r - p\ _2$	r_p	$\ p_r - p\ _2$	r_p
8^3	7,0709e-02	—	6,9105e-02	—
16^3	1,7066e-02	4,2	1,7522e-02	3,9
32^3	4,1422e-03	4,2	4,4421e-03	3,9
64^3	1,0081e-03	4,1	1,0479e-03	4,2

Para testar as equações de Navier-Stokes, Eq. (4.1), foram adotadas as seguintes expressões para a variação transiente das componentes de velocidade e da pressão:

$$p(x, y, z, t) = \cos(t + 2\pi(x + y + z)) \quad (5.4)$$

$$u(x, y, z, t) = \sin(t + 2\pi(x + y + z))^2 \quad (5.5)$$

$$v(x, y, z, t) = \cos(t + 2\pi(x + y + z))^2 \quad (5.6)$$

$$w(x, y, z, t) = 1.0 \quad (5.7)$$

com as quais o requerimento de divergente nulo, pela Eq. (4.2), é satisfeito. A viscosidade cinemática e a densidade foram adotadas, respectivamente, como $\nu = 1$ e $\rho = 1$. No teste foram impostas condições de contorno de Dirichlet para as velocidades e também para a correção de

pressão ϕ . Para todas as 5 resoluções de níveis de malha, os cálculos foram feitos com 2 processadores¹ em uma mesma máquina. Diferente do método *RCB*, empregado na malha retilínea, o particionamento via curvas *SFC* não produz conjuntos bem comportados de malha para cada um dos processadores, o que pode influenciar no tempo de solução. Um exemplo ilustrativo do particionamento da malha utilizada nos testes com as equações Eq. (5.4) à Eq. (5.7), entre 11 processadores, é mostrado na Fig. 5.5a. Na Fig. 5.5b são traçados os valores para a solução do campo pressão, referente à Eq (5.4). Na Tab. 5.1.2 são apresentadas convergências dos erros, baseadas na norma $L2$ com $\nu = 1$ e $\rho = 1$ e comportamento dado pelo conjunto de equações (5.4 a 5.7). Para as cinco resoluções da malha octree, com um nível de refinamento, os valores são apresentados somente para o caso paralelo.

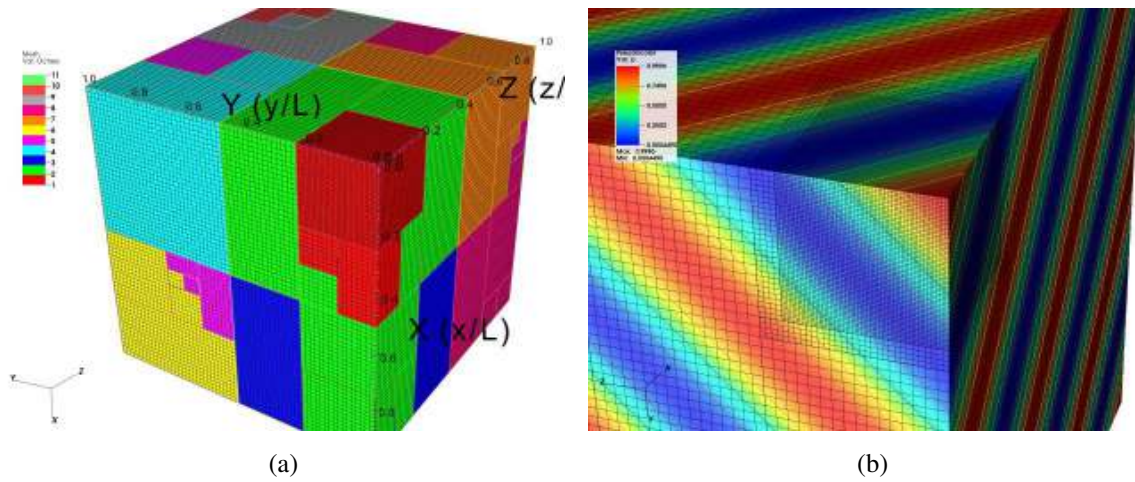


Figura 5.5: Testes na malha adaptativa: (a) Distribuição dos blocos em 11 processadores, para uma configuração genérica de refinamento. Detalhe da solução da pressão, referente à Eq (5.7), mostrada em uma das diferentes configurações testadas no trabalho.

Tabela 5.1.2: Convergência dos erros, baseados na norma $L2$, para as equações de Navier-Stokes.

Malha	$\ p_r - p\ _2$	r_p	$\ u_r - u\ _2$	r_u	$\ v_r - v\ _2$	r_v	$\ w_r - w\ _2$	r_w
4^3	9,747400e-02	—	6,309570e-06	—	1,052500e-05	—	4,701350e-06	—
8^3	4,061417e-02	2,4	1,971741e-06	3,2	1,046526e-06	3,1	2,127050e-06	4,5
16^3	1,310135e-02	3,1	6,161687e-07	3,2	3,270394e-07	3,2	8,467940e-07	4,0
32^3	4,679054e-03	2,8	1,579920e-07	3,9	8,838903e-08	3,7	3,328300e-07	3,9
64^3	1,613467e-03	2,9	4,514057e-08	3,5	2,455251e-08	3,6	1,211530e-07	3,6

¹Com núcleos de processamento reais, embora a máquina em questão fosse habilitada com *HyperThreading*.

É possível notar nos testes que a ordem de convergência se aproxima de 2 para as velocidades enquanto que a pressão se apresenta acima de primeira ordem, mas não chega a atingir segunda ordem. Para ambos os testes realizados, quando mais de um nível de refinamento é utilizado, ou a geometria da região de refinamento não é bem comportada, ou a topologia da malha varia muito entre os sucessivos refinamentos para o teste. Por isso, apenas a região de 1 oitante foi refinada com um nível de refinamento acima do nível base. Alguns testes, realizados com mais de um nível de refinamento, mostraram que não há diferença significativa ao adicionar mais de um nível no erro calculado.

A tolerância máxima imposta para a solução do sistema de equações lineares foi determinada como sendo o equivalente ao valor médio do que seria calculado para todos níveis de malha. Entretanto, isso ainda não está plenamente definido, podendo ser alterado por algum critério melhor definido na literatura para a solução de malhas cartesianas adaptativas do tipo caixa-preta. Os dois testes apresentados serviram, de certa forma, para uma verificação grosseira de seu funcionamento, frente à possibilidade de haver de erros de implementação e/ou metodologia empregados.

5.2 Cavity com tampa deslizante

Aqui utilizou-se o clássico problema da cavidade com tampa deslizante, como um dos casos teste, devido à simplicidade de implementação e ao grande número de dados disponíveis para comparação na literatura. O problema é apresentado na Fig. 5.21.

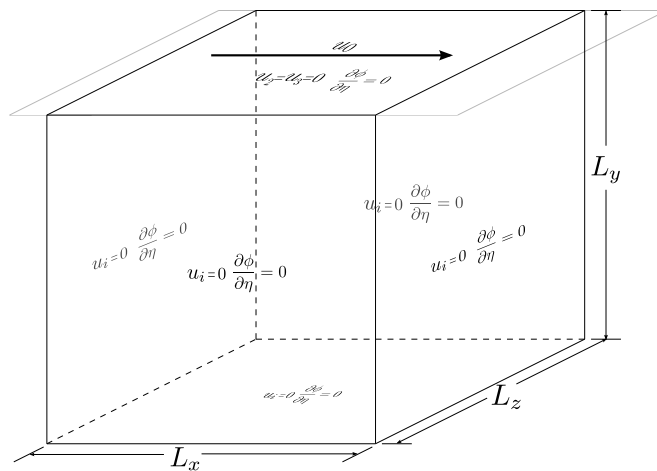


Figura 5.6: Diagrama esquemático para a cavidade quadrada com tampa deslizante.

Trata-se de uma cavidade de seção transversal quadrada, paralelepipedica, com condições de contorno de não-deslizamento em todas as paredes. A parede superior possui velocidade prescrita u_0 . As dimensões e condições de contorno impostas para a simulação da cavidade são mostradas na Fig. 5.21, na qual, exceto para a velocidade $u = u_0$ na tampa da cavidade, todas as componentes de velocidade no contorno são nulas. Em cada uma das faces da parede, é imposta a condição de derivada nula para a correção de pressão $\left(\frac{\partial \phi}{\partial \eta} = 0\right)$. O parâmetro adimensional que governa o escoamento é o número de Reynolds Re que, nesse caso, é dado por:

$$Re = \frac{u_0 L}{\nu} \quad (5.8)$$

na qual $L = L_x = L_y$ é o comprimento dos lados da cavidade e ν é a viscosidade cinemática do fluido. Entre as primeiras investigações do problema numérico, bidimensionalmente, estão (KAWAGUTI, 1961; BOZEMAN; DALTON, 1973). Ainda para o caso bidimensional, Ghia, Ghia e Shin (1982) simularam o problema variando o número de Reynolds de $Re = 10^2$ a $Re = 10^4$. Para isso foram utilizadas malhas com alta resolução com a formulação de segunda ordem do tipo *função corrente-vorticidade* (DAVIS; MALLINSON, 1976). Um extenso resumo sobre o assunto pode ser visto no trabalho de Shankar e Deshpande (2000). Experimentalmente, resultados detalhados do escoamento dentro de uma cavidade paralelepipedica são apresentados nos trabalhos de Migeon, Texier e Pineau (2000) e Migeon, Pineau e Texier (2003), para $Re = 10^3$.

Para as simulações realizadas aqui, a razão de aspecto para a cavidade será definida como $\Gamma = L/L_z$, onde $L = L_x = L_y = 1,0m$ é a dimensão de um dos lados da seção quadrada da cavidade.

Razão de aspecto $\Gamma_{lc} = 1$

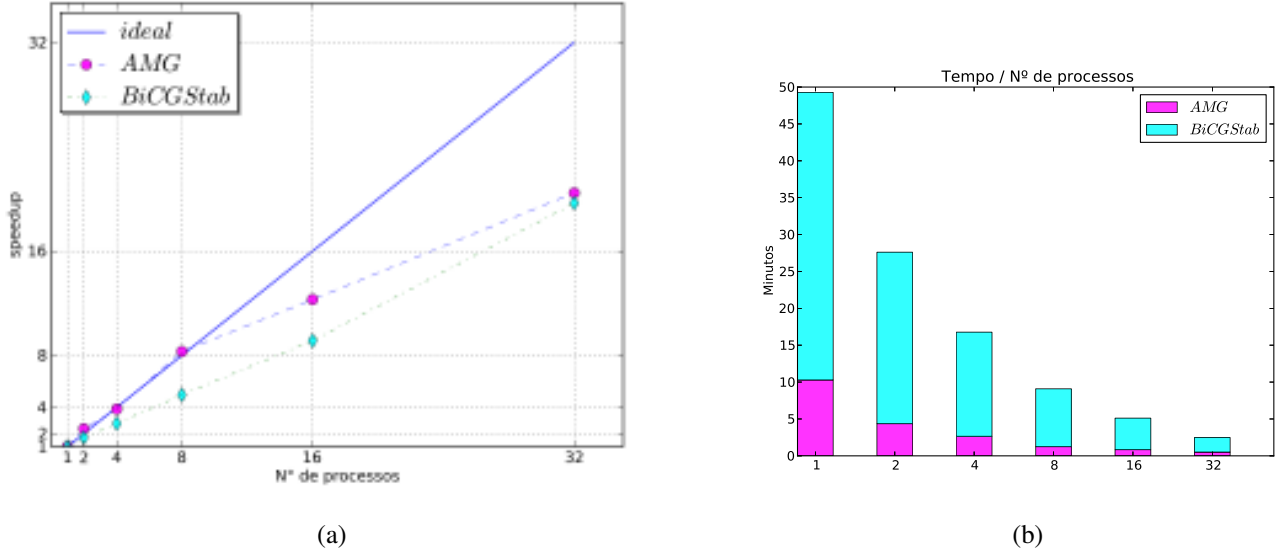


Figura 5.7: Problema da cavidade com tampa deslizante a $Re = 1000$ com soluções por *AMG* e *BiCGStab*: (a) Ganho em processamento (*speedup*) em função do número de processadores e curva ideal de ganho para até 32 processadores e; (b) Tempo gasto em 25 iterações de solução.

Foi realizado um teste de *speedup* para $Re = 10^3$ e $\Gamma = 1$, com $32 \times 32 \times 32$ volumes no nível base e 3 níveis de refinamento estático adicionais, aplicados próximos às paredes da cavidade. Os resultados são apresentados na Fig. 5.7a, enquanto que o gráfico de barras na Fig. 5.7b mostra os tempos relativos aos métodos *AMG* e *BiCGStab*, empregados na resolução do sistema linear para 25 iterações de solução. Na verdade os resolvidores são, respectivamente, *GMRES* preconditionado por *AMG* e *BiCGStab* preconditionado por relaxações do tipo *Jacobi*. O método *AMG* mostrou-se ao menos duas vezes mais rápido que o método *BiCGStab* na solução do sistema linear, como pode ser visto no gráfico da Fig. 5.7b. Pode-se notar que o desempenho apresentado é muito menor que o caso com a malha retilínea, mostrado na Fig. B.11, cujas máquinas possuíam o dobro de memória cache do que os computadores utilizados para o teste de *speedup* na malha adaptativa. Embora tenham sido utilizados apenas núcleos reais de processamento, sem multitarefas, o comportamento das curvas é muito diferente daquele que seria esperado de um teste de *speedup*. Foi descartada a influência das matrizes, uma vez que testes realizados com casos sem refinamento e apresentando matrizes iguais ao do caso com a malha retilínea (heptadiagonais), também apresentaram o mesmo comportamento no *speedup* da Fig. 5.7a. Talvez testes mais apurados possam ser realizados futuramente de forma a apontar possíveis causas desse comportamento. Todavia, os testes realizados no presente trabalho se mostraram satisfatórios.

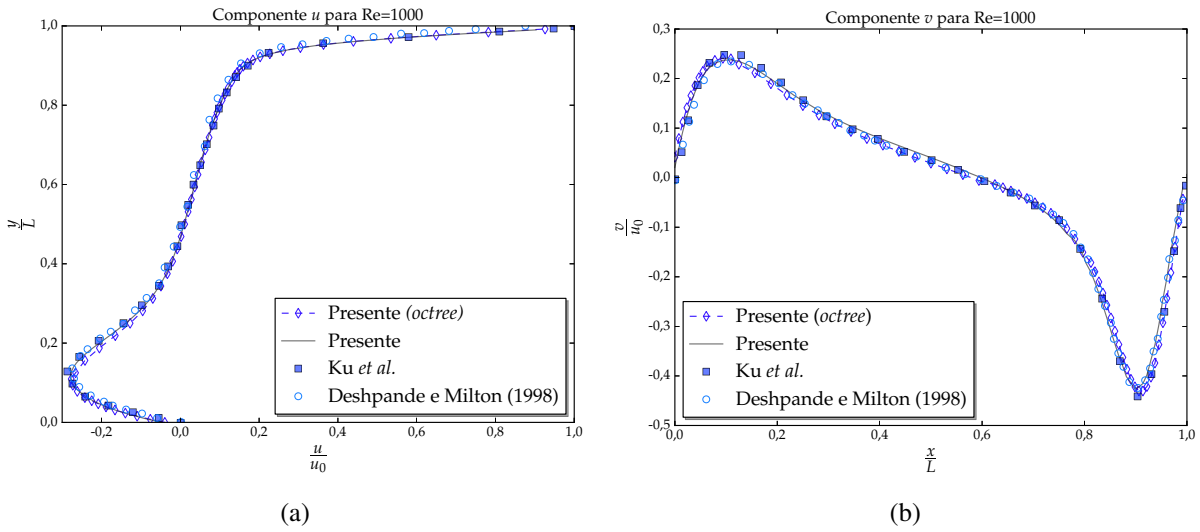


Figura 5.8: Comparação dos perfis das componentes de velocidade determinados no presente trabalho, com aqueles de Ku, Hirsh e Taylor (1987), Deshpande e Milton (1998) a $Re = 1000$: (a) Componente u sobre a linha vertical em y e passando pelo ponto $(x/L = 0,5; z/L = 0,5)$ e; (b) Componente v sobre a linha horizontal na direção x , cruzando o ponto $(y/L = 0,5; z/L = 0,5)$.

De modo a verificar a consistência dos resultados determinados com a malha adaptativa, é feita uma comparação com o código com malha retilínea para $Re = 1000$ e razão de aspecto $\Gamma = 1$. Essa comparação é apresentada no gráfico da Fig. 5.8, juntamente com os resultados de Ku, Hirsh e Taylor (1987) e de Deshpande e Milton (1998). Novamente, o perfil para velocidade u foi tomado sobre a linha vertical em $(x/L = 0,5; z/L = 0,5)$ e a distribuição para a componente de velocidade v foi retirada em uma linha horizontal passando pelo ponto $(y/L = 0,5; z/L = 0,5)$. A malha utilizada na simulação é apresentada na Fig. 5.9. Nota-se que a malha é relativamente grosseira, cujo nível base de dimensões $32 \times 32 \times 32$ é predominante, mesmo quando a simulação atinge o estado de regime permanente. A influência do refinamento é claramente visível nos gráficos da Fig. 5.8, cujos valores se aproximam dos dados de Ku, Hirsh e Taylor (1987) nas regiões onde a resolução de malha é maior. Entretanto, nas regiões, mesmo refinadas, próximas às interfaces entre os diferentes níveis de malha, essa aproximação não apresenta resultados melhores do que com malha uniforme.

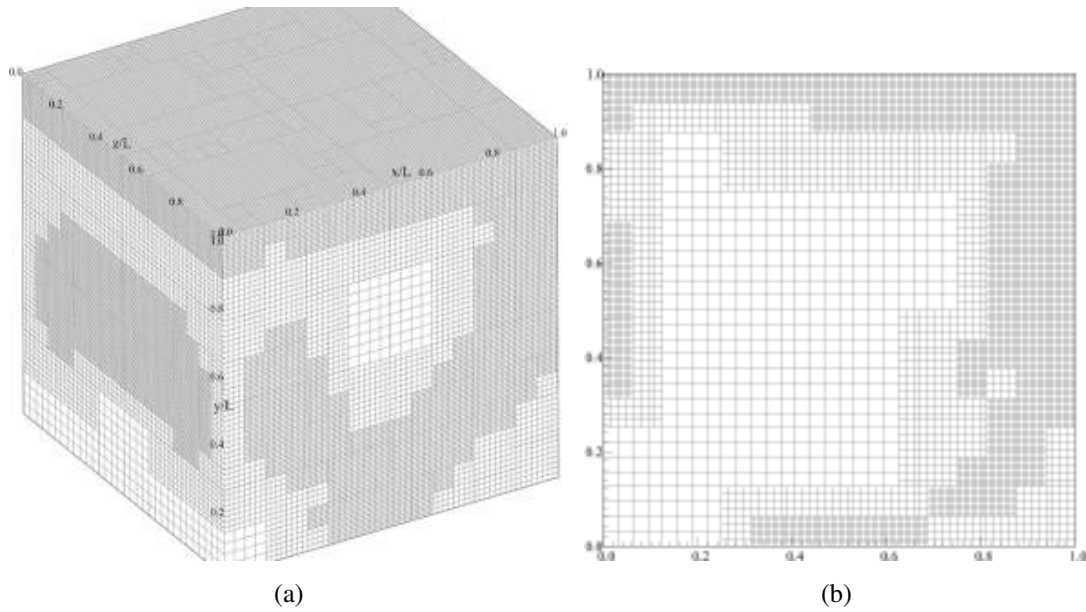


Figura 5.9: Malha computacional empregada na solução do problema de cavidade quadrada com tampa deslizante a $Re = 10^3$: (a) Visão externa mostrando detalhes das regiões refinadas e; (b) corte no plano $x - y$, em $(z/L = 0,5)$.

Razão de aspecto $\Gamma_{lde} = 2$

Aqui são apresentados os resultados para a simulação do escoamento em cavidade paralelepípedica com tampa deslizante, para $\Gamma = 2$ e $Re = 10^3$, assim como as comparações com valores experimentais apresentados nos trabalhos de Migeon, Texier e Pineau (2000) e Migeon, Pineau e Texier (2003). Entre os dados da referência estão visualizações para determinados passos de tempo e perfis das velocidades, obtidos a partir da técnica de PIV. Na Fig. 5.10, são mostradas as comparações para as componentes u e v , respectivamente, sobre as linhas de centro ($z/L = 1$) vertical e horizontal, para os instantes de tempo $t = 6s$, $t = 8s$ e $t = 12s$. Embora a tendência de todas as curvas estejam em concordância com os resultados experimentais, pode-se notar uma diferença significativa nos resultados da componente v , principalmente próximo dos máximos e mínimos das curvas. De maneira a mensurar a influência do refinamento de malha nos resultados, nas figuras Fig. 5.10a e Fig. 5.10b são comparados também os resultados bidimensionais, determinados com uma malha de dimensões 256×256 para o instante de $t = 2s$. Segundo Migeon, Texier e Pineau (2000), somente a partir de $t = 6s$, o escoamento no centro da cavidade, em $z = 1$, deixa de ser essencialmente bidimensional, o que permite que a comparação em questão seja realizada.

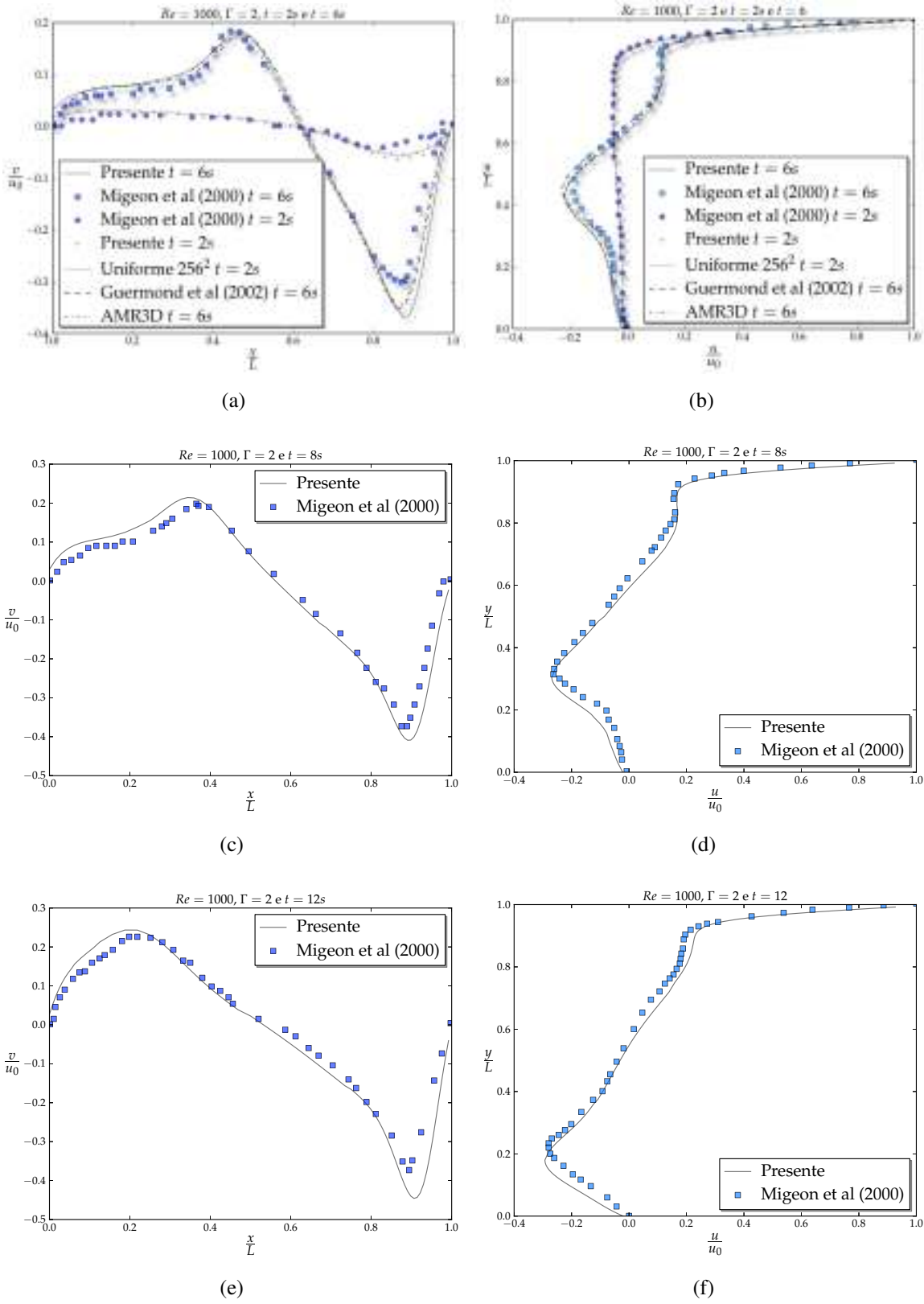


Figura 5.10: Perfis com a variação das componentes de velocidade u e v ao longo das linhas de centro vertical e horizontal, respectivamente: (a) e (b) $t = 6s$, (c) e (d) $t = 8s$ e (e) e (f) $t = 12s$.

Nota-se que, mesmo com o dobro da maior resolução utilizada na malha octree e um passo de tempo 16 vezes menor (modificado à partir do CFL), os resultados ainda não apresentam uma concordância satisfatória com o experimento para a variação de v na linha horizontal. No entanto, quando a comparação é feita com dados numéricos, os valores são muito próximos. Neste caso, os desvios podem ser devidos aos métodos utilizados e às aproximações empregadas. Na Fig. 5.10, as linhas tracejadas e com traço e ponto, são referentes aos resultados determinados numericamente nos trabalhos de Guermond et al. (2002) e Lima (2012), respectivamente. Guermond et al. (2002) utiliza o método de elementos finitos e resolve somente metade da cavidade, adotando que a solução final é perfeitamente simétrica com relação ao plano $z = 1,0$. Na tese de doutorado de Lima (2012), o método de diferenças finitas foi aplicado na solução de todo o domínio da cavidade, com esquema temporal do tipo semi-implícito, implementados no código *AMR3D*.

Detalhes da resolução da malha no plano onde foram retiradas as curvas para comparação, em cada um dos instantes de tempo, são apresentados na Fig. 5.12. Com base nas figuras, pode ser observado que as regiões nas quais foram retiradas as curvas das comparações não estavam completamente refinadas, com algumas ainda no nível base, no instante analisado. Isso se deve ao critério utilizado para o refinamento, apresentado na Seção 4.5.2. Na Fig. 5.11 são apresentados os contornos de pressão e uma visão mais detalhada da configuração final da malha adaptativa com os contornos do módulo da velocidade $|\mathbf{U}|$ para o escoamento completamente desenvolvido.

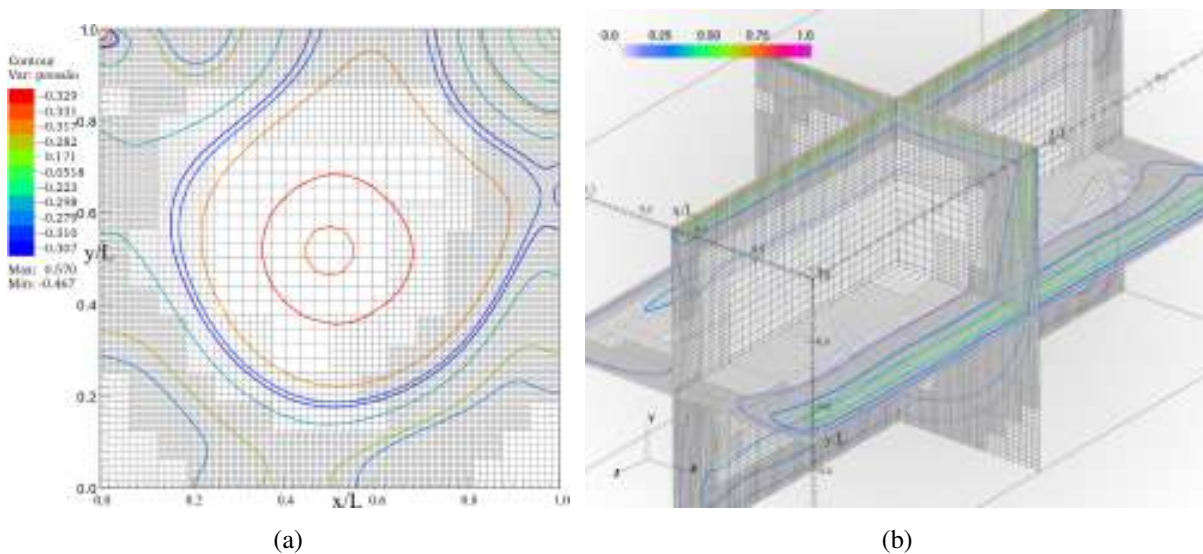


Figura 5.11: Resultados para cavidade já em regime permanente, no instante $t = 30s$, $\Gamma_{lbc} = 2$ e $Re = 10^3$: (a) Contornos de pressão em um corte no plano $x - y$, em ($z/L = 0,5$) e; (b) corte mostrando detalhes da malha juntamente com os contornos do módulo da velocidade $|\mathbf{U}|$, nos planos centrais nas três direções coordenadas.

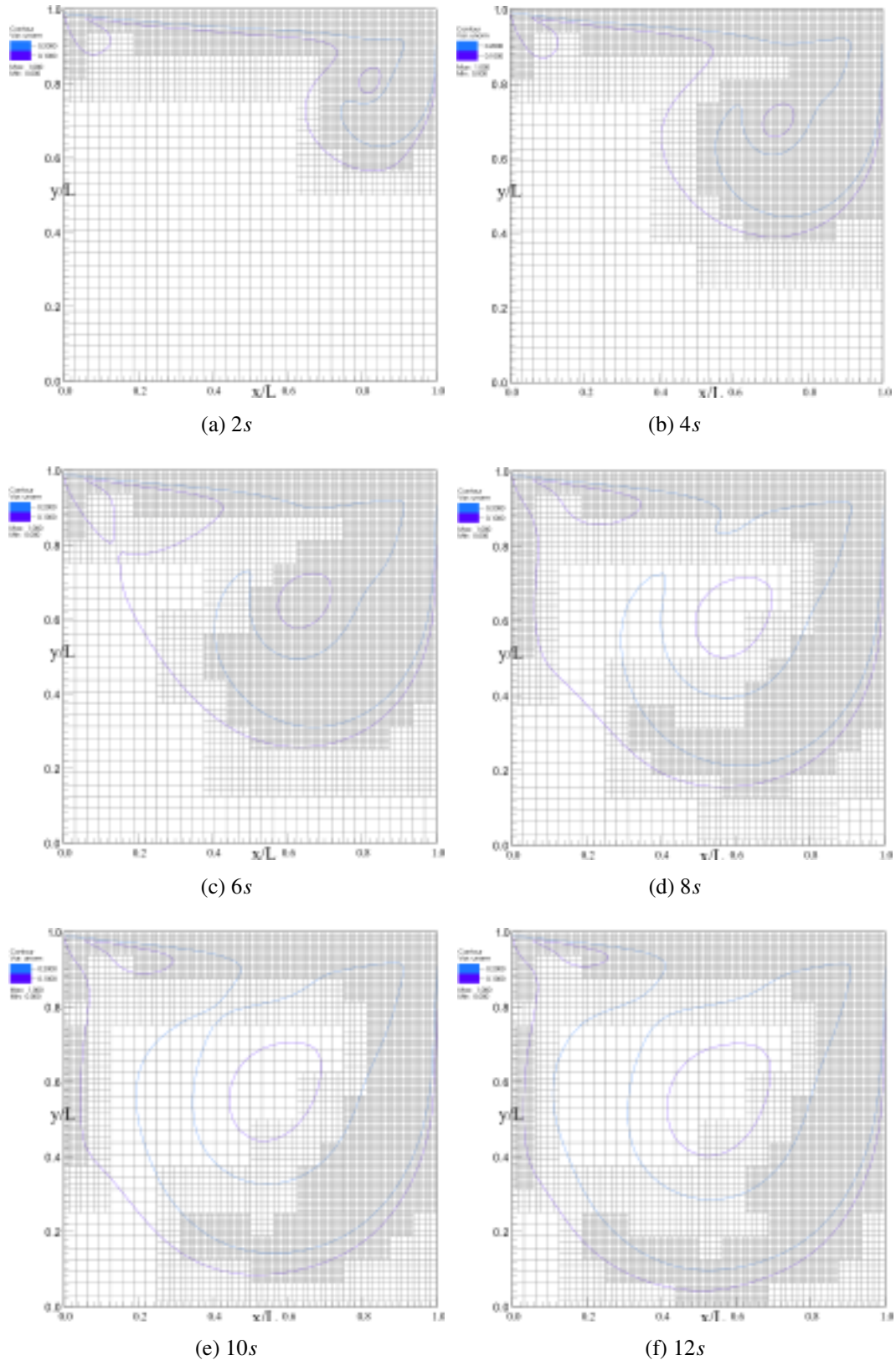
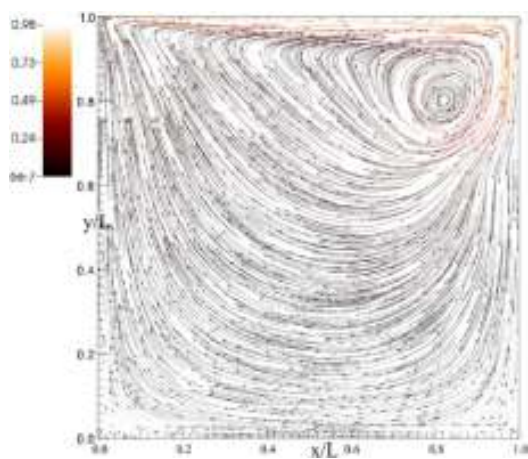
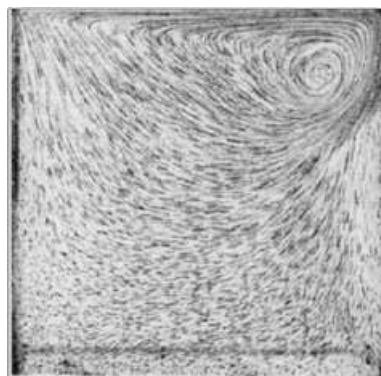
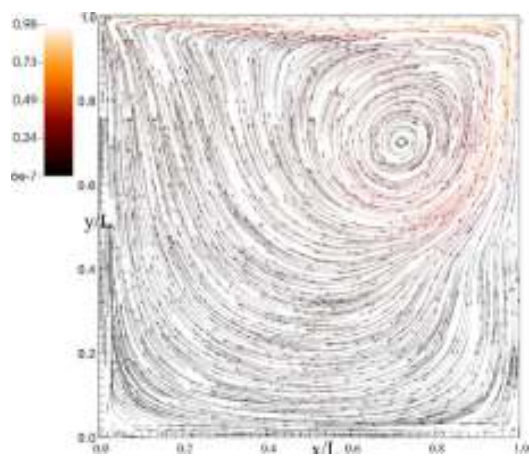
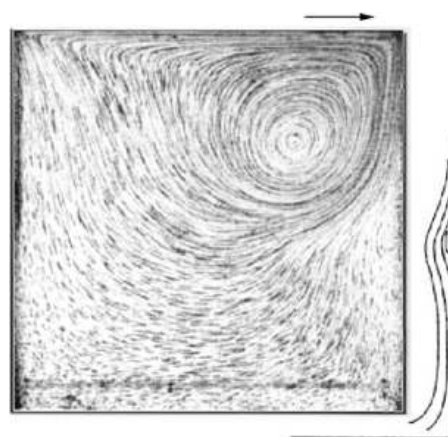
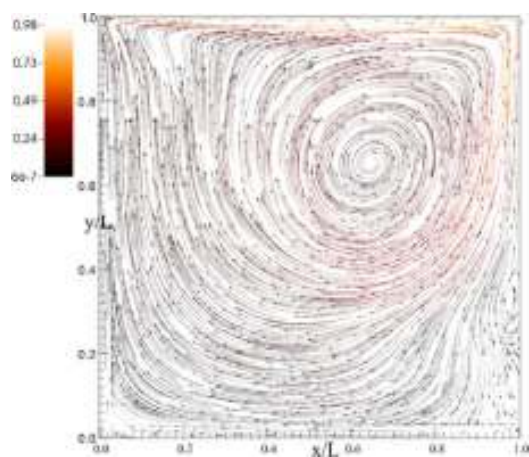
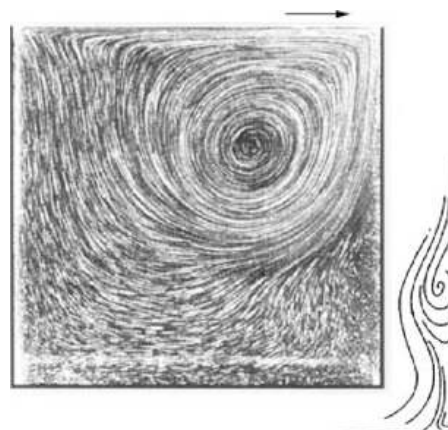
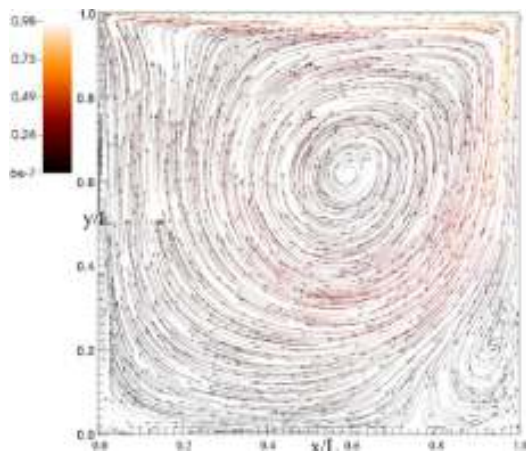
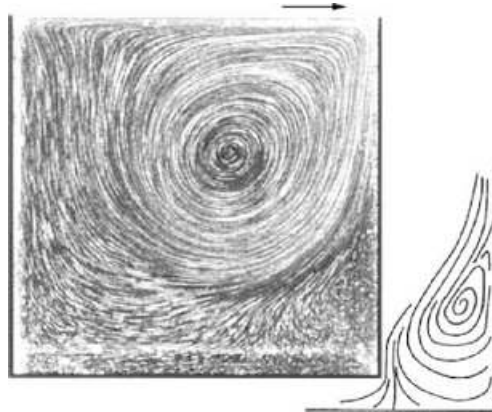


Figura 5.12: Configurações de malha após refinamento adaptativo para os instantes de $t = 2s$, $t = 4s$, $t = 6s$, $t = 8s$, $t = 10s$ e $t = 12s$ no plano $x-y$, em $z = 1$. Os contornos são relativos ao módulo da velocidade para os valores $|\mathbf{U}| = 0,1m/s$ e $|\mathbf{U}| = 0,2m/s$.

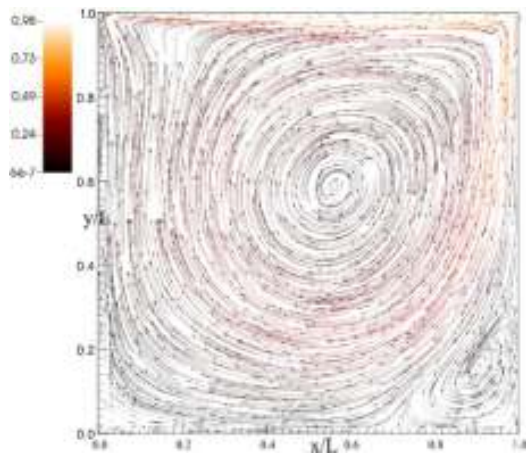
(a) $2s$ (b) $2s$ (c) $4s$ (d) $4s$ (e) $6s$ (f) $6s$



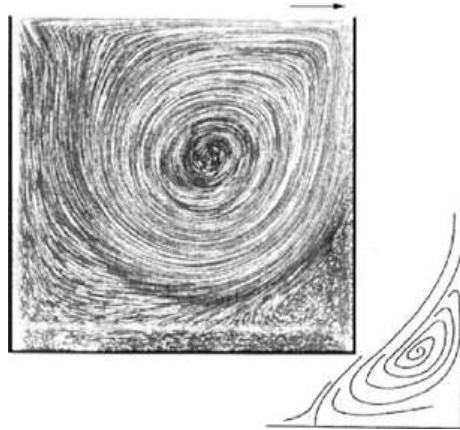
(g) 8s



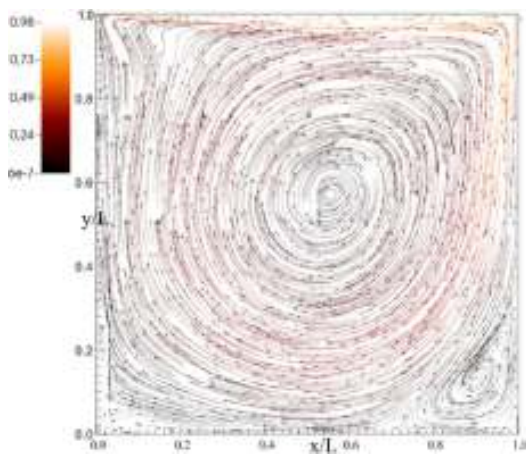
(h) 8s



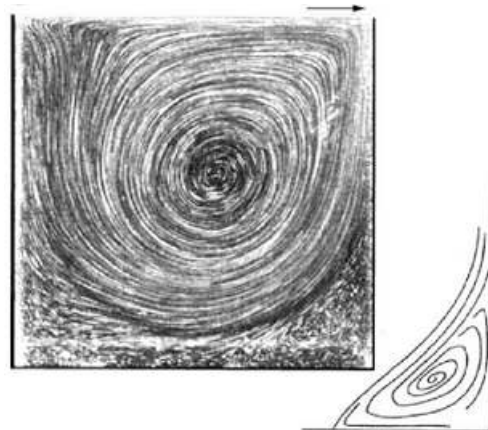
(i) 10s



(j) 10s



(k) 12s



(l) 12s

Figura 5.13: Cortes no plano $x-y$, em $z = 1$ para o problema da cavidade a $Re = 10^3$ e $\Gamma_{ldc} = 2$ para os instantes de tempo $t = 2s$, $t = 4s$, $t = 6s$, $t = 8s$, $t = 10s$ e $t = 12s$. Na coluna direita são mostrados os resultados experimentais de Migeon, Texier e Pineau (2000) e as linhas de corrente determinadas no presente trabalho são mostradas na coluna da esquerda.

A Fig. 5.13 mostra algumas comparações qualitativas do escoamento no plano mediano em $\frac{z}{L} = 1$. As visualizações são apresentadas aos pares, em uma sequência temporal, para os instantes de tempo $t = 2s$, $t = 4s$, $t = 6s$, $t = 8s$, $t = 10s$ e $t = 12s$. Inicialmente, o fluido da cavidade encontra-se em repouso, para a simulação e no experimento, até o momento em que a tampa é posta em movimento com velocidade constante u_0 . Na coluna direita, são mostrados os resultados experimentais de Migeon, Texier e Pineau (2000) e as linhas de corrente determinadas no presente trabalho são mostradas na coluna da esquerda. A configuração da malha em cada instante de tempo é mostrada na Fig. 5.12.

As comparações com os resultados experimentais apresentam muito boa concordância, evidenciada pela formação dos vórtices primário e secundário, exceto para o tempo de $t = 4s$, no qual a formação do vórtice secundário anterior não pode ser visualizada. Isso pode ser devido ao fato que o centro do vórtice secundário estaria justamente na interface entre dois níveis de refinamento, e as operações de restrição e prolongamento teriam amortecido sua formação. Até mesmo as interpolações lineares utilizadas para salvar o arquivo de dados ou as interpolações do aplicativo de visualização utilizado em todas as visualizações tridimensionais, (*VisIt*), podem ter influenciado sua visualização, uma vez que a velocidade na região em questão é muito baixa. Todavia, nos instantes seguintes, o vórtice secundário ganha mais intensidade à medida que o escoamento se desenvolve, se deslocando até o canto inferior da cavidade. O vórtice primário, cujo centro de rotação se inicia no canto superior direito da cavidade, se desloca na diagonal em direção ao centro da cavidade, mudando sua forma e intensidade. Para ilustrar esse deslocamento, as posições instantâneas do centro do vórtice primário são apresentadas no gráfico da Fig. 5.14. Para fins de comparação, na mesma figura podem ser observados os resultados determinados de maneira experimental e numérica por Migeon, Texier e Pineau (2000) e Guermond et al. (2002), respectivamente. Nota-se uma boa concordância entre os resultados.

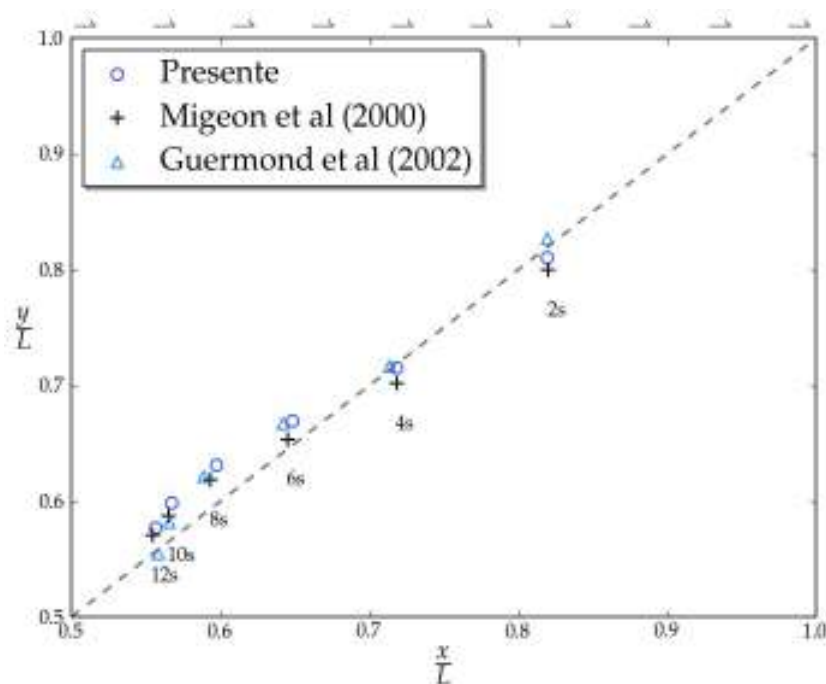


Figura 5.14: Posição do centro do vórtice primário para os instantes de tempo de $t = 2s$ à $t = 12s$.

Para o instante $t = 12s$, são mostrados na Fig. 5.15 os vórtices de canto e o início da formação de estruturas do tipo *Taylor-Gortler*. Segundo Migeon, Pineau e Texier (2003), a condição necessária para o surgimento dessas instabilidades é decorrente dos próprios vórtices de canto.

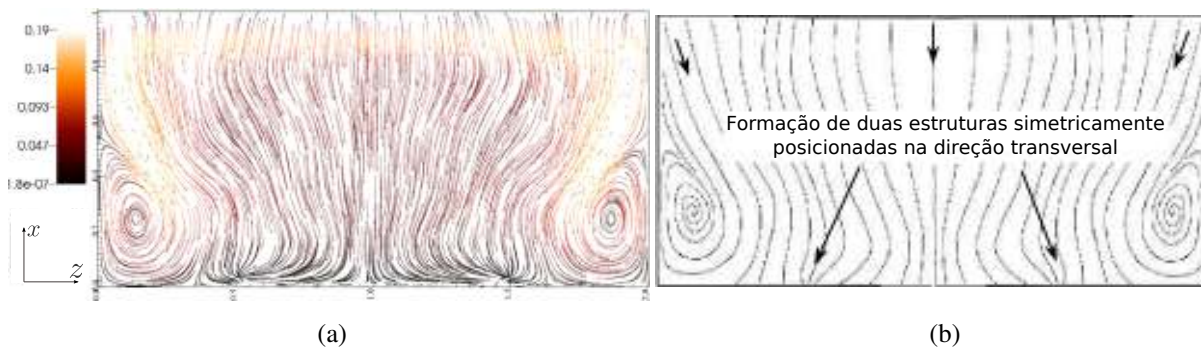


Figura 5.15: Corte no plano correspondente a $\frac{y}{L} = 0,5$ no escoamento em cavidade com $Re = 10^3$ e $\Gamma = 2$ evidenciando os vórtices de canto e a fase inicial de manifestações de estruturas do tipo *Taylor* no instante $t = 12s$: (a) Linhas de corrente determinadas no presente trabalho e; (b) Dedução do escoamento feita com base em dados experimentais de Migeon, Pineau e Texier (2003).

Outra comparação é apresentada na Fig. 5.16 para o instante $t = 8s$ no plano em $\frac{y}{L} = 0,75$. Nota-se que o corte não permite a visualização dos vórtices de canto, somente suas "pontas" nos cantos superiores direito e esquerdo da figura. O vórtice primário é apresentado

como sendo a região demarcada horizontalmente no centro da figura. Na região de escoamento a montante, mostrada na parte horizontal inferior da figura, pode-se ver que as linhas de corrente estão todas em ascensão. Esta também é considerada a parte mais instável do escoamento, segundo Migeon, Pineau e Texier (2003).

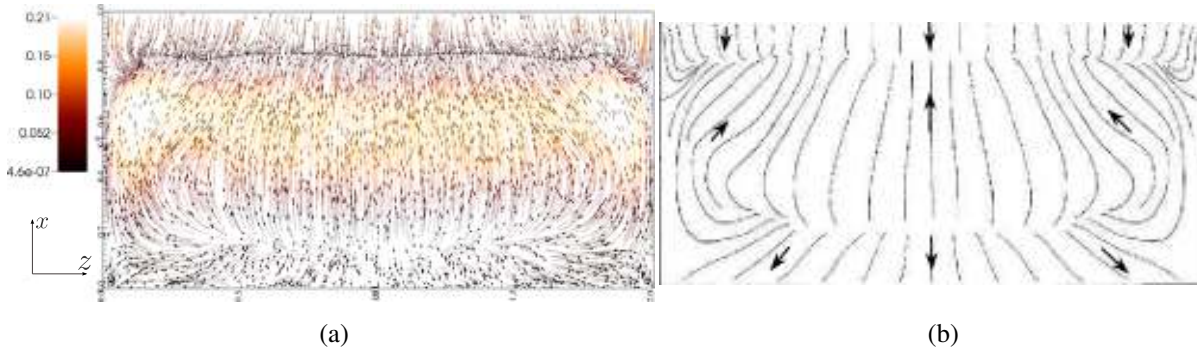


Figura 5.16: Corte no plano correspondente a $y/L = 0,75$ no escoamento em cavidade com $Re = 10^3$ e $\Gamma = 2$ no instante $t = 8s$: (a) Linhas de corrente determinadas no presente trabalho e; (b) Dedução da configuração do escoamento feita com base em dados experimentais de Migeon, Pineau e Texier (2003).

Além da simulação apresentada para $\Gamma_{ldc} = 2$, também foi simulado o mesmo escoamento, com malha refinada de maneira estática para o mesmo número de processadores. Neste caso, a simulação com refinamento estático, embora possuísse menos volumes nas últimas iterações da simulação, apresentou uma média de tempo gasto por iteração de 2,97 segundos e 1,6 GB de memória RAM, enquanto que para a malha adaptativa essa média foi de 0,98 segundos e 2 GB. Essa diferença se deve principalmente ao fato de que, no início da simulação o refinamento estático produz muitos volumes desnecessários. Mesmo com volumes a mais no final da simulação, a malha com refinamento dinâmico posiciona estes volumes de acordo com um critério predeterminado pelo usuário, o que acaba por melhorar a solução final em relação ao refinamento estático.

5.3 Jato incompressível laminar

Outra aplicação testada para a estrutura de dados desenvolvida nesse trabalho é a simulação do escoamento laminar de um jato incompressível, com número de Reynolds $Re_{r_0} = 65$, calculado com base no diâmetro do bocal. A configuração utilizada é similar àquela empregada nos trabalhos de Calegari (2012) e Lima (2012). Trata-se de jato gerado na saída de um bocal circular, com diâmetro d_0 e com velocidade normal constante v_0 . Assim como no caso de cavidade com tampa deslizante, foi utilizado um domínio computacional paralelepípedo. Nesse

domínio, são utilizadas condições de contorno de Dirichlet na entrada, derivada da componente normal de velocidade nula na saída ($\frac{\partial v}{\partial y} = 0$) e condição de simetria nas demais faces. As condições de contorno, assim como as dimensões do problema são esquematizadas na Fig. 5.17.

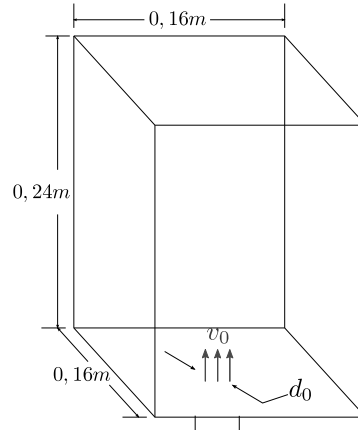


Figura 5.17: Diagrama esquemático para o problema de jato incompressível laminar.

As dimensões de $0,16m \times 0,24m \times 0,16m$ apresentadas na Fig. 5.17, são discretizadas em uma malha computacional de $16 \times 24 \times 16$ volumes, respectivamente nas direções coordenadas x , y e z . Pelo diâmetro do bocal $d_0 = 0,004m$, o fluido, com propriedades físicas constantes, $\rho = 1,184kg/m^3$ e $\nu = 1,5625 \cdot 10^{-5}m^2/s$, escoar com velocidade constante de $v_0 = 0,25m/s$. Na seção do bocal é imposto um refinamento manual, para garantir a maior resolução possível nessa região ($1/256$), garantindo 6 volumes no diâmetro do bocal. Todos os outros refinamentos ocorrem em determinados passos de tempo, e são aplicados através do mesmo critério utilizado para a cavidade com tampa deslizante, do valor máximo das derivadas das componentes de velocidade. As simulações foram realizadas com 16 processadores, gastando exatamente 7,75 horas de relógio para determinar resultados para 7 segundos de simulação.

Após 8,2 segundos de simulação, os resultados já não apresentavam alterações significantes no núcleo potencial ($< 10^{-5}$). Os resultados determinados no regime já permanente foram comparados com os resultados numéricos de Calegari (2012), Lima (2012) e analíticos de Boersma, Brethouwer e Nieuwstadt (1998). A solução analítica é válida para $\frac{y}{r_0} > 0,375Re_{r_0}$ (CALEGARI, 2012), ou seja, a partir de $\frac{y}{d_0} \geq 12$. Sua expressão é dada como:

$$\frac{v(y)}{v_0} = 0,375Re_{r_0} \frac{r_0}{y} \quad (5.9)$$

que é aplicada na linha vertical passando pelo ponto $(x = 0,08m; z = 0,08m)$.

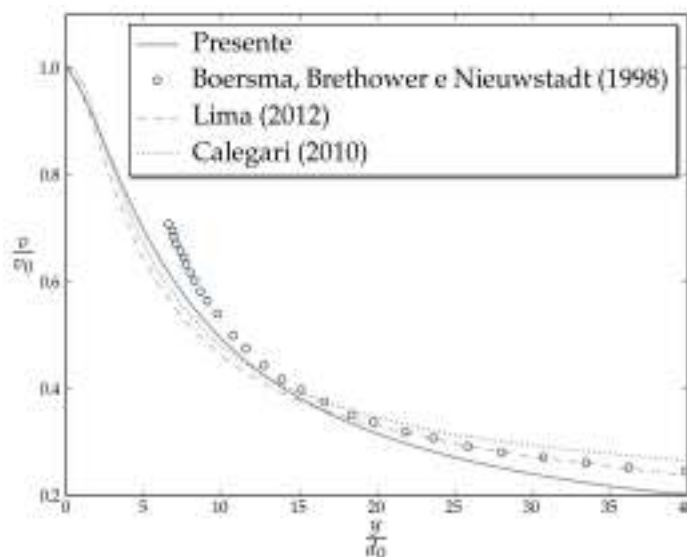


Figura 5.18: Curva de decaimento para a componente vertical de velocidades v na linha de centro na direção y .

A comparação é apresentada na Fig. 5.18, na qual pode-se notar uma boa concordância entre os valores de v para esse caso de escoamento completamente desenvolvido. Contudo, para ilustrar o desenvolvimento temporal do escoamento, assim como da malha computacional utilizada, são apresentados de maneira qualitativa, os valores da velocidade v em um corte, normal à direção z , exatamente no centro do domínio para os instantes de tempo de aproximadamente $t = 0,05s$, $t = 1,04s$, $t = 2,64s$ e $t = 7,5s$. Visualmente, é possível notar o ganho em redução de malha utilizada na simulação. Esse ganho já havia sido reportado por Calegari (2012) como sendo de 146 vezes de maior rapidez, no início da simulação, até 16 vezes, nos instantes finais, quando comparado com uma malha completamente refinada com o nível mais fino. Nesse caso, três ganhos podem ser contabilizados: o de refinamento adaptativo, o de refinamento dinâmico e o paralelismo, que juntos proporcionam um aumento no desempenho da solução.

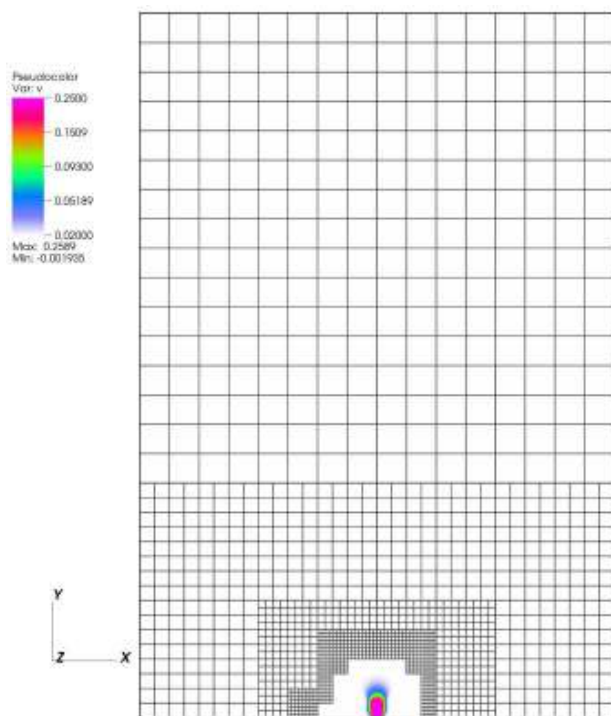
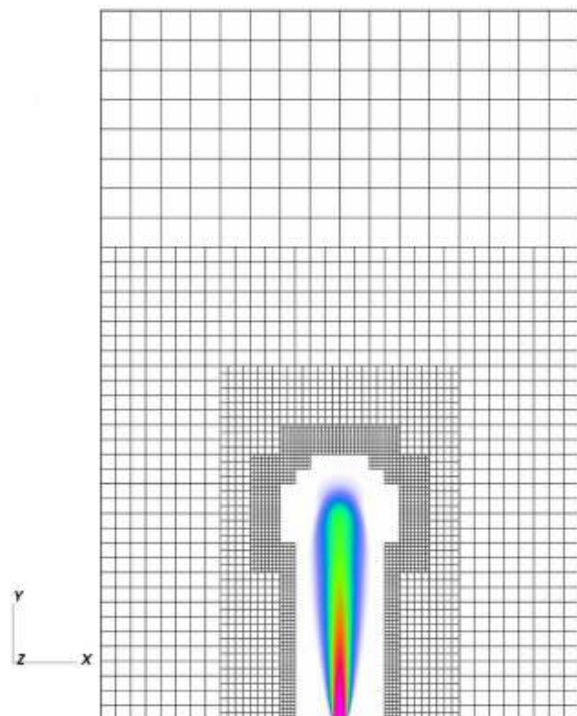
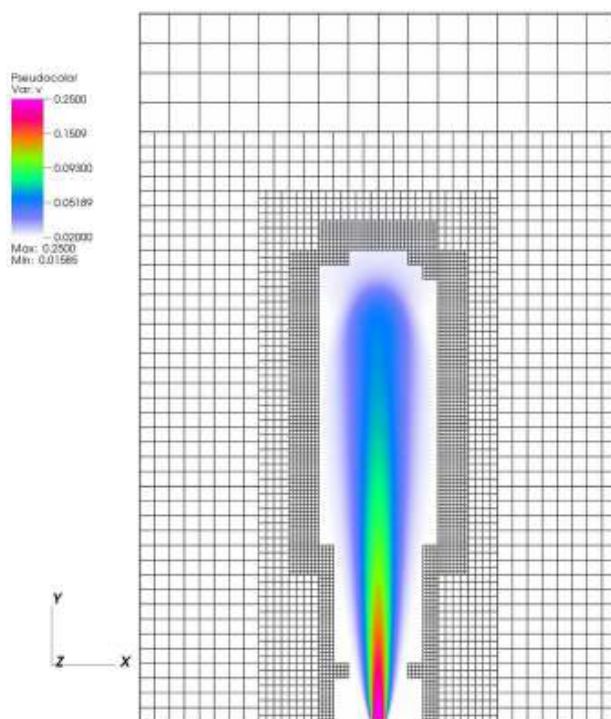
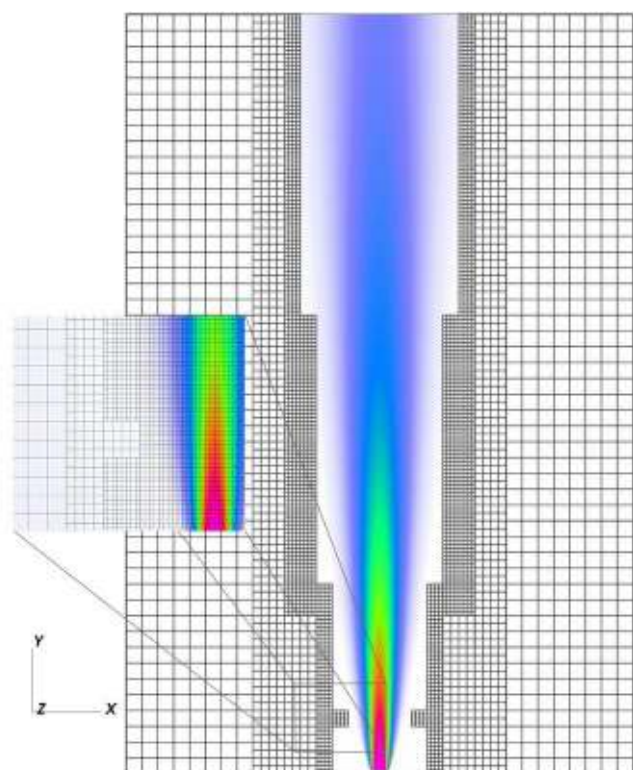
(a) $0,05s$ (b) $1,04s$ (c) $2,64s$ (d) $7,5s$

Figura 5.19: Configurações de malha após refinamento adaptativo para os instantes de $t = 0,05s$, $t = 1,04s$, $t = 2,64s$ e $t = 7,5s$ no plano $x - y$, em $z = 0,08$. Os valores são para a componente de velocidade v . O nível mais fino não é mostrado para uma maior clareza dos resultados.

Para ilustrar a distribuição da malha entre os 16 processadores ao longo da simulação, na Fig. 5.20 são apresentadas as malhas em 9, dos 16 processadores, para os instantes $t = 0,05s$ e $t = 8,01s$. Enquanto que no início da simulação os blocos são aglomerados próximo ao bocal, no regime permanente a distribuição da malha é quase simétrica na direção y .

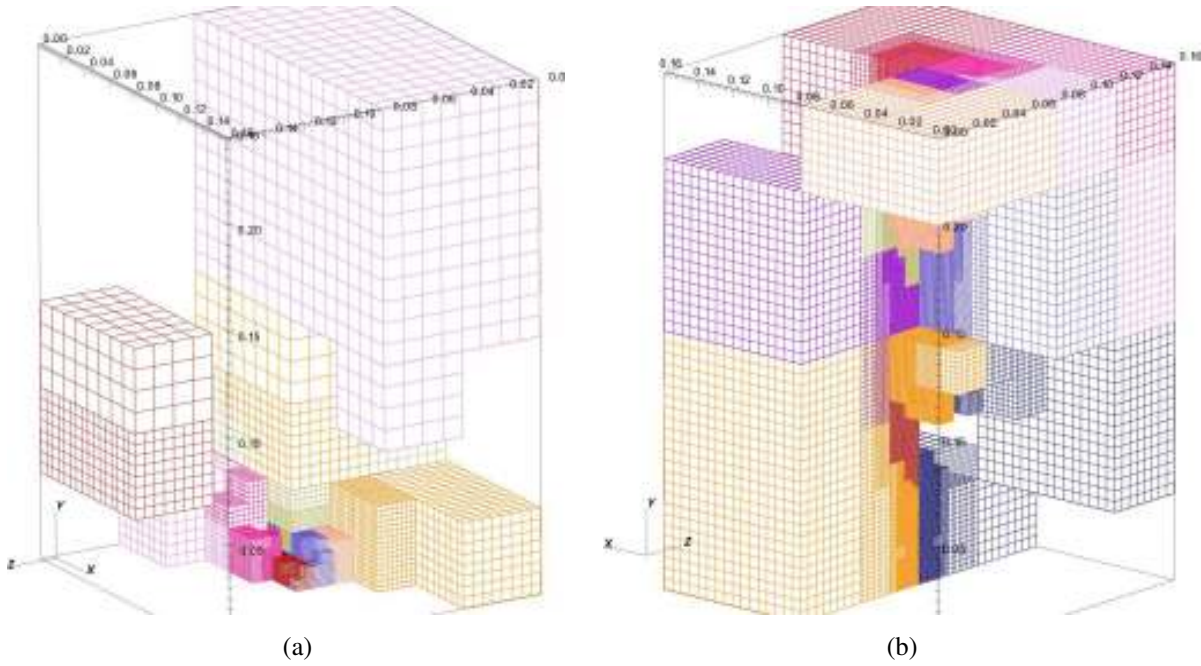


Figura 5.20: Configuração dos blocos de malha relativos aos processadores de número 8 à 16, nos instantes (a) $t = 0,05s$ e (b) $t = 8,01s$. As cores para cada bloco são as mesmas nas duas figuras.

5.4 Escoamento de Taylor-Couette

O escoamento entre cilindros rotativos concêntricos, conhecido como escoamento de Taylor-Couette, é amplamente utilizado, teórica e experimentalmente, em estudos sobre estabilidade e transição hidrodinâmicas. Uma das primeiras referências para o problema é dada por Taylor (1923) que, de forma teórica e experimental, constata o surgimento de instabilidades hidrodinâmicas, a partir da introdução de pequenas perturbações no escoamento. Foi determinado, ainda que, para pequenos valores da razão entre o espaçamento entre os cilindros $E = R_e - R_i$ e o raio interno R_i , o escoamento é simplificado e passa a depender basicamente do número de Taylor (Ta), definido de acordo com Hwang e Yang (2004) como:

$$Ta = \frac{\omega R_i E}{\nu} \quad (5.10)$$

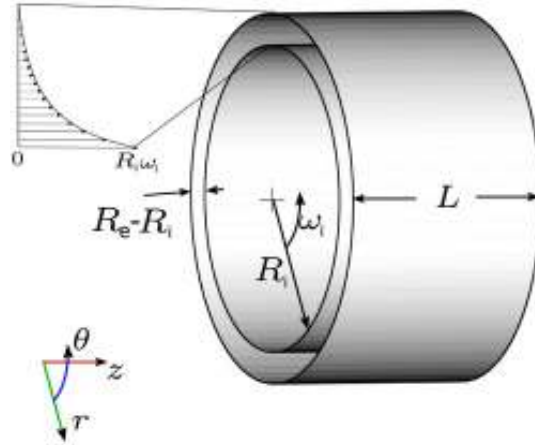


Figura 5.21: Representação esquemática do sistema de cilindros concêntricos para o escoamento de Taylor-Couette. O cilindro interno, de raio R_i possui velocidade angular ω_i .

Para pequenos valores do número de Taylor, observa-se o chamado escoamento circular de Couette, no qual existe somente a componente de velocidade azimutal, que pode ser escrita analiticamente segundo a equação (WHITE, 1974):

$$v(r) = Ar + \frac{B}{r} \quad (5.11)$$

na qual os termos do lado direito da Eq. (5.11) representam a combinação da rotação de corpo sólido e do vórtice potencial, e são dados respectivamente como:

$$A = \frac{(\omega_e R_e^2 - \omega_i R_i^2)}{R_e^2 - R_i^2}$$

$$B = \frac{(\omega_e - \omega_i) R_i^2 R_e^2}{R_e^2 - R_i^2}$$

Quando Ta ultrapassa um certo valor crítico Ta_{crit} , surgem as chamadas instabilidades de Taylor-Couette. Estas instabilidades se manifestam nos chamados vórtices de Taylor, estruturas toroidais contra-rotativas e axissimétricas, com comprimentos de onda próximos de $2E$. Nos testes, o espaço anular ou *gap* entre os cilindros E , assim como a razão entre raios $\eta = \frac{R_e}{R_i}$, possuem o mesmo valor em todos os casos estudados, com $E = 0,275m$ e $\eta = 3,2$. A razão de aspecto é definida como $\Gamma = L_z/E$, na qual L_z é o comprimento axial. As propriedades

físicas para o fluido newtoniano com massa específica $1 \frac{kg}{m^3}$ e viscosidade cinemática $0,01 \frac{m^2}{s}$, são mantidos constantes em todos os casos. O domínio de cálculo para a malha euleriana possui as dimensões $(1,0; 1,0; L_z)$, nas direções x, y e z respectivamente. Já os cilindros interno e externo apresentam raios de $R_i = 0,125m$ e $R_e = 0,4m$ respectivamente, e comprimento axial igual a $L_z = 0,4m$.

Na Seção B.2, são apresentados os resultados para o escoamento de Taylor-Couette com a malha retilínea. Entretanto, como pode ser observado na Fig. B.5, os resultados determinados não apresentaram uma boa concordância com a curva da solução proveniente pelo aplicativo em coordenadas cilíndricas, nas regiões próximas nos cilindros interno e externo. Como discutido por Borges (2009), isso ocorre principalmente devido à pequena resolução de malha nas proximidades da fronteira imersa. Aqui, utilizando a estrutura de malha adaptativa, desenvolvida no presente trabalho, o problema é simulado, com maior resolução próxima à malha lagrangiana e com condição de contorno de simetria na direção z , ao invés da condição de periodicidade empregada anteriormente. Utilizando-se a mesma razão de aspecto $\Gamma = 1,455$, os contornos da componente de velocidade $w = -0,1m/s$ e $w = 0,1m/s$, calculados com 4 processadores, são mostrados na Fig. 5.22a. Na mesma figura é mostrado o nível mais fino, que envolve completamente a fronteira imersa, como visualizado no corte da Fig. 5.22b, na qual a malha lagrangiana é representada pelas linhas tracejadas.

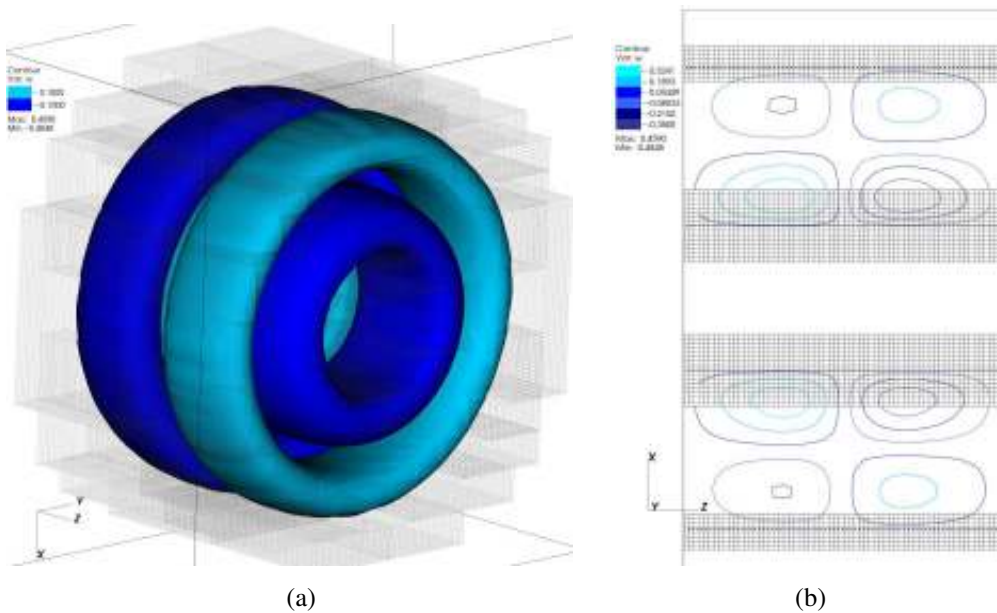


Figura 5.22: Escoamento de Taylor-Couette na malha adaptativa: (a) Contornos da componente $w = -0,1m/s$ e $w = 0,1m/s$, calculados utilizando 4 processadores e; (b) Corte mostrando detalhes das malhas lagrangiana (linhas horizontais tracejadas) e euleriana (somente o mais fino de 3 níveis).

A Fig. 5.23 apresenta detalhes da malha refinada em relação aos contornos, assim como a influência nestes (Fig. 5.23b);

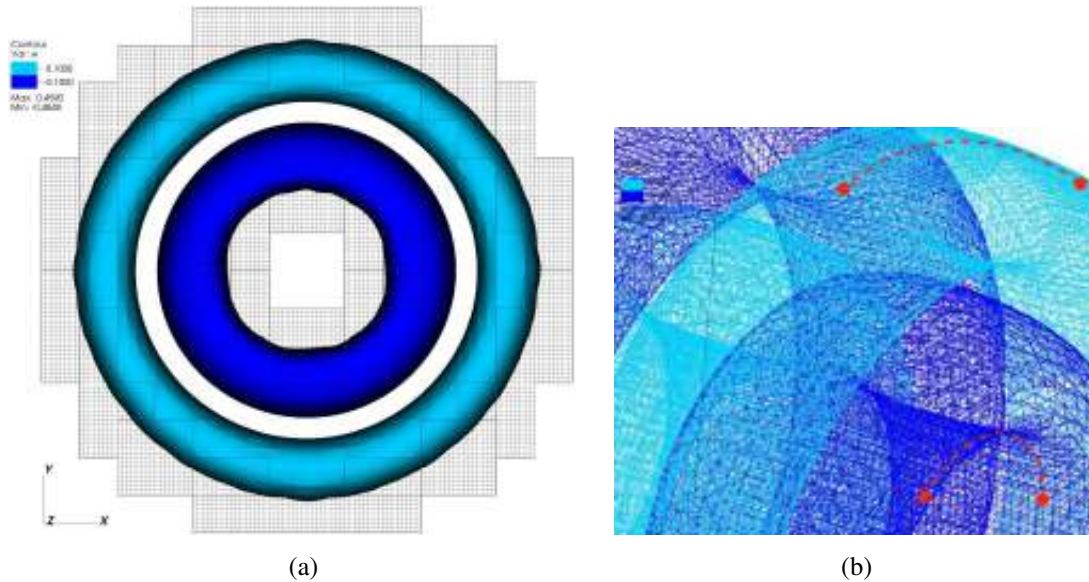


Figura 5.23: Detalhes e influência da malha adaptativa no cálculo dos contornos da componente de velocidade $w = \pm 0,1$.

Novamente, são comparados os resultados do presente trabalho com os obtidos com o código de Padilla e Silveira Neto (2008) em coordenadas cilíndricas e utilizando os métodos de fronteira imersa de Modelo Físico Virtual (MFV) e de Multiforçamento Direta (MDF). A melhora dos resultados pode ser visto claramente na Fig. 5.24, principalmente nas regiões onde foram refinadas. De modo semelhante, a variação da componente de velocidade radial adimensional $u_r/(R_i\omega)$ na direção axial, mostrada na Fig. 5.25, também foi levemente melhorada. Os dados são retirados na linha axial que passa por $R_i + E/2$, ponto $(x = 0,5; y = 0,76)$, nos limites compreendidos entre um par de vórtices de Taylor. Embora em fase inicial, estes resultados comprovam que o refinamento próximo às malhas lagrangianas contribuiu para uma melhor representação da fronteira imersa, tendendo para o caso no qual a malha foi ajustada à superfície (coordenadas cilíndricas).

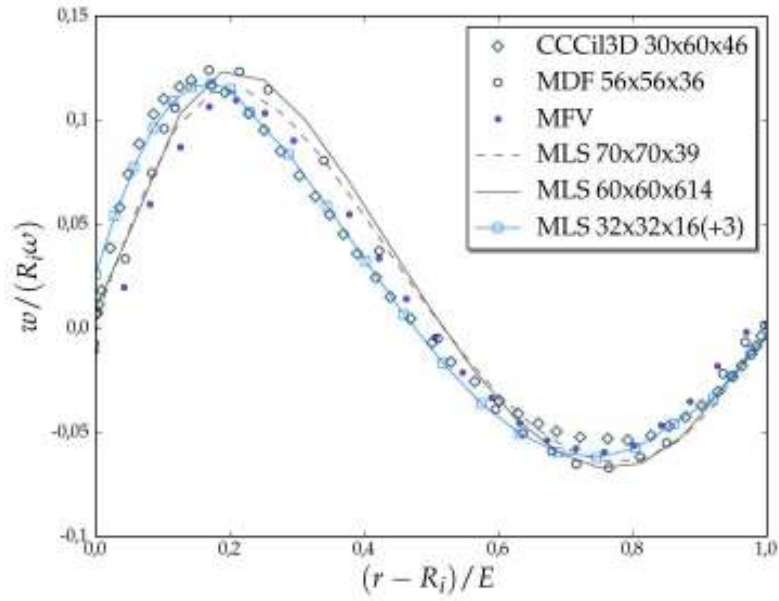


Figura 5.24: Variação de velocidade axial adimensional $w/(R_i\omega)$ na direção radial para $Ta = 100$. Os valores são referentes à linha passando pelo núcleo de um dos vórtices.

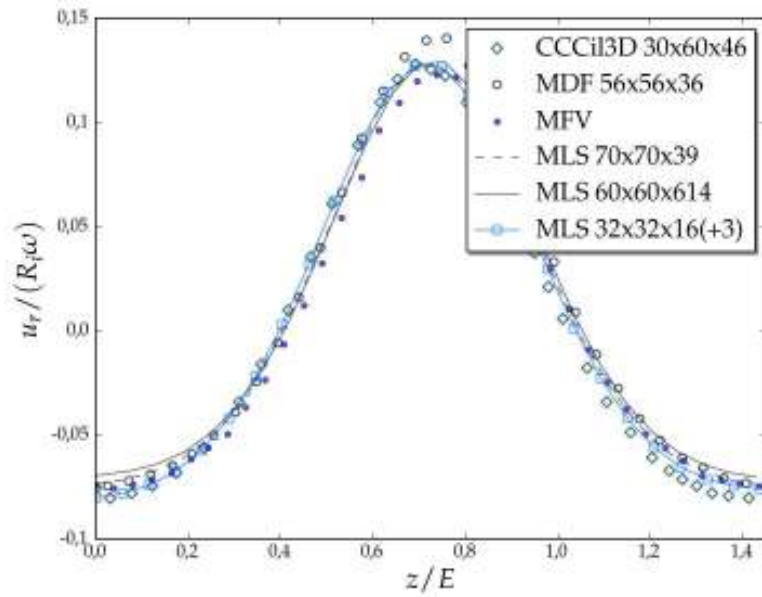


Figura 5.25: Componente de velocidade radial adimensional $u_r/(R_i\omega)$ sobre a linha axial passando pelo centro dos vórtices para $Ta = 100$. Os valores são tomados no intervalo entre dois vórtices contrarrotativos.

6 Conclusões e Perspectivas

Foram apresentadas duas implementações computacionais para resolver as equações de Navier-Stokes aplicadas a escoamentos incompressíveis, tridimensionais, em regime laminar e com coeficientes constantes, cuja solução do sistema de equações lineares é provida por uma biblioteca computacional externa. Essa flexibilidade permite ao usuário escolher entre as muitas possibilidades disponíveis de *solvers*, tanto de código fonte aberto, como códigos comerciais, aquela que melhor lhe servir em determinado problema. A malha computacional é composta por aninhamento de blocos de células que, por sua vez, são gerenciados por uma estrutura do tipo *octree*. O ganho computacional apresentado, devido ao número menor de células exigidas para representar um determinado problema, mostra claramente que a ferramenta desenvolvida será de grande contribuição para a simulação de problemas físicos, exibindo interações do tipo fluido-estrutura, que constituem a motivação inicial do trabalho.

Implementou-se a paralelização em um código numérico já existente no Laboratório de Mecânica dos Fluidos *MFLab* (Seção 4.5), discretizado temporalmente pelo esquema de segunda ordem de Adams-Bashforth, espacialmente por diferenças-centradas, também de segunda ordem de exatidão e com método de acoplamento pressão-velocidade por passos-fracionados. O conjunto de bibliotecas para computação científica *Trilinos* foi utilizado no particionamento do problema com balanceamento de carga e também na solução do sistema linear, decorrente da correção de pressão no método de passos fracionados. Os resultados são apresentados na Seção B. Para o problema de cavidade com tampa deslizante, nota-se um alto ganho com o aumento no número de processadores, como mostrado pelo gráfico de *speedup*. As comparações realizadas com resultados experimentais e numéricos para os problemas de cavidade com tampa deslizante e também para o problema de Taylor-Couette, com fronteira imersa, validaram o paralelismo implementado.

Esse primeiro exercício serviu como base para implementar uma nova estrutura de dados, também paralela, com blocos de malha cartesiana, uniforme e indexada geometricamente em uma árvore octonária (*octree*), apresentada na Seção 2.3. Embora a nova estrutura tenha

representado uma grande economia, tanto em memória como em processamento, o aumento de poder de cálculo com o número de processadores ficou abaixo do esperado. As comparações com dados experimentais (Seção 5.2), para uma cavidade quadrada com tampa deslizante e razão de aspecto 2, apresentaram boa concordância, tanto qualitativa como quantitativamente. Pode ser comprovado que, mesmo para o problema da cavidade com tampa deslizante, no qual, nos instantes próximos do regime permanente, quase 70% do domínio é refinado, tanto a memória como o poder computacional são significativamente menos exigidos, quando comparados com o mesmo cálculo em uma malha uniforme e com a resolução do nível mais fino.

Outros testes ainda são necessários para validar satisfatoriamente o aplicativo. Entretanto, o presente cenário, com malha adaptativa, paralelismo e um conjunto de *solvers* para solução de sistemas lineares à disposição, é muito mais propício para lidar com os desafios inicialmente introduzidos na proposta para o presente trabalho. Entre os trabalhos futuros estão a implementação de esquema temporal implícito ou semi-implícito, que possua menos restrições com o passo de tempo, fortemente limitado no presente trabalho pelo nível mais fino da malha. Realizar uma implementação mais robusta da metodologia de fronteira imersa, mais próxima do que foi apresentada na Seção 4.2.3. Uma vez que a presente implementação está limitada a pequenos valores de Re , também é pretendida a implementação de modelos de turbulência. Outro trabalho futuro talvez seria a implementação de modelos de fluidos não-newtonianos na malha adaptativa já que, atualmente, muitos são os sistemas, como na perfuração de poços de petróleo, cuja simulação envolve este tipo de fluido.

Referências

- AFTOSMIS, M. J.; BERGER, M. J.; ADOMAVICIUS, G. A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries. *AIAA Paper 2000-0808*, 2000.
- APPLIED, H. S.; SIMON, H. D. *Partitioning of Unstructured Problems for Parallel Processing*. 1991.
- ARRUDA, J. M. de. *Modelagem matemática de escoamentos internos forçados utilizando o Método da Fronteira Imersa e o Modelo Físico Virtual*. 153 f. Tese (Doutorado) — Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2004.
- BABU, V.; KORPELA, S. A. Numerical solution of the incompressible three-dimensional navier-stokes equations. *Computers & Fluids*, v. 23, n. 5, p. 675 – 691, 1994. ISSN 0045-7930. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0045793094900094>>.
- BASHFORTH, F.; ADAMS, J. *An attempt to test the theories of capillary action: by comparing the theoretical and measured forms of drops of fluid. With an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops*. [S.l.]: University Press, 1883.
- BERGER, M. et al. Performance of a new cfd flow solver using a hybrid programming paradigm. *Journal of Parallel and Distributed Computing*, v. 65, n. 4, p. 414 – 423, 2005. ISSN 0743-7315. Disponível em: <<http://www.sciencedirect.com/science/article/B6WKJ-4FG2X3J-1/2/a3f0569fc9003ff9bee33d224450eb1b>>.
- BERGER, M. J.; OLIGER, J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, v. 53, p. 482–512, 1984.
- Boersma, B. J.; Brethouwer, G.; Nieuwstadt, F. T. M. A numerical investigation on the effect of the inflow conditions on the self-similar region of a round jet. *Physics of Fluids*, v. 10, p. 899–909, abr. 1998.
- BORGES, J. E. *Modelagem Matemática e Simulação de Escoamentos em Canais Anulares com Interação Fluido Estrutura*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2009.
- BOTTA, E. F. F.; WUBS, F. W. The convergence behaviour of iterative methods on severely stretched grids. *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, Ltd, v. 36, n. 19, p. 3333–3350, 1993. ISSN 1097-0207.
- BOZEMAN, J. D.; DALTON, C. Numerical study of viscous flow in a cavity. *Journal of Computational Physics*, v. 12, n. 3, p. 348 – 363, 1973. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999173901575>>.
- BREZINSKI, C. *Numerical methods and algorithms*. [S.l.]: Springer, 2005.

CALEGARI, P. C. *Simulação computacional de escoamentos reativos com baixo número de Mach aplicando técnicas de refinamento adaptativo de malhas*. 90 f. Tese (Doutorado) — Universidade de São Paulo, São Paulo, SP, Brasil, 2012.

CAMPBELL, P. M. et al. *Dynamic octree load balancing using space-filling curves*. [S.l.], 2003.

CAMPREGHER, R. *Modelagem matemática tridimensional para problemas de interação fluido-estrutura*. 179 f. Tese (Doutorado) — Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2005.

CATALYUREK, U. et al. Hypergraph-based dynamic load balancing for adaptive scientific computations. In: *Proc. of 21st International Parallel and Distributed Processing Symposium (IPDPS'07)*. [S.l.]: IEEE, 2007. Best Algorithms Paper Award.

CENICEROS, H. D. et al. A robust, fully adaptive hybrid level-set/front-tracking method for two-phase flows with an accurate surface tension computation. *Commun. Comput. Phys.*, v. 8, p. 51 – 94, 2010.

CHARLTON, E.; POWELL, K. An octree solution to conservation laws over arbitrary regions (oscar). *AIAA Paper 97-0198*, Jan. 1997.

COIRIER, W. J. *An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations*. 252 f. Tese (Doutorado) — NASA Lewis Research Center, Cleveland, OH, 1994.

CONNELL, S. D.; HOLMES, D. G. A 3d unstructured adaptive multigrid scheme. *AIAA Paper 93-3339*, 1993.

DAVEY, A. The growth of Taylor vortices in flow between rotating cylinders. *Journal of Fluid Mechanics*, v. 14, p. 336–368, 1962.

DAVIS, G. de V.; MALLINSON, G. An evaluation of upwind and central difference approximations by a study of recirculating flow. *Computers & Fluids*, v. 4, n. 1, p. 29 – 43, 1976. ISSN 0045-7930. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0045793076900104>>.

DAWES, W. N. et al. A practical demonstration of scalable, parallel mesh generation. In: *47th AIAA Aerospace Sciences Meeting & Exhibit*. Orlando, FL: [s.n.], 2009. p. 5–8.

DESHPANDE, M.; MILTON, S. Kolmogorov scales in a driven cavity flow. *Fluid Dynamics Research*, v. 22, n. 6, p. 359 – 381, 1998. ISSN 0169-5983. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0169598397000439>>.

FADLUN, E. A. et al. Finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, v. 161, p. 35–60, 2000.

FALTSCHEART, R. B. et al. Numerical simulation of flow interference between two circular cylinders in tandem. *International Conference on Offshore Mechanics and Arctic Engineering*, v. 19, p. 1–8, 2000.

FERZIGER, J. H.; PERIC, M. *Computational methods for fluid dynamics*. 3. ed. [S.l.]: Springer, 2002. 431p p.

FLATSCHART, R. B.; MENEGHINI, J. R. Parallel simulation of a marine riser using mpi. In: *23th International Conference on Offshore Mechanics and Arctic Engineering*. [S.l.]: Proceedings of OMAE04, 2004. Best Algorithms Paper Award.

FREEMAN, C.; SOETE, L. *The economics of industrial innovation*. [S.l.], 1997. v. 3, 85–105; 265–285 p.

FRYXELL, B. et al. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophysical Journal*, v. 131, n. (Suppl.), p. 273–334, 2000.

FRYXELL, B. et al. Numerical simulations of thermonuclear flashes on neutron stars. *Nuclear Physics A*, v. 688, n. 1-2, p. 172 – 176, 2001. ISSN 0375-9474. Disponível em: <<http://www.sciencedirect.com/science/article/B6TVB-43GCDFN-19/2-/6b2aa696f186e14c1e52850cd37162d3>>.

GAFFNEY, R.; HASSAN, H.; SALAS, M. Euler calculations for wings using cartesian grids. *AIAA Paper 87–0356*, Jan. 1987.

GEE, M. et al. *ML 5.0 Smoothed Aggregation User's Guide*. [S.l.], 2006.

GEORGE, A.; LIU, J. *Computer solution of large sparse positive definite systems*. Prentice-Hall, 1981. (Prentice-Hall series in computational mathematics). Disponível em: <<http://books.google.com.br/books?id=pIYpAQAAMAAJ>>.

GHIA, U.; GHIA, K.; SHIN, C. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, v. 48, n. 3, p. 387 – 411, 1982. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999182900584>>.

GOLDSTEIN, D.; ADACHI, T.; SAKATA, H. Modeling a no-slip flow with an external force field. *Journal of Computational Physics*, New York, v. 105, p. 354, 1993.

GREENGARD, L. F.; HUANG, J. A new version of the fast multipole method for screened coulomb interactions in three dimensions. *Journal of Computational Physics*, v. 180, n. 2, p. 642 – 658, 2002. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/B6WHY-46G47HY-C/2/e30e42b51d48870e38c09e0713c52703>>.

GROPP, W. D.; KEYES, D. E.; TIDRIRI, M. D. Parallel implicit solvers for steady, compressible aerodynamics. In: *Parallel Computational Fluid Dynamics*. Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 1995. p. 391–399.

GROSSMAN, B.; WHITAKER, D. Supersonic flow computations using a rectangular-coordinate finite-volume method. *AIAA Paper 86–0442*, Jan. 1986.

GUERMOND, J. L. et al. Start-up flows in a three-dimensional rectangular driven cavity of aspect ratio 1:1:2 at $re = 1000$. *Journal of Fluid Mechanics*, v. 450, p. 169–199, 2002. Disponível em: <<http://dx.doi.org/10.1017/S0022112001006383>>.

HAASE, G. et al. Algebraic multigrid methods based on element preconditioning. *International Journal of Computer Mathematics*, v. 78, n. 4, p. 575–598, 2001. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/00207160108805133>>.

HENDRICKSON, B.; DEVINE, K. Dynamic load balancing in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, v. 184, n. 2-4, p. 485 – 500, 2000. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/B6V29-3YW270P-J/2/bbdf31e2be710e61790593b96a2da1b>>.

HEROUX, M. et al. *An Overview of Trilinos*. [S.l.], 2003.

HILBERT, D. Ueber die stetige abbildung einer line auf ein flächenstück. *Mathematische Annalen*, <http://www.springerlink.com/content/v1u6427kk33k8j56>, v. 38, n. 3, p. 459–460, 1891.

HUANG, J.; GREENGARD, L. A fast direct solver for elliptic partial differential equations on adaptively refined meshes. *SIAM Journal on Scientific Computing*, v. 21, n. 2, p. 1551 – 1566, Julho 2000. Disponível em: <<http://epubs.siam.org/sam-bin/dbq/article/34623>>.

HWANG, J. Y.; YANG, K. S. Numerical study of Taylor-Couette flow with an axial flow. *Computer & fluids*, v. 33, p. 97–118, 2004.

JURIC, D.; TRYGGVASON, G. A front-tracking method for dendritic solidification. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 123, p. 127–148, January 1996. ISSN 0021-9991. Disponível em: <<http://portal.acm.org/citation-.cfm?id=227647.227661>>.

KALLINDERIS, J. G.; BARON, J. R. Adaptation methods for a new navier-stokes algorithm. In: *AIAA 8th Computational Fluid Dynamics Conference*. [S.l.: s.n.], 1987.

Karman Jr., S. Splitflow: A 3d unstructured cartesian prismatic grid cfd code for complex geometries. *AIAA Paper 95–0343*, Jan. 1995.

KATO, Y.; KAWAI, H.; TANAHASHI, T. Numerical flow analysis in a cubic cavity by the gsmac finite-element method : In the case where the reynolds numbers are 1000 and 3200. *JSME international journal. Ser. 2, Fluids engineering, heat transfer, power, combustion, thermophysical properties*, v. 33, n. 4, p. 649–658, 1990. ISSN 09148817. Disponível em: <<http://ci.nii.ac.jp/naid/110002492540/>>.

KAWAGUTI, M. Numerical solution of the navier-stokes equations for the flow in a two-dimensional cavity. *Journal of the Physical Society of Japan*, The Physical Society of Japan, v. 16, n. 11, p. 2307–2315, 1961. Disponível em: <<http://jpsj.ipap.jp/link?JPSJ/16/2307/>>.

KIM, J.; KIM, D.; CHOI, H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, v. 171, p. 132–150, 2001.

KIM, J.; MOIN, P. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of Computational Physics*, v. 59, n. 2, p. 308 – 323, 1985. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/B6WHY-4DD1VRS-146/2-/a3d27f0031b262180e467036ab03e9db>>.

KU, H. C.; HIRSH, R. S.; TAYLOR, T. D. A pseudospectral method for solution of the three-dimensional incompressible navier-stokes equations. *Journal of Computational Physics*, v. 70, n. 2, p. 439 – 462, 1987. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999187901902>>.

Le Moigne, Y. Adaptive mesh refinement sensors for vortex flow simulations. In: *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*. Finlândia: ECCOMAS, 2004. v. 1.

LIAO, C.-C. et al. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, v. 39, n. 1, p. 152 – 167, 2010. ISSN 0045-7930. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045793009001108>>.

LILLY, D. K. On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems. *Monthly Weather Review - American Meteorological Society*, San Diego, CA, v. 93, p. 24, Agosto 1965.

Lima e Silva, A. L. F.; Silveira Neto, A. da; DAMASCENO, J. J. R. Numerical simulation of two-dimensional flows over a circular circular cylinder using the immersed boundary method. *Journal of Computational Physics*, v. 189, p. 351–370, 2003.

LIMA, R. S. de. *Desenvolvimento e implementação de malhas adaptativas bloco-estruturadas para computação paralela em mecânica dos fluidos*. 90 f. Tese (Doutorado) — UFU, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2012.

LIU, G. R. *Mesh Free Methods: Moving Beyond the Finite Element Method*. 1. ed. [S.l.]: CRC Press, 2002. Hardcover. ISBN 0849312388.

LOPES, R. K. D. *Análise de estruturas sujeitas a vibrações induzidas por vórtices*. Dissertação (Mestrado) — COPPE/UFRJ, Rio de Janeiro, Brasil, 2006.

LOURENÇO, M. A. de S. *Simulação numérica de escoamentos hipersônicos sobre corpos rombudos pelo método de elementos finitos*. 106 f. Dissertação (Mestrado) — Unesp, Campus de Ilha Solteira, Ilha Solteira, SP, Brasil, 2007.

LUEPTOW, R. M.; DOCTER, A.; MIN, K. Stability of axial flow in an annulus with a rotating inner cylinder. *Physics of Fluids*, v. 4, p. 2446–2455, nov. 1992.

LÖHNER, R. An adaptive finite element scheme for transient problems in cfd. *Computer Methods in Applied Mechanics and Engineering*, v. 61, p. 323–338, 1987.

MACCURDY, E. *The notebooks of Leonardo da Vinci*. New York: Reynal & Hitchcock, 1938.

MARIANO, F. P. *Simulação de escoamentos não periódicos utilizando a metodologia pseudo-espectral de Fourier acoplado com o método da fronteira imersa*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2007.

MELTON, J. E. *Automated three-dimensional cartesian grid generation and euler flow solutions for arbitrary geometries*. Tese (Doutorado) — Univ. CA, Univ. CA. Davis CA, 1996.

MIGEON, C.; PINEAU, G.; TEXIER, A. Three-dimensionality development inside standard parallelepipedic lid-driven cavities at $re=1000$. *Journal of Fluids and Structures*, v. 17, n. 5, p. 717 – 738, 2003. ISSN 0889-9746. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0889974603000094>>.

MIGEON, C.; TEXIER, A.; PINEAU, G. Effects of lid-driven cavity shape on the flow establishment phase. *Journal of Fluids and Structures*, v. 14, n. 4, p. 469 – 488, 2000. ISSN 0889-9746. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0889974699902827>>.

MITTAL, R. et al. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 227, n. 10, p. 4825–4852, maio 2008. ISSN 0021-9991. Disponível em: <<http://dx.doi.org/10.1016/j.jcp.2008.01.028>>.

Mohd-Yusof, J. Combined Immersed-Boundary/B-Spline methods for simulations of flow in complex geometries. *CTR Annual Research Briefs*, Stanford, CA, 1997.

Mohd-Yusof, J. Development of immersed boundary methods for complex geometries. *CTR Annual Research Briefs*, Stanford, CA, p. 325–336, 1998.

NAKAHASHI, K. An automatic grid generator for the unstructured upwind method. In: *AIAA 9th Computational Fluid Dynamics Conference*. [S.l.: s.n.], 1989.

OLIVEIRA, G. P. *Desenvolvimento de interface gráfica para o solver de dinâmica dos fluidos computacional Fluids-3D*. 137 f. Dissertação (Mestrado) — Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2007.

OLIVEIRA, J. E. S. *Método da fronteira imersa aplicado à modelagem matemática e simulação numérica de escoamentos turbulentos sobre geometrias móveis e deformáveis*. 164 f. Tese (Doutorado) — Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2006.

OLIVEIRA, J. E. S. et al. Simulação numérica do escoamento a baixo Reynolds sobre o cilindro de diâmetro variável usando MFI/MFV. In: *Congresso Brasileiro de Ciências Térmicas - ENCITE2004*. Rio de Janeiro: Anais do ENCITE2004, 2004. v. 1, p. 1–12.

OLIVEIRA, J. E. S. et al. Comparative analysis between different methodologies of turbulence modeling: URANS, DES e LES. In: *Escola Brasileira de Transição e Turbulência*. Porto Alegre: Anais da ETT2004, 2004. v. 1, n. ETT2004, p. 1–10.

PADILLA, E. L. M.; Silveira Neto, A. da. Desenvolvimento de um código computacional tridimensional para solução de escoamentos em transição e turbulência. In: *Proceedings of the XXVII Iberian Latin-American Congress on Computational Methods in Engineering (CILAMCE)*. [S.l.]: Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering, 2005.

PADILLA, E. L. M.; Silveira Neto, A. da. Large-eddy simulation of the transition to turbulence in natural convection in a horizontal annular cavity. *International Journal of Heat and Mass Transfer*, v. 10, 2008. Meio de divulgação: Impresso; ISSN/ISBN: 00179310.

PAN, D.; SHEN, T.-T. Computation of incompressible flows with immersed bodies by a simple ghost cell method. *International Journal for Numerical Methods in Fluids*, John Wiley & Sons, Ltd., v. 60, n. 12, p. 1378–1401, 2009. ISSN 1097-0363. Disponível em: <<http://dx.doi.org/10.1002/fld.1942>>.

PARHAMI, B. *Introduction to Parallel Processing*. Algorithms and architectures. New York: Kluwer Academic Publishers, 2002. 530 p. (Plenum Series in Computer Science). ISBN: 0-306-46964-2.

PEMBER, R. B. et al. Adaptive cartesian grid methods for representing geometry in inviscid compressible flow. *AIAA Paper 91-1542*, Junho 1991.

PERAIRE, J.; PEIRO, J.; MORGAN, K. Adaptive remeshing for three-dimensional compressible flow calculations. *Journal of Computational Physics*, 1992.

PESKIN, C. S. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, New York, v. 10, p. 252–271, 1972.

PESKIN, C. S. Numerical analysis of the blood flow in the heart. *Journal of Computational Physics*, New York, v. 25, p. 220, 1977.

PINHO, A. L. S. de. *Redução de tensões em risers rígidos de plataformas TLP*. Dissertação (Mestrado) — COPPE/UFRJ, Rio de Janeiro, Brasil, 2001.

POPINET, S. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics*, v. 190, n. 2, p. 572–600, 2003. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/B6WHY-49324BB-3/2/c5955123de59e888754d5c11c3bd6877>>.

POPINET, S.; RICKARD, G. A tree-based solver for adaptive ocean modelling. *Ocean Modelling*, v. 16, n. 3-4, p. 224 – 249, 2007. ISSN 1463-5003. Disponível em: <<http://www.sciencedirect.com/science/article/B6VPS-4MC14YJ-1/2-/7a845021b7e7874aa500e899a00a0b5b>>.

PRESS, W. H. et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3. ed. Cambridge University Press, 2007. Hardcover. ISBN 0521880688. Disponível em: <<http://www.worldcat.org/isbn/0521880688>>.

PURVIS, J.; BURKHALTER, J. Prediction of critical mach number for store configurations. *AIAA Journal*, v. 17, p. 11, 1979.

QUIRK, J. J. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers and Fluids*, v. 23, p. 125–142, 1992.

RICKER, P. M. A direct multigrid poisson solver for oct-tree adaptive meshes. *The Astrophysical Journal Supplement Series*, v. 176, n. 1, p. 293, 2008. Disponível em: <<http://stacks.iop.org/0067-0049/176/i=1/a=293>>.

ROMA, A. M.; PESKIN, C. S.; BERGER, M. J. An adaptive version of the immersed boundary method. *Journal of Computational Physics*, v. 153, n. 2, p. 509 – 534, 1999. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/B6WHY-45GMW70-2M/2-/6b004410bc469b61cd3bb807d4d9b3b6>>.

SAAD, Y.; SCHULTZ, M. H. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 7, n. 3, p. 856–869, jul. 1986. ISSN 0196-5204. Disponível em: <<http://dx.doi.org/10.1137/0907058>>.

SAIKI, E. M.; BIRINGEN, S. Numerical simulation of a cylinder in a uniform flow: application of a virtual boundary method. *Journal of Computational Physics*, Colorado, v. 123, p. 450–465, 1996.

Shankar, P. N.; Deshpande, M. D. Fluid Mechanics in the Driven Cavity. *Annual Review of Fluid Mechanics*, v. 32, p. 93–136, 2000.

SHEU, T.; TSAI, S. Flow topology in a steady three-dimensional lid-driven cavity. *Computers & Fluids*, v. 31, n. 8, p. 911 – 934, 2002. ISSN 0045-7930. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045793001000834>>.

SHU, C.; LIU, N.; CHEW, Y. T. A novel immersed boundary velocity correction-lattice boltzmann method and its application to simulate flow past a circular cylinder. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 226, p. 1607–1622, October 2007. ISSN 0021-9991. Disponível em: <<http://portal.acm.org/citation.cfm?id=1290206.1290439>>.

SMAGULOV, S. Fictitious Domain Method for Navier-Stokes equation. *Preprint CC SA USSR*, p. 68, 1979.

SOUZA, A. M. de; SOUZA, F.; SILVEIRA NETO, A. da. Métodos pseudo-espectrais e diferenças finitas de segunda ordem para análise da transição à turbulência em jatos circulares livres. In: *ENCIT2006*. Curitiba: ENCIT2006, 2006. p. 1–11.

STEENSLAND, J. et al. Towards an adaptive meta-partitioner for parallel samr applications. In: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing System*. Las Vegas: [s.n.], 2000. p. 425 – 430.

STONE, H. L. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM Journal on Numerical Analysis*, v. 5, p. 530 – 558, 1968.

STÜBEN, K. A review of algebraic multigrid. *J. Comput. Appl. Math.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 128, p. 281–309, March 2001. ISSN 0377-0427. Disponível em: <[http://dx.doi.org/10.1016/S0377-0427\(00\)00516-1](http://dx.doi.org/10.1016/S0377-0427(00)00516-1)>.

SU, S.-W.; LAI, M.-C.; LIN, C.-A. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers & Fluids*, v. 36, n. 2, p. 313–324, 2007. ISSN 0045-7930. Disponível em: <<http://www.sciencedirect.com/science/article/B6V26-4J2TVG2-2/2/5fadc6d60893e5db5264e90a7945f202>>.

TAYLOR, G. I. Stability of a viscous liquid contained between two rotary cylinders. *Philosophical Transactions of the Royal Society of London*, p. 289–343, 1923. A223.

THOMPSON, J. F. *Numerical grid generation*. [S.l.]: Elsevier Science Publishing Company, 1982.

TU, T. et al. From mesh generation to scientific visualization: an end-to-end approach to parallel super-computing. *SC2006*, Tampa, FL, 2006.

TÓTH, G.; ROE, P. Divergence- and curl-preserving prolongation and restriction formulas. *Journal of Computational Physics*, v. 180, n. 2, p. 736 – 750, 2002. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0021999102971202>>.

TÖLKE, J.; KRAFCZYK, M. Teraflop computing on a desktop pcs with gpus for 3d cfd. v. 22, n. 7, p. 443–456, Agosto 2008.

UHLMANN, M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 209, p. 448–476, November 2005. ISSN 0021-9991. Disponível em: <<http://dx.doi.org/10.1016/j.jcp.2005.03.017>>.

UNVERDI, S. O.; TRYGGVASON, G. A front tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, Michigan, v. 100, p. 25–37, 1992.

Van Doormaal, J. P.; RAITHBY, G. D. An evaluation of the segregated approach for predicting incompressible fluid flows. *Presented at the National Heat Transfer Conference*, 1985.

VANELLA, M.; BALARAS, E. A moving-least-squares reconstruction for embedded-boundary formulations. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 228, n. 18, p. 6617–6628, out. 2009. ISSN 0021-9991. Disponível em: <<http://dx.doi.org/10.1016/j.jcp.2009.06.003>>.

VANELLA, M.; RABENOLD, P.; BALARAS, E. A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems. *J. Comput. Phys.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 229, n. 18, p. 6427–6449, set. 2010. ISSN 0021-9991. Disponível em: <<http://dx.doi.org/10.1016/j.jcp.2010.05.003>>.

VEDOVOTO, J. M. *Modelagem matemática de simulação numérica de escoamentos incompressíveis sobre geometrias complexas tridimensionais utilizando o método da fronteira imersa*. 125 f. Dissertação (Mestrado) — Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2007.

VILLAR, M. M. *Análise numérica detalhada de escoamentos multifásicos bidimensionais*. Tese (Doutorado) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2007.

WANG, Z.; FAN, J.; LUO, K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, v. 34, n. 3, p. 283 – 302, 2008. ISSN 0301-9322. Disponível em: <<http://www.sciencedirect.com/science/article/B6V45-4R0CKKW-2/2/4332c9bdb64aa70d6a324a75a556f1f1>>.

WARREN, M. S.; SALMON, J. K. A parallel hashed oct-tree n-body algorithm. In: *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*. New York, NY, USA: ACM, 1993. (Supercomputing '93), p. 12–21. ISBN 0-8186-4340-4. Disponível em: <<http://doi.acm.org/10.1145/169627.169640>>.

WERELEY, S. T.; LUEPTOW, R. M. Velocity field for Taylor-Couette flow with an axial flow. *Physics of Fluids*, v. 11, p. 3637–3649, dez. 1999.

WHITE, F. M. *Viscous fluid flow*. New York: McGraw-Hill, 1974.

WILLDEN, R. H. J.; GRAHAM, J. M. R. Multi-modal vortex-induced vibrations of a vertical riser pipe subject to a uniform current profile. *Eur. J. Mech.*, v. 23, n. 1, p. 209–218, 2004.

YE, T. et al. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, v. 156, p. 209–240, 1999.

ZEEUW, D. D.; POWELL, K. G. An adaptively refined cartesian mesh solver for the euler equations. *Journal of Computational Physics*, v. 104, p. 56–68, 1991.

ZHANG, Y.; WEBBER, R. E. Space diffusion: an improved parallel halftoning technique using space-filling curves. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1993. (SIGGRAPH '93), p. 305–312. ISBN 0-89791-601-8. Disponível em: <<http://doi.acm.org/10.1145/166117.166156>>.

ÁSSI, G. R. da S. *Estudo experimental do efeito de interferência no escoamento ao redor de cilindros alinhados*. 250 f. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2005.

APÊNDICE A – Paralelização do aplicativo com malha retilínea

Neste apêndice apresenta-se detalhes do aplicativo utilizado como base para implementação daquele com malha adaptativa, o qual emprega malhas cartesianas retilíneas. Os detalhes de modelagem e/ou de implementação são os mesmos apresentados para as simulações com malha adaptativa. Alguns detalhes, provenientes da malha não ser uniforme, são apresentados nas seções a seguir, assim como os resultados determinados com a malha retilínea (Apêndice B).

A.1 Discretização espacial

Assim como no código com refinamento adaptativo, apresentado na Seção 4.2, as derivadas nas interfaces do volume de controle são aproximadas por diferenças centradas. A partir da Fig. A.1, a derivada da componente u_i para a face w pode ser escrita na forma:

$$\left(\frac{\partial u_i}{\partial x_i}\right)_w = \frac{u_{iP} - u_{iW}}{\Delta x_{iw}} \quad (\text{A.1})$$

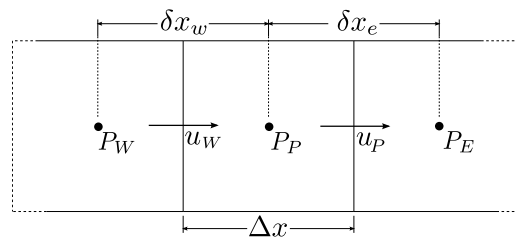


Figura A.1: Esquema para discretização ponderada dos termos advectivos por diferenças centradas.

Entretanto, para a malha retilínea, no cálculo envolvendo os termos advectivos é adotada uma média ponderada, semelhante à utilização anterior por diferenças centradas. Essa ponderação só se aplica para casos com malha não-uniforme. Para a face w , fica:

$$u_{iw} = \alpha u_{iP} + (1 - \alpha)u_{iW} \quad (\text{A.2})$$

$$\alpha = \frac{\Delta x}{2\delta x_w} \quad (\text{A.3})$$

A.2 Solução do Sistema de Equações Lineares

No código sequencial, sua solução é obtida utilizando os algoritmos SIP (*Strongly Implicit Procedure*) e MSIP, sendo que a última alternativa é mais rápida que a primeira, ainda que ocupe uma quantidade maior de memória (STONE, 1968). No código paralelizado, a solução de Poisson é realizada na mesma forma descrita na Seção 4.3, para o caso da solução através da construção de uma matriz global, exceto pelo fato de que aqui não são inseridas as interpolações do tipo fina-grossa, pois a malha é retilínea.

A.3 Paralelização

Na paralelização da malha retilínea, são empregados os mesmos operados de importação e exportação apresentados na Seção 4.5, uma vez que não existe perda de generalização devido ao fato que os blocos aqui são de malhas de mesmo nível, mas em processadores diferentes.

A.3.1 Balanço de carga com Zoltan

Para a malha retilínea, o balanceamento de carga foi realizado a partir do aplicativo Zoltan. Entre os algoritmos disponibilizados para balanceamento de carga, foi adotado o método de *Bisseção Recursiva por Coordenadas (RCB)*. O método se mostrou adequado para a malha em questão, além de possuir baixo custo computacional para criação das partições, como apresentado em (APPLIED; SIMON, 1991).

Para utilizar esse algoritmo, são necessárias a dimensão do domínio, a posição geométrica das células ou volumes que formam o domínio e um peso associado a cada célula. A utilização de pesos é uma vantagem para o balanceamento de carga de regiões dentro da

simulação que irão necessitar de maior processamento computacional, como as iterações de convergência para o método de fronteira imersa (Seção 2.4). Nesse caso, o balanceamento de carga poderia ou não levar em conta a presença de pontos lagrangianos entre os pontos eulerianos a serem particionados. Um exemplo de malha cartesiana bidimensional particionada sem e com a utilização de pesos é mostrado na Fig. A.2.

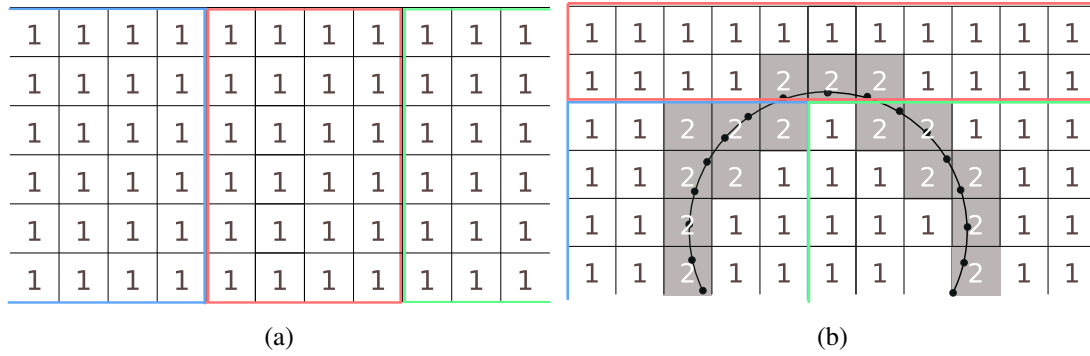


Figura A.2: Exemplo de particionamento de uma malha cartesiana bidimensional. Os pesos são definidos para cada volume para os casos de (a) balanceamento não ponderado e (b) ponderado.

Na figura Fig. A.2 é mostrada uma ponderação simples, no caso da necessidade de se atribuir pesos maiores na presença de uma interface como, por exemplo, pontos lagrangianos, onde é necessário um cálculo adicional em relação aos outros pontos eulerianos.

APÊNDICE B – Resultados para o caso com malha retilínea

Nesta seção, são apresentados alguns resultados relativos à paralelização do aplicativo apresentado em Padilla e Silveira Neto (2005). Para a malha euleriana, serão analisados 3 casos do problema de cavidade com tampa deslizante e para o método de fronteira imersa, o escoamento Taylor-Couette. Outros testes como os de *speedup* e de tempo de processamento, também são apresentados.

B.1 Cavidade quadrada com tampa deslizante

O sistema composto por uma cavidade de seção transversal quadrada (plano $x - y$) e com uma tampa deslizante na superfície superior, apresentado na Seção 5.2, é analisado novamente, utilizando-se a malha retilínea.

As comparações são realizadas com os trabalhos numéricos de Ku, Hirsh e Taylor (1987) Kato, Kawai e Tanahashi (1990), Babu e Korpela (1994), Sheu e Tsai (2002), Deshpande e Milton (1998), Padilla e Silveira Neto (2005). Os trabalhos de Kato, Kawai e Tanahashi (1990) e Sheu e Tsai (2002) são baseados no métodos de elementos finitos enquanto que Babu e Korpela (1994) utilizam o método de diferenças finitas. Nesse último trabalho, devido à importância acentuada das estruturas do escoamento no plano de simetria em z , foi utilizada uma malha de $63 \times 63 \times 33$. Ku, Hirsh e Taylor (1987) apresenta soluções de alta exatidão, determinadas a partir de métodos pseudo-espectrais para $Re = 100$, $Re = 400$ e $Re = 1000$. O método de diferenças finitas também é utilizado no trabalho de Deshpande e Milton (1998), que resolvem o problema para diferentes razões de aspecto e números de Reynolds.

Na Fig. B.1 é apresentada a distribuição para a componente de velocidade u sobre a linha vertical em $(x/L = 0,5; z/L = 0,5)$ para $Re = 100$ e $Re = 400$. Também para números de Reynolds de $Re = 100$ e $Re = 400$, a distribuição para a componente de velocidade v é apresentada na Fig. B.2. Esta se encontra sobre a linha horizontal em $(y/L = 0,5; z/L = 0,5)$. Os resultados para $Re = 100$ foram comparados apenas com aqueles de Ku, Hirsh e Taylor (1987),

apresentando boa concordância, exceto nas regiões próximas às paredes da cavidade. Isso pode ser devido à malha grosseira ($32 \times 32 \times 32$) nessas regiões, além da influência de interpolações lineares utilizadas no pós-processamento.

Para $Re = 400$, os resultados se mostraram de acordo com aqueles obtidos pelo método pseudo-espectral de Ku, Hirsh e Taylor (1987). A diferença entre os resultados de Padilla e Silveira Neto (2005) se deve, provavelmente, às resoluções distintas de malha de $(50 \times 50 \times 50)$ para aquela empregada neste trabalho de $(60 \times 60 \times 60)$. No presente trabalho, foi utilizada uma resolução maior buscando uma melhor concordância com os resultados de Ku, Hirsh e Taylor (1987).

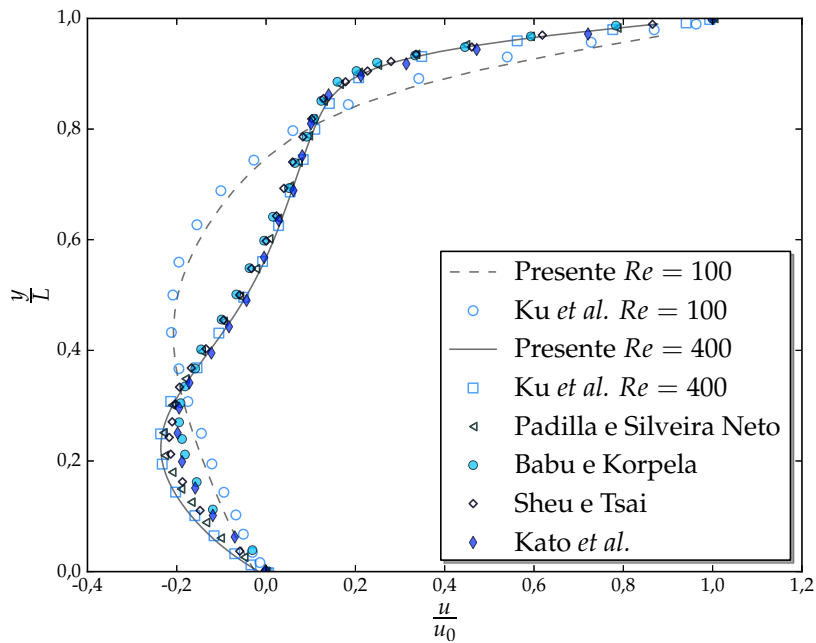


Figura B.1: Perfis da componente de velocidade u retirado do centro do cubo, passando pelo ponto $(x/L = 0,5; z/L = 0,5)$. As comparações são feitas para $Re = 100$ e $Re = 400$.

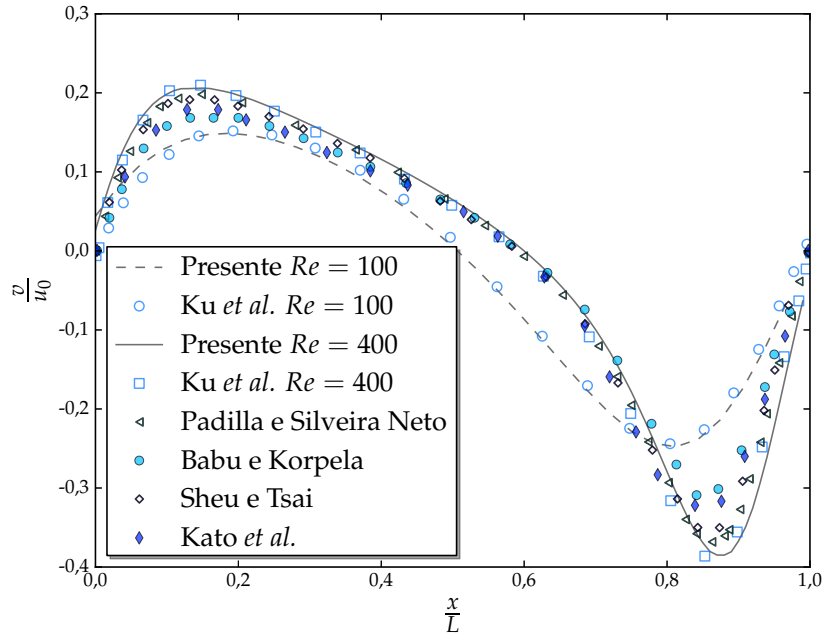
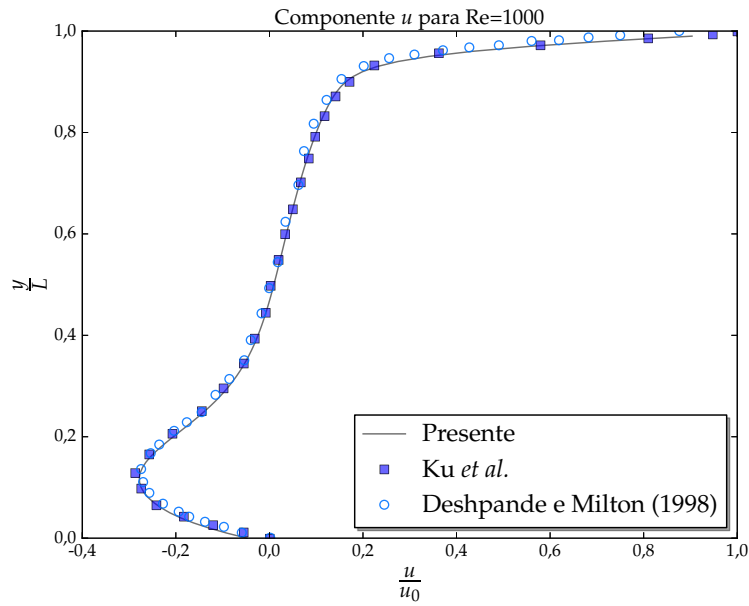
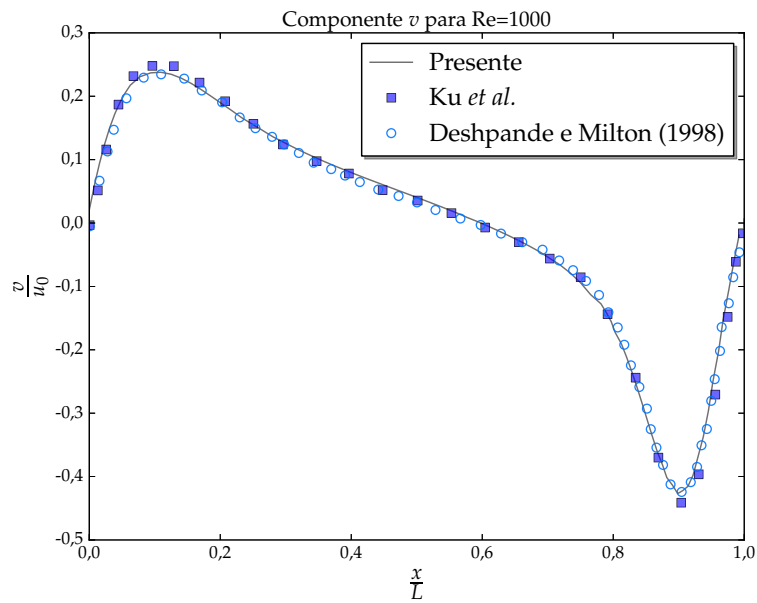


Figura B.2: Perfis da componente de velocidade v retirado do centro do cubo, passando pelo ponto $(y/L = 0,5; z/L = 0,5)$. As comparações são feitas para $Re = 100$ e $Re = 400$.

Na Fig. B.3 são apresentados os resultados das comparações para $Re = 1000$. As comparações, assim como nos casos para $Re = 100$ e $Re = 400$, apresentam boa concordância com os perfis da referência. O perfil para velocidade u foi tomado sobre a linha vertical em $(x/L = 0,5; z/L = 0,5)$. A distribuição para a componente de velocidade v se encontra sobre a linha horizontal em $(y/L = 0,5; z/L = 0,5)$. Visualmente, os contornos das velocidades e da pressão para o caso sequencial são iguais ao paralelo, para os valores obtidos no regime permanente e, por isso, não são mostrados nas figuras.



(a)



(b)

Figura B.3: Comparação dos perfis das componentes de velocidade determinados no presente trabalho, com aqueles de Ku, Hirsh e Taylor (1987), Deshpande e Milton (1998) a $Re = 1000$: (a) Componente u sobre a linha vertical em y e passando pelo ponto $(x/L = 0,5; z/L = 0,5)$ e; (b) Componente v sobre a linha horizontal na direção x , cruzando o ponto $(y/L = 0,5; z/L = 0,5)$.

B.2 Escoamento de Taylor-Couette

Na Seção 5.4, é apresentado o escoamento de Taylor-Couette. Aqui apresenta-se os resultados para o mesmo escoamento utilizando uma malha retilínea, juntamente com comparações e os mesmos métodos de fronteira imersa empregados naqueles obtidos com a malha adaptativa.

Na Fig. B.4 é mostrado o resultado do particionamento da malha lagrangiana, como proposta na Seção 4.5. Embora no exemplo mostrado na Fig. B.4a a geometria seja simétrica em todas as direções, nota-se que o mapeamento da malha lagrangiana foi realizado com sucesso em função do balanço de carga euleriano. A figura ilustra o sistema usado para simular o problema de Taylor-Couette com $Ta = 100$ e $\Gamma = 2$ e malha euleriana com $70 \times 70 \times 40$ volumes. Para a mesma razão de aspecto, contornos da componente de velocidade $w = -0,1 m/s$ e $w = 0,1 m/s$, calculados com 12 processadores, são mostrados na Fig. B.4b. Qualitativamente, nota-se a ausência de artefatos numéricos, mesmo quando o particionamento produz blocos não simétricos. No presente trabalho, embora o espaço anular seja maior que o raio interno ($E > R_i$), a desestabilização da estrutura acima de Ta_{crit} , com a formação dos vórtices de Taylor-Couette permanece válida, uma vez que, para $\eta = 3,2i$, tem-se $Ta_{crit} = 65$ (LUEPTOW; DOCTER; MIN, 1992).

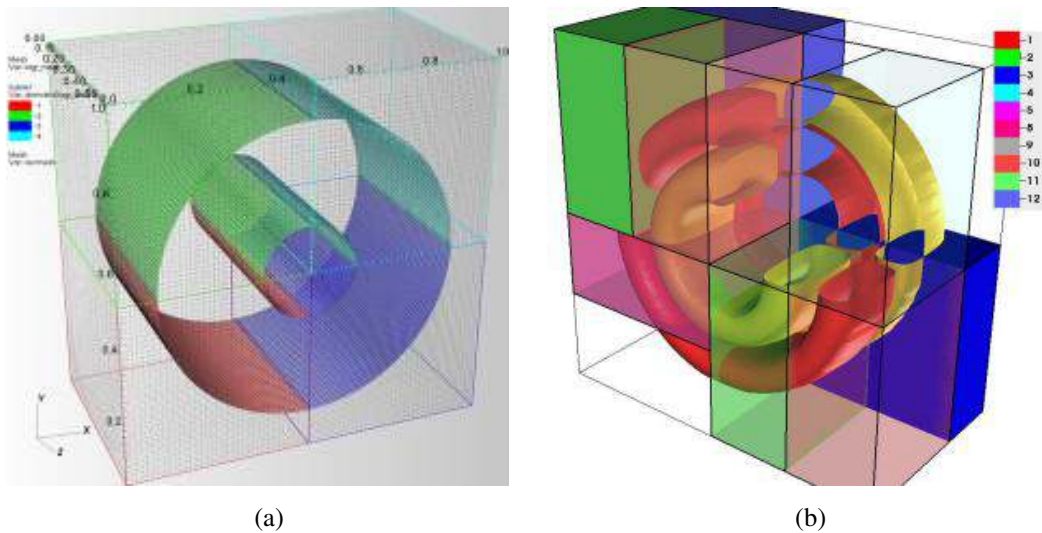


Figura B.4: Partição do domínio lagrangiano: (a) Casca cilíndrica imersa na malha euleriana de modo particionado em 4 processadores e; (b) Corte mostrando os contornos da componente w entre os blocos distribuídos entre 12 processadores.

As comparações são realizadas entre os resultados do presente trabalho com aqueles obtidos com o código em coordenadas cilíndricas de Padilla e Silveira Neto (2008). Este foi

validado com os resultados para escoamento anular de Couette, de Davey (1962), com os resultados experimentais por *Velocimetria por imagem de partículas* (PIV), de Wereley e Lueptow (1999), e com os resultados numéricos com FVM, de Hwang e Yang (2004). Em (DAVEY, 1962), são apresentados os resultados, por expansão em séries, somente para as componentes de velocidade axial e radial. Também são comparados com os resultados determinados a partir dos métodos de fronteira imersa de Modelo Físico Virtual (MFV) e de Multiforçagem Direta (MDF).

Na Fig. B.5 é apresentado o perfil da componente axial de velocidade adimensional $w/(R_i\omega)$, na direção radial, exatamente sobre a linha que passa no centro de um dos vórtices. Os valores simulados com o método MLS estão em conformidade com aqueles dos métodos MFV e MDF, exceto na região próxima ao cilindro interno onde o efeito de espessura do MDF reportado por Borges (2009) se mostra menos pronunciado no método MLS. Nota-se também que no regime permanente, os resultados para a razão de aspecto $\Gamma = 36$ são muito próximos daqueles para $\Gamma = 2$. O código em coordenadas cilíndricas possui malha não-uniforme e resolução muito maior entre cilindros ($N_r = 30$ volumes) do aquela no método de fronteira imersa ($N_r \leq 20$), que pode indicar uma possível causa da grande diferença entre os resultados.

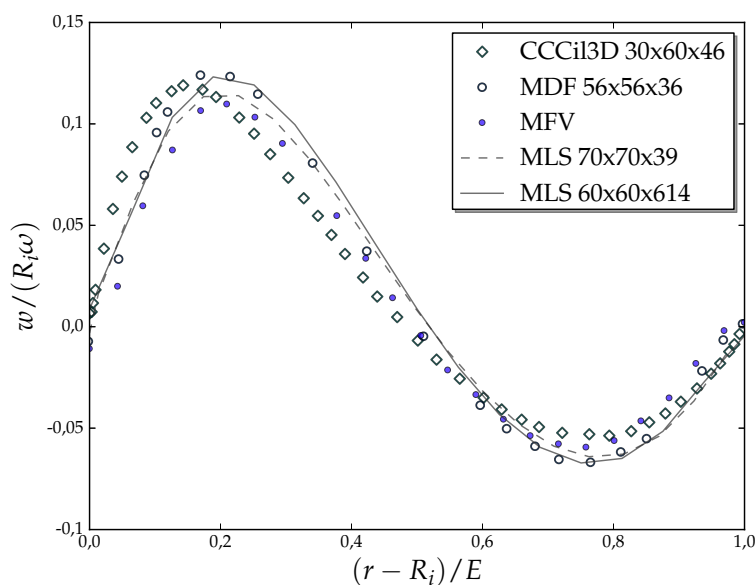


Figura B.5: Variação de velocidade axial adimensional $w/(R_i\omega)$ na direção radial para $Ta = 100$. Os valores são referentes à linha passando pelo núcleo de um dos vórtices.

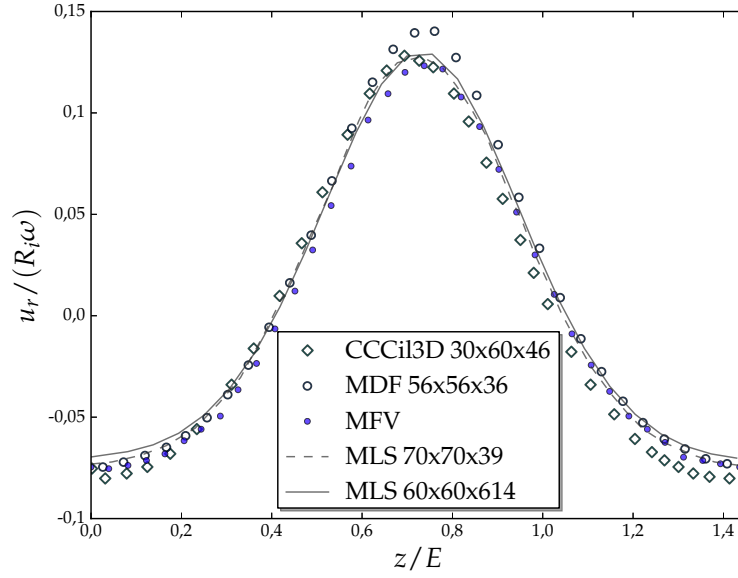


Figura B.6: Componente de velocidade radial adimensional $u_r/(R_i\omega)$ sobre a linha axial passando pelo centro dos vórtices para $Ta = 100$. Os valores são tomados no intervalo entre dois vórtices contrarotativos.

A variação da componente de velocidade radial adimensional $u_r/(R_i\omega)$ na direção axial é mostrada na Fig. B.6. Os dados são retirados na linha axial que passa por $R_i + E/2$, ponto $(x = 0,5; y = 0,76)$, nos limites compreendidos entre um par de vórtices de Taylor. Os perfis para todas as curvas relativas aos métodos de fronteira imersa são quase coincidentes nas fronteiras entre as estruturas na direção axial. Entretanto, no centro entre as estruturas, o método MDF apresenta valores levemente superestimados para a componente radial. Verifica-se também que, assim como os resultados apresentados da Fig. B.5, o método MLS tende a se aproximar do método MFV, à medida que é refinado, principalmente nas regiões menos suaves.

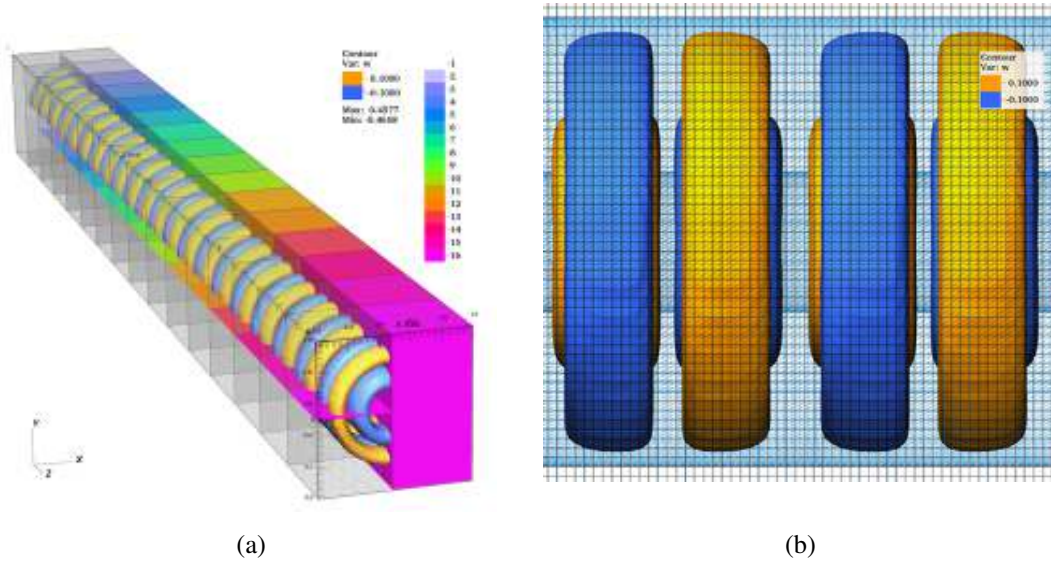


Figura B.7: Escoamento de Taylor-Couette a $Ta = 100$ e $\Gamma = 36$: (a) particionamento em 16 processadores utilizando o método *RCB* e; (b) Visão aproximada, mostrando detalhes da malha lagrangiana, de elementos triangulares e com nós não coincidentes com os volumes eulerianos.

Talvez o aspecto mais importante na utilização da metodologia de fronteira imersa seja o ganho em desempenho ao lidar com um número muito menor de nós e/ou elementos de uma malha não-estruturada. Tanto para problemas pequenos, como o caso de Taylor-Couette com $\Gamma = 2$, ou problemas um pouco maiores, como o caso com $\Gamma = 36$, pode-se notar visualmente, a partir da Fig. B.7, que o número de elementos da malha lagrangiana é muito menor que a quantidade total de volumes eulerianos. A estratégia utilizada para paralelização também foi muito facilitada ao se utilizar o método, uma vez que a malha lagrangiana, com todas as suas conexões, nós e faces, pode ser facilmente distribuída aos processadores correspondentes, em função do particionamento da malha euleriana, retilínea e na qual toda a estrutura é representada apenas por vetores representando os eixos coordenados.

De forma a verificar a implementação de paralelismo no aplicativo, foram comparados, para razão de aspecto $\Gamma = 36$, os valores do campo de velocidades para o tempo de 3,5 segundos de simulação, apresentando erros máximos de 10^{-8} entre o aplicativo sequencial e sua versão paralela. Entretanto, para esta razão de aspecto, utilizando a condição de contorno de periodicidade, o tempo físico para atingir o regime permanente aumenta, dos aproximadamente 4 segundos para $\Gamma = 2$, para cerca de 24 segundos até atingir variações máximas de 10^{-4} entre os valores de tempos consecutivos de simulação. A sonda foi posicionada no meio do espaço anular e no meio nas direções x e z , que para o caso $\Gamma = 36$ fica em $x = 0,5m$ e $z = 5m$. Entretanto, verificou-se que as posições das formações das primeiras estruturas também são consistentes

com o caso sequencial, ou seja, não existe um ruído, ao menos pronunciável, sendo gerado na interface dos blocos de malha de volumes diferentes, nem devido à atualização euleriana, nem devido à contribuição da força hidrodinâmica da fronteira imersa de outros processadores. A formação, para o caso com $\Gamma = 36$ e no tempo de 3,5 segundos é mostrada na Fig. B.8.

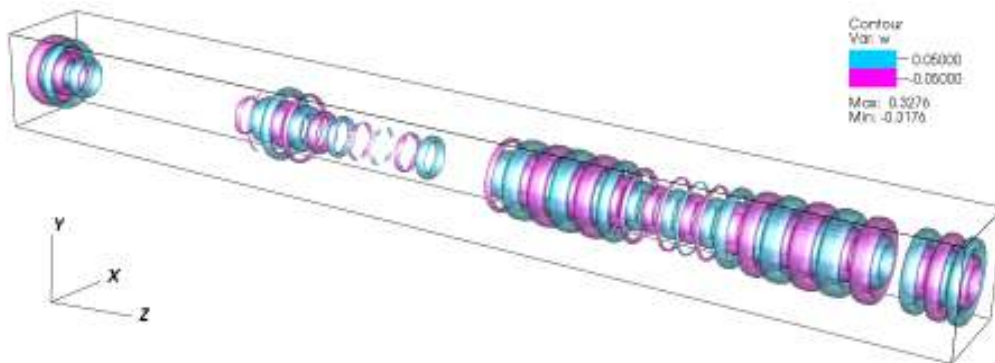


Figura B.8: Formação dos vórtices de Taylor-Couette para $Ta = 100$ e razão de aspecto $\Gamma = 36$. A assimetria na evolução dos vórtices na simulação sequencial, é igualmente reproduzida em paralelo.

Na Fig. B.9 são apresentados os campos vetoriais no plano $y-z$, em $x = 0,5m$, compreendendo um par de vórtices de Taylor, lembrando que embora as velocidades nos cantos da figura sejam baixas, foi utilizada a condição de contorno de periodicidade na direção z . Em coordenadas cilíndricas, como é apresentado na Fig. B.8, isto seria equivalente ao corte em um plano na direção azimutal. As linhas denotam a componente u na malha euleriana que, no plano escolhido, corresponde à componente azimutal de velocidades. Nota-se a característica aproximação dos núcleos dos vórtices na região que as maiores velocidades são rumo à parede do cilindro externo, enquanto que o fluxo na direção do cilindro interno possui velocidades baixas. A partir dos contornos de velocidade azimutal, pode-se concluir que essa diferença nos fluxos de massa, visualizada no plano $y-z$, é contrabalanceada pela velocidade azimutal. Claramente, o método de fronteira imersa implementado em paralelo, conseguiu reproduzir, qualitativamente, os principais aspectos do escoamento que foram capturados pelo aplicativo em coordenadas cilíndricas.

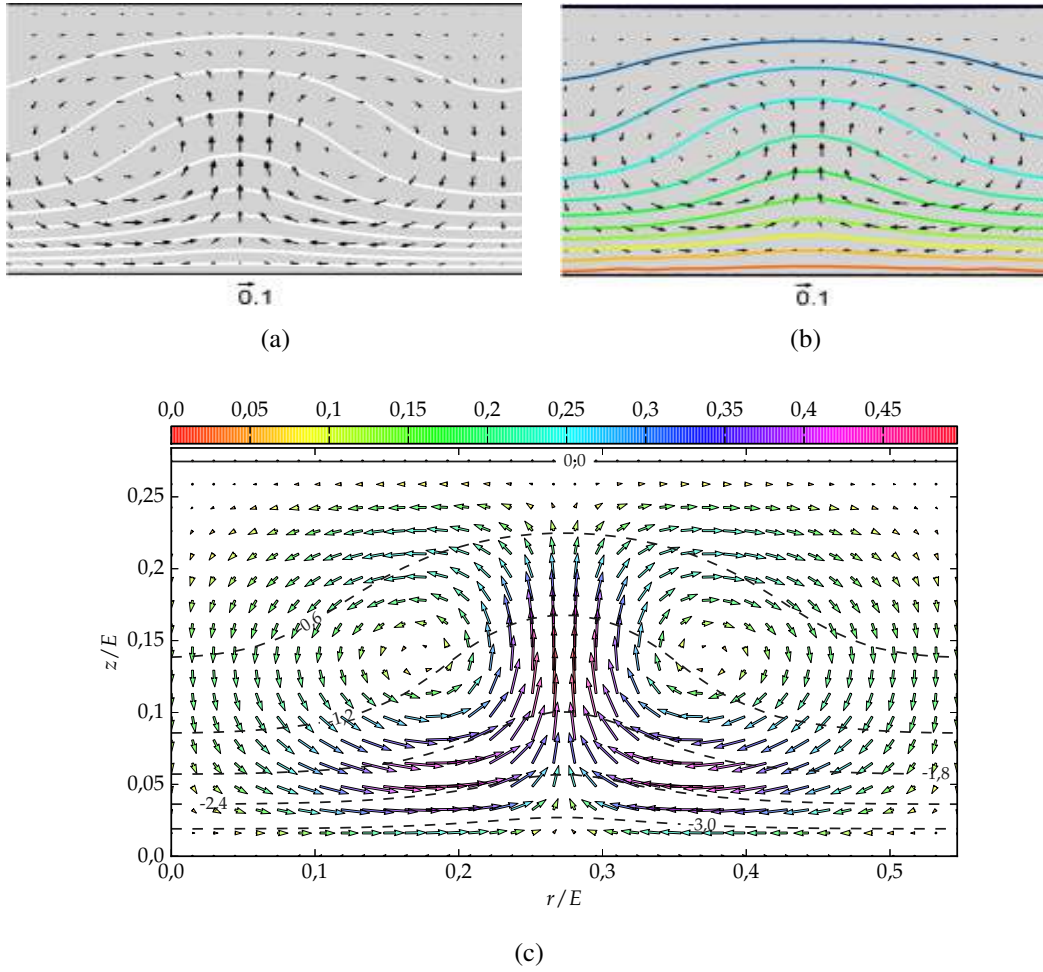


Figura B.9: Vetores de velocidade em um corte transversal na direção azimutal θ para o escoamento a $Ta = 100$ completamente desenvolvido. São mostradas também as isolinhas da componente de velocidade azimutal: (a) corte realizado no plano $r-z$ do código em coordenadas cilíndricas e no plano $y-z$ da malha euleriana e fronteira imersa com os métodos (b) MFV e (c) MLS.

B.3 Aspectos computacionais

Para a malha retilínea, a resolução da equação de Poisson para todos os casos foi feita com os pacotes *Epetra*, *AztecOO* e *ML* do *framework Trilinos*, apresentado na Seção 4.3. O método GMRES foi utilizado com os parâmetros predefinidos da biblioteca *AztecOO*, preconditionado pelo método de multigrid algébrico, da biblioteca *ML*. Foi empregado como parâmetro um relaxador baseado em *polinômios de Tchebychev* de grau 2. Para melhorar a transferência de dados em paralelo, o número de níveis no AMG é limitado a 4, para 16 a 32 processadores e a 3, para simulações com mais de 32 núcleos. Para até 8 processadores, o número de níveis é a parte inteira e positiva de $\log_2(N_{min})$, onde N_{min} é o mínimo entre o

número de volumes em cada direção coordenada. Outras opções, detalhes dos parâmetros de entrada, métodos de otimização do pacote para pré-condicionamento por multigrid *ML* podem ser encontradas em (GEE et al., 2006).

Nos testes, foram utilizadas malhas cartesianas de $32 \times 32 \times 32$, $60 \times 60 \times 60$ e $100 \times 100 \times 100$, nas simulações para $Re = 100$, $Re = 400$ e $Re = 1000$, respectivamente. Para $Re = 1000$, foi utilizada uma malha retilínea não uniforme, com aumento progressivo da resolução próxima às paredes de aproximadamente 6%. Os valores do número de iterações são dependentes dos parâmetros passados no método AMG. Verificou-se que o número de iterações é inversamente proporcional ao tempo gasto por iteração, além de ser dependente do problema e do número de processadores utilizados. A malha lagrangiana empregada no problema de Taylor-Couette é proporcional à euleriana, de dimensão $60 \times 60 \times 614$ nas três direções coordenadas.

Tabela B.3.1: Erros calculados com base na norma $L2$ para a equação de Poisson, com termo fonte a partir da equação manufaturada Eq. (5.3). Para as quatro resoluções da malha octree, com um nível de refinamento, os valores são apresentados para os casos sequencial e paralelo.

Re	Cavidade			Taylor-Couette
	100	400	10^3	$100(Ta)$
Malha	Uniforme	Uniforme	Refinada ^a	Uniforme
Resolução	$32 \times 32 \times 32$	$60 \times 60 \times 60$	$100 \times 100 \times 100$	$60 \times 60 \times 614$
Nº de processadores	2	8	16	16
Nº de iterações de solução	4	5	5	6

^aPor estiramento simples, próximo às paredes.

A Fig. B.10 mostra o balanceamento de carga para uma malha particionada em 64 processadores. A malha possui $128 \times 128 \times 128$ volumes nas três direções coordenadas e o resultado obtido mostra que o particionamento apresenta pouca diferença entre o número de volumes distribuídos para cada processador.

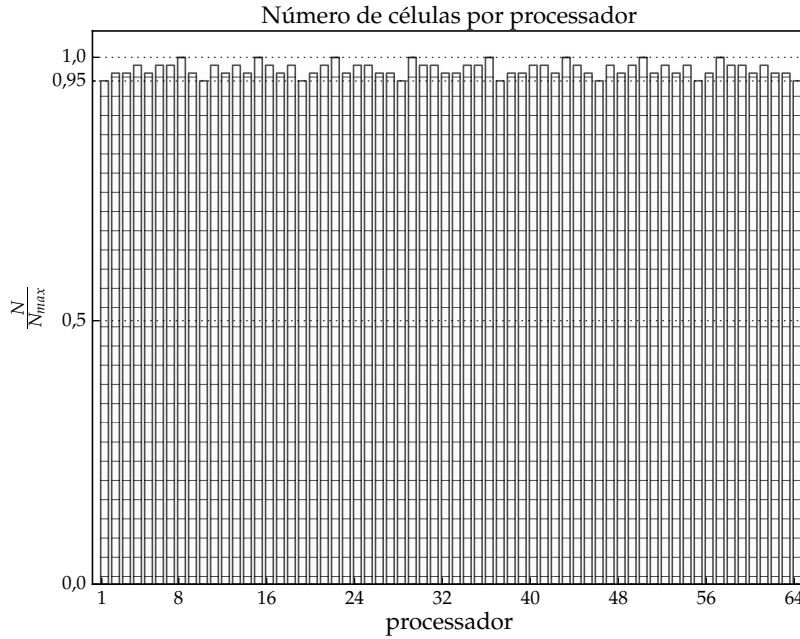


Figura B.10: Balanço de carga na solução paralela do problema de cavidade com tampa deslizando a $Re = 400$. Observa-se, nos valores normalizados entre 0 e 1, que o desbalanceamento máximo é de 5% entre as 64 partições.

Para a simulação a $Re = 1000$, além da comparação dos perfis de velocidade também foi realizado um teste de speedup. Para isso foi utilizada uma malha de $256 \times 256 \times 256$ volumes nas direções x, y e z , respectivamente, particionada para 2, 4, 8, 16, 32, 64 e 128 processos. Todos os casos utilizaram núcleos físicos, sem *Hyperthreading*¹, exceto para caso com 128 processos, que foi simulado com 64 núcleos virtuais. Para o teste, foram realizadas 25 passos de integração de solução no tempo, com o método *BiCGStab* para resolver da equação de Poisson. O ganho de processamento devido ao aumento no número de processadores para resolver o problema é apresentado na Fig. B.11.

¹Sistema computacional que processa duas ou mais tarefas simultaneamente, mais conhecido por SMT (*Simultaneous Multi Threading*). A intenção é utilizar melhor os recursos do processador.

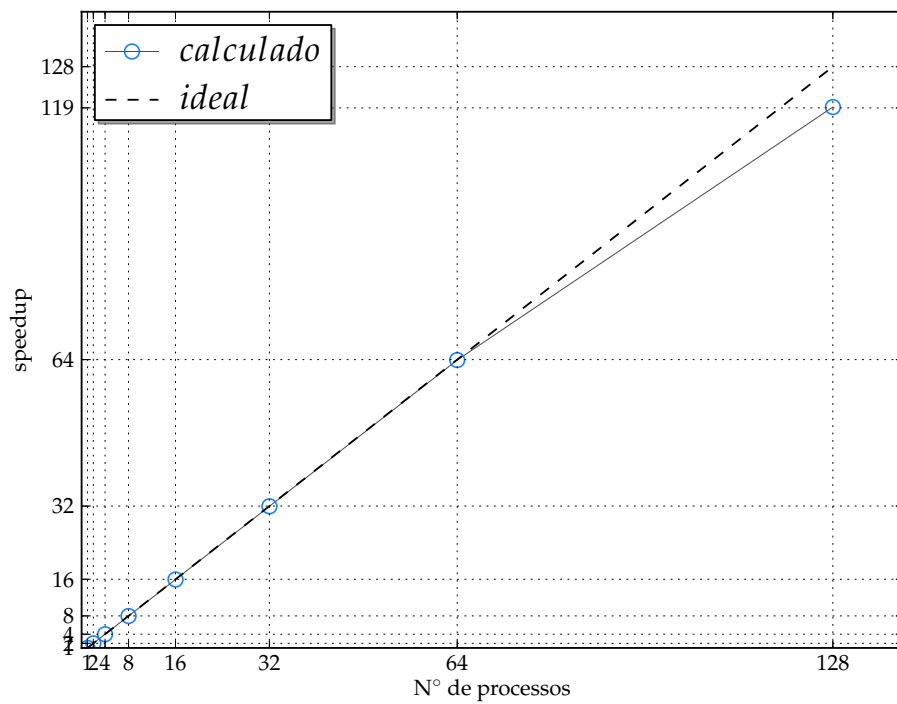


Figura B.11: Ganho em processamento (*speedup*) em função do número de processadores e curva ideal de ganho, para o problema de cavidade com tampa deslizante a $Re = 1000$.