

UNIVERSIDADE FEDERAL DO PARANÁ

RÉVERTON LUIS ANTUNES NEUNDORF

SOLUÇÃO DAS EQUAÇÕES DE NAVIER-STOKES UTILIZANDO OS MÉTODOS
MULTIGRID E MÉTODOS DE PROJEÇÃO PARALELIZADOS NO TEMPO E NO ESPAÇO

CURITIBA

2018

RÉVERTON LUIS ANTUNES NEUNDORF

SOLUÇÃO DAS EQUAÇÕES DE NAVIER-STOKES UTILIZANDO OS MÉTODOS
MULTIGRID E MÉTODOS DE PROJEÇÃO PARALELIZADOS NO TEMPO E NO ESPAÇO

Tese apresentada como requisito parcial à obtenção do
grau de Doutor em Ciências, pelo Programa de Pós-
Graduação em Métodos Numéricos em Engenharia,
Setores de Ciências Exatas e Tecnologia, Universidade
Federal do Paraná.

Área de concentração: *Mecânica Computacional*.

Orientador: Marcio Augusto Villela Pinto.

Coorientador: Leonardo Calvetti, Luciano Kiyoshi
Araki.

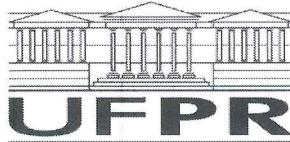
CURITIBA

2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

N494s	<p>Neundorf, Réverton Luis Antunes Solução das equações de Navier-Stokes utilizando os métodos multigrid e métodos de projeção paralelizados no tempo e no espaço [recurso eletrônico] / Réverton Luis Antunes Neundorf. – Curitiba, 2018.</p> <p>Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2018.</p> <p>Orientador: Marcio Augusto Villela Pinto – Coorientador: Leonardo Calvetti, Luciano Kiyoshi Araki.</p> <p>1. Navier-Stokes, Equações de. 2. Projeção. 3. Métodos multigrid (Análise numérica). 4. Método dos volumes finitos. 5. Paralelização. I. Universidade Federal do Paraná. II. Pinto, Marcio Augusto Villela. III. Araki, Leonardo Calvetti, Luciano Kiyoshi. IV .Título.</p>
	CDD: 515.353

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



MINISTÉRIO DA EDUCAÇÃO
SETOR CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MÉTODOS NUMÉRICOS
EM ENGENHARIA - 40001016030P0

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **REVERTON LUIS ANTUNES NEUNDORF** intitulada: **SOLUÇÃO DAS EQUAÇÕES DE NAVIER-STOKES UTILIZANDO OS MÉTODOS MULTIGRID E MÉTODOS DE PROJEÇÃO PARALELIZADOS NO TEMPO E NO ESPAÇO**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.
A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 15 de Dezembro de 2018.



MARCIO AUGUSTO VILLELA PINTO
Presidente da Banca Examinadora



ADMILSON TEIXEIRA FRÂNCIO
Avaliador Externo (UTFPR)



ROBERTO RIBEIRO SANTOS JUNIOR
Avaliador Externo (UFPR)



SIMONE DE FATIMA TOMAZZONI GONÇALVES
Avaliador Externo (UFPR)



COSMO DAMIÃO SANTIAGO
Avaliador Externo (UTFPR)



FRANCISCO JOSÉ GASPAR LORENZ
Avaliador Externo (UNIZAR)

*Dedico este trabalho exclusivamente
a uma pessoa: Neyde Abdalla
("Dona Dina")!*

Agradecimentos

Agradeço imensamente ao meu orientador, Prof. Dr. Marcio Augusto Villela Pinto, por ter aceitado me orientar neste trabalho, pelo apoio, didática, por todo o conhecimento recebido e, acima de tudo, pela infinita paciência que apresentou durante o desenvolvimento deste trabalho.

Agradeço ao meu co-orientador, Prof. Dr. Luciano Kiyoshi Araki, pela dedicação, conhecimento recebido e pelas valiosas (e precisas) observações sobre o texto desta tese.

Agradeço imensamente ao meu co-orientador, Dr. Leonardo Calvetti, por ter me mostrado o caminho da pós-graduação, pela paciência, pela confiança e por todo o conhecimento recebido.

Aos professores, funcionários e colegas do CESEC, pela amizade e disposição em ajudar.

Aos meus pais, pela compreensão e apoio incondicional.

Agradeço o apoio dos Kamers, que considero minha família.

A todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Agradeço especialmente a, excelentíssima, Neyde Abdalla (“Dona Dina”), por seu esforço extraordinário em promover a educação e a cultura. Sem sua influência este trabalho não existiria.

Resumo

Neste trabalho foi realizado um estudo do escoamento laminar bidimensional em regime transiente de fluido incompressível, modelado pelas equações de Navier-Stokes, utilizando método de projeção paralelizado no tempo e no espaço. Para tanto, utilizou-se o método de volumes finitos (MVF), com esquema de aproximação de segunda ordem (CDS), além de método de projeção com correção incremental na pressão. O *solver* utilizado foi o método de Gauss-Seidel *red-black*. Para a obtenção da solução foi empregado o método *multigrid* geométrico, com ciclo V, esquema de correção CS, restrição por ponderação completa, prolongação utilizando interpolação bilinear e número máximo de níveis para os casos estudados. Os métodos de projeção presentes na literatura, normalmente, apresentam avanço temporal sequencial, isto é, resolvem as equações de Navier-Stokes a cada passo de tempo. Neste trabalho são apresentadas novas formulações dos métodos de projeção paralelizados no tempo e no espaço. Nesta nova formulação as equações de Navier-Stokes foram aproximadas em todos os passos de tempo em paralelo. A paralelização no tempo foi realizada através da reformulação dos métodos de projeção e a paralelização no espaço foi realizada aplicando-se metodologia de particionamento do domínio. Com a paralelização do método de projeção no tempo e no espaço foi possível reduzir em até 12 vezes, utilizando 16 processadores, o tempo de CPU necessário para obter as soluções das equações de Navier-Stokes.

Palavras-chave: Navier-Stokes, métodos de projeção, *multigrid*, volumes finitos, paralelização.

Abstract

In this work it has been studied the two-dimensional laminar flow in a transient regime of an incompressible fluid modeled by the Navier-Stokes equations using a projection method parallelized in time and space. It was used the finite volume method (MVF) with a second order approximation scheme (CDS), as well as a projection method with incremental pressure correction. The solver used was the Gauss-Seidel red-black method. To obtain the solution, the geometric multigrid method was used, with V - cycle, correction scheme (CS), full-weighting restriction operator, prolongation by bilinear interpolations and maximum number of levels for the studied cases. The projection methods present in the literature usually show sequential temporal advancement, that is, they solve the Navier-Stokes equations at each time step. In this work, new formulations of the projection methods parallelized in time and space are presented, in this new formulation the Navier-Stokes equations were approximated in all time steps in parallel. Parallelization in time was accomplished through the reformulation of the projection methods and the parallelization of the space was accomplished applying a methodology of domain partitioning. With the parallelization of the projection method in time and space it was possible to reduce the CPU time needed to obtain the solutions of the Navier-Stokes equations by up to 12 times using 16 processors.

Keywords: Navier-Stokes, projection methods, multigrid, finite volume, parallelization.

Listas de Figuras

1.1	Discretização do domínio	17
1.2	Ordenação lexicográfica.	18
2.1	Processo de engrossamento de uma malha uniforme e razão de engrossamento $r = 2$	25
2.2	Diagrama do ciclo V para $L = 5$	25
2.3	Armazenamento adjacente na memória.	26
2.4	Nomenclatura dos vizinhos genéricos dos volumes da malha bidimensional uniforme.	27
2.5	Posições da pressão P e velocidades u e v na ordem lexicográfica.	28
2.6	Processo de restrição para variável centrada (pressão)	29
2.7	Processo de restrição para variáveis deslocadas (u e v)	29
2.8	Processo de prolongação para a variável pressão.	30
2.9	Processo de prolongação para variável deslocada u	31
2.10	Processo de prolongação para variável deslocada v	32
2.11	Particionamento da malha em subdomínios.	35
2.12	Divisão exata de um subdomínio.	37
2.13	Divisão não exata de um subdomínio.	38
2.14	Domínio com 49 volumes particionado em (a) 4 subdomínios e (b) 8 subdomínios.	38
2.15	Premissas da lei de Amdahl.	40
2.16	Limite superior de <i>speedup</i> para programas com frações paralelas entre 50% a 95%.	41
3.1	Domínio e condições de contorno.	44
4.1	Disposição das variáveis: velocidades nas faces e pressão no centro dos volumes.	45
4.2	Malha e ordenação lexicográfica das variáveis pressão e velocidades.	46
4.3	Volumes de controle.	47
4.4	Velocidade u de um volume com vizinhos fictícios e contornos.	57
4.5	Pressão de um volume com contornos.	60
5.1	Domínio <i>red-black</i> para dois passos de tempo consecutivos.	67
6.1	Ordem efetiva do erro (P_E) para as variáveis u , v e P	74
6.2	Taxas de convergência para as variáveis u , v e P	75
6.3	Tempo de CPU para <i>multigrid</i> versus <i>singlegrid</i>	76

6.4	Perfis das velocidades u e v em $x = 0,5$ e $y = 0,5$ para malhas de 1024×1024 .	78
6.5	Perfis das velocidades u e v em $x = 0,5$ e $y = 0,5$ para $Re = 5000$.	78
6.6	Comparativo entre iterações externas contra iterações do <i>solver</i> .	80
6.7	Ordem efetiva do erro (P_E) para todas as variáveis primitivas.	81
6.8	Taxas de convergência das variáveis primitivas u , v e P .	82
6.9	<i>Speedups</i> para os métodos <i>singlegrid</i> e <i>multigrid</i> para várias malhas e 16 <i>threads</i>	83
6.10	Escalabilidade para os métodos <i>singlegrid</i> e <i>multigrid</i> .	83
6.11	Paralelização espacial apresentada em Neundorf et al. (2016) <i>versus</i> paralelização temporal.	84

Lista de Tabelas

1.1	Comparação entre os métodos de solução de problemas em Engenharia (adaptada de Pletcher et al. (1997))	16
6.1	Taxas de convergência para os métodos sequenciais	74
6.2	Tempo de CPU <i>multigrid</i> (MG) <i>versus</i> <i>singlegrid</i> (SG)	76
6.3	Esforço computacional dos algoritmos <i>multigrid</i> e <i>singlegrid</i>	77
6.4	Diferenças percentuais de v em $y = 0,5$ para $Re = 5000$	79
6.5	Diferenças percentuais de u em $x = 0,5$ para $Re = 5000$	79

Lista de Acrônimos

CDS	<i>Central Difference Scheme</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Correction Scheme</i>
MG	<i>Multigrid</i>
MPE	<i>Minimum Perimeter Equi-partition</i>
MSP	<i>Minimum Striped Perimeter</i>
MVF	Método de Volumes Finitos
SG	<i>Singlegrid</i>
SPMD	<i>Single Program Multiple Data</i>
SW	<i>South-West</i>
TME	<i>Textbook Multigrid Efficiency</i>

Lista de Símbolos

$a.$	Coeficientes da matriz A : a_p, a_w, a_E, a_S, a_N
A	Matriz de coeficientes
$A_{i,j}^n$	Termos advectivos
b	Vetor de termos do sistema linear
b_P	Termo fonte do sistema
C	Número que “ativa” a correção do número de nós por subdomínio
$D_{i,j}^n$	Termos difusivos
E	Eficiência da paralelização
E_h	Erro numérico
e	Vetor contendo os valores do erro numérico
f	Parcela do processamento que pode ser melhorada (com a paralelização)
f_t	Índice que determina o fim de cada subdomínio
g_0, g_1	Termos advectivos
h	Tamanho dos elementos da malha
h_x	Tamanho dos elementos da malha na direção x
h_y	Tamanho dos elementos da malha na direção y
i_t	Índice que determina o início de cada subdomínio
I_h^{2h}	Operador de Restrição
I_{2h}^h	Operador de Prolongação
l_∞	Norma l_∞
l_1	Norma l_1
l_2	Norma l_2
L	Número de níveis
L_{\max}	Número máximo de níveis
L_x	Comprimento do domínio bidimensional na direção x
L_y	Comprimento do domínio bidimensional na direção y
m	Aceleração (fator de melhoria) do processamento
n	Número de processadores
N	Número de volumes da malha
N_t	Último passo de tempo considerado na simulação
N_{xp}	Número de volumes da pressão na direção x
N_{xu}	Número de faces da componente da velocidade u na direção x

N_{xv}	Número de faces da componente da velocidade v na direção x
N_x	Número de volumes da malha na direção x
N_y	Número de volumes da malha na direção y
N_{sub}	Número de volumes por subdomínio
P	Pressão do fluido
$P^{*,n}$	Pressão extrapolada na generalização do método de projeção
P_E	Ordem efetiva do erro
P_L	Ordem assintótica do erro
q	Parâmetro utilizado na generalização do método de projeção
r	Razão de engrossamento, parâmetro utilizado na generalização do método de projeção
\mathbf{r}	Vetor contendo valores do resíduo
Re	Número de Reynolds
S_P	<i>Speedup</i>
S_n	<i>Speedup</i> obtido com n processadores
S_{Amdahl}	<i>Speedup</i> previsto pela lei de Amdahl
t	Número que identifica o <i>thread</i> , variável temporal
t_c	<i>Thread</i> a partir do qual os subdomínios receberão um volume extra
\bar{T}	Número de <i>threads</i>
T'	Tempo total de execução do programa
T_i	Tempo inicial da simulação
T_P	Tempo de execução empregado em processamento paralelo
T'_P	Fração do tempo de execução devido ao processamento paralelo
T_S	Tempo de execução empregado em processamento serial
T'_S	Fração do tempo de execução devido ao processamento serial
t_n	Tempo de CPU utilizando n processadores
\mathbf{u}	Vetor que representa a solução do sistema linear. Vetor que representa a velocidade do escoamento
\mathbf{u}^t	Vetor que representa a velocidade temporária (ou auxiliar) do escoamento
u	Componente horizontal da velocidade \mathbf{u}
u_{pu}^h	Velocidade u na face pu da malha fina h
u_{pu}^{2h}	Velocidade u na face pu da malha grossa $2h$
\mathbf{u}	Aproximação do vetor solução \mathbf{u} .
v	Componente vertical da velocidade \mathbf{u}
v_{pv}^h	Velocidade v na face pv da malha fina h
v_{pv}^{2h}	Velocidade v na face pv da malha grossa $2h$
x, y	Variáveis do sistema cartesiano bidimensional

Símbolos gregos

$\alpha_0, \alpha_1, \alpha_2$	Coeficientes utilizados na generalização do método de projeção
β_1, β_0	Constante dos termos advetivos
χ	Parâmetro utilizado na generalização do método de projeção
ε	Tolerância
ϕ	Correção da pressão
ϕ_{pp}^h	Correção da pressão no volume pp da malha fina h
ϕ_{pp}^{2h}	Correção da pressão no volume pp da malha grossa $2h$
Ω	Domínio contínuo
Ω^h	Malha fina
Ω^{2h}	Malha grossa
ν_1	Número de pré-suavizações
ν_2	Número de pós-suavizações

Subscritos

i	Contador na direção x
j	Contador na direção y
pp	Índice lexicográfico da pressão
pu	Índice lexicográfico da componente horizontal da velocidade
pv	Índice lexicográfico da componente vertical da velocidade

Sobrescritos

$h, 2h, 3h, \dots$	Indicadores do nível de malha
k	Iteração do ciclo externo do método de projeção paralelizado
n	Passo de tempo

Sumário

1	Introdução	16
1.1	Generalidades em dinâmica dos fluidos computacional	16
1.2	Motivação	19
1.3	Revisão bibliográfica	19
1.3.1	Métodos de projeção para resolução das equações de Navier-Stokes	19
1.3.2	Paralelização do método <i>multigrid</i>	21
1.4	Objetivos	23
2	Fundamentação	24
2.1	Método <i>multigrid</i>	24
2.1.1	Considerações sobre o método <i>multigrid</i>	24
2.1.2	Ordenação lexicográfica	26
2.1.3	Processo de restrição	26
2.1.4	Processo de prolongação	28
2.1.5	Esquema de correção para duas malhas	31
2.2	Conceitos de paralelização	34
2.2.1	Subdomínios e balanceamento de carga	36
2.2.2	Lei de Amdahl e métricas de desempenho	39
3	Modelo matemático	42
4	Metodologia numérica.	45
4.1	Tipo de malha utilizada	45
4.2	Discretização espacial por volumes finitos	46
4.3	Métodos de projeção sequenciais no tempo	49
4.3.1	Método 1	51
4.3.2	Método 2	52
4.3.3	Generalização e Método 3	52
4.4	Coeficientes das velocidades	54
4.4.1	Termos difusivos na ordem lexicográfica	54
4.4.2	Termos advectivos na ordem lexicográfica	55
4.4.3	Gradiente na ordem lexicográfica	55
4.4.4	Coeficientes para as velocidades dos volumes internos	56
4.4.5	Coeficientes para as velocidades dos volumes com contornos	56

4.5	Coeficientes da pressão	58
4.5.1	Termos difusivos na ordem lexicográfica	58
4.5.2	Divergente na ordem lexicográfica	58
4.5.3	Coeficientes da pressão para os volumes internos	58
4.5.4	Coeficientes da pressão para os volumes com contornos	59
4.6	Condição de integrabilidade e estabilidade	61
4.7	Algoritmo do método de projeção sequencial	62
5	Métodos de projeção em paralelo.	63
5.1	Método de projeção sequencial de referência.	63
5.2	Métodos de projeção paralelizados no tempo.	64
5.3	Paralelização temporal no método <i>multigrid</i>	67
5.4	Paralelização espacial da atualização da pressão	67
5.5	Métodos de projeção paralelizados no tempo e no espaço	69
6	Resultados	71
6.1	Dados de implementação	71
6.2	Avaliação dos métodos de projeção sequenciais	72
6.2.1	Ordem efetiva	73
6.2.2	Taxa de convergência	73
6.2.3	<i>Multigrid versus singlegrid</i>	76
6.2.4	Resultados para o Problema 2.	77
6.3	Avaliação do método de projeção paralelizado	80
6.3.1	Número de iterações externas <i>versus</i> o número de iterações do <i>solver</i>	80
6.3.2	Ordem efetiva e taxa de convergência	81
6.3.3	Desempenho paralelo	81
6.3.4	Consolidação dos resultados	84
7	Conclusão	86
7.1	Constatações gerais	86
7.2	Contribuições	86
7.3	Trabalhos futuros	87
	Referências	88

1 Introdução

Neste capítulo são apresentadas generalidades em Dinâmica dos Fluidos Computacional (DFC), uma revisão da literatura sobre métodos de projeção, a motivação e objetivos do trabalho.

1.1 Generalidades em dinâmica dos fluidos computacional

Modelos matemáticos que representam fenômenos físicos, em geral, não possuem soluções analíticas conhecidas. Buscam-se então soluções via métodos numéricos, transformando-se os modelos contínuos em modelos discretos.

Além dos métodos analíticos e numéricos, denominados métodos teóricos porque trabalham com modelos matemáticos (Maliska, 2004), existe uma terceira estratégia que pode ser empregada na solução de problemas em Engenharia: os métodos experimentais. Cada estratégia possui vantagens e desvantagens, que dependem das características do problema estudado, mas devem ser consideradas complementares. A Tab. 1.1, extraída de Pletcher et al. (1997), traz uma comparação entre as três estratégias.

Tabela 1.1: Comparação entre os métodos de solução de problemas em Engenharia (adaptada de Pletcher et al. (1997))

Método	Vantagens	Desvantagens
Experimental	- mais realístico	<ul style="list-style-type: none"> - são exigidos equipamentos (empregados na medição) - problemas de escala - dificuldades de medição - custo operacional
Analítico	<ul style="list-style-type: none"> - solução aplicável a qualquer ponto no domínio - representação matemática do fenômeno real 	<ul style="list-style-type: none"> - restrita a geometrias e processos físicos simples - geralmente restrita a problemas lineares - erros de modelagem
Numérico	<ul style="list-style-type: none"> - não há restrição quanto à linearidade - geometrias e processos complicados - evolução temporal do processo 	<ul style="list-style-type: none"> - erros numéricos - prescrições das condições de contorno apropriadas - custo computacional - erros de modelagem

Os métodos numéricos consistem em resolver as equações diferenciais dos modelos matemáticos, que representam os fenômenos físicos, aproximando derivadas por expressões algébricas que contenham a função incógnita. O processo, de aproximação de derivadas por expressões algébricas, é denominado de discretização.

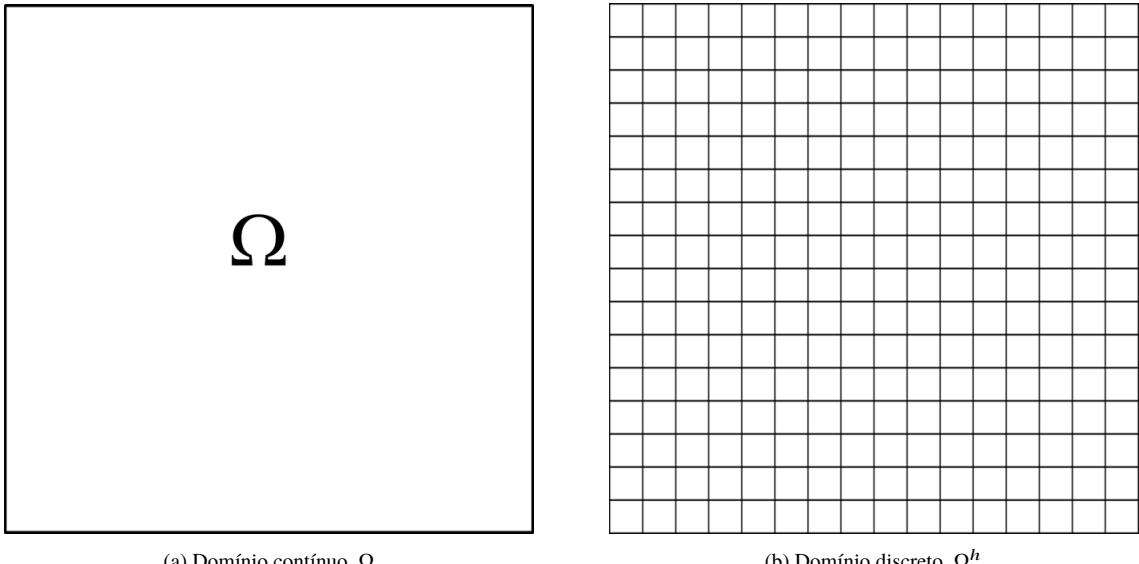


Figura 1.1: Discretização do domínio

Existem várias formas de se obter as expressões algébricas que aproximam derivadas, das quais três métodos são citados abaixo:

- Elementos finitos (Hughes, 2000; Reddy e Gartling, 2010);
- Diferenças finitas (Pletcher et al., 1997; Fortuna, 2000);
- Volumes finitos (Maliska, 2004; Versteeg e Malalasekera, 2007).

Neste trabalho será utilizado o método dos volumes finitos (Maliska, 2004; Versteeg e Malalasekera, 2007) que, em problemas bidimensionais cartesianos, consiste em particionar o domínio do problema $\{(x, y) \in R^2 : 0 \leq x \leq L_x \text{ e } 0 \leq y \leq L_y\}$ em subconjuntos de incógnitas (volumes) dados por $N = N_x N_y$, onde N_x e N_y são os números de volumes nas direções coordenadas x e y , respectivamente, e L_x e L_y determinam o comprimento do domínio de cálculo.

Considerando malhas estruturadas e uniformes por direção tem-se os pontos:

$$(x_i, y_j) = ((i-1)h_x, (j-1)h_y), \text{ com } h_x = \frac{L_x}{N_x} \text{ e } h_y = \frac{L_y}{N_y}, \quad (1.1)$$

onde $i = 1, \dots, N_x$, $j = 1, \dots, N_y$, e h_x e h_y representam o tamanho de cada volume (ou passo) nas direções coordenadas x e y , respectivamente. A este conjunto de pontos dados pela Eq. (1.1), dá-se o nome de malha e denota-se por Ω^h .

Diferentemente do método analítico, que permite calcular os valores das variáveis dependentes para infinitos pontos (solução contínua), o método numérico fornece valores para um número discreto de pontos (volumes) definido pela malha computacional (Mesquita, 2000). A Fig. 1.1(a) representa um domínio contínuo e a Fig. 1.1(b) o mesmo domínio, mas discreto.

Para localizar os volumes do domínio discreto (malha obtida após o particionamento) é necessário algum tipo de ordenação. A Fig. 1.2 exemplifica um tipo de ordenação conhecida como ordenação lexicográfica para $N_x = N_y = 8$.

Cada retângulo na Fig. 1.2 representa um volume na malha e é localizado por um índice (número) p .

57	58	59	60	61	62	63	64
49	50	51	52	53	54	55	56
41	42	43	44	45	46	47	48
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

Figura 1.2: Ordenação lexicográfica.

O processo de discretização do domínio e da equação diferencial do modelo matemático gera um sistema de equações algébricas do tipo:

$$A\mathbf{u} = \mathbf{b}, \quad (1.2)$$

onde A é a matriz dos coeficientes, \mathbf{b} é o vetor dos termos independentes e \mathbf{u} o vetor de incógnitas. A estrutura da matriz A depende do método de discretização e pode, em alguns casos, ter elementos diferentes de zero apenas na diagonal principal (e algumas paralelas a esta).

Várias técnicas computacionais são estudadas com o objetivo de resolver o sistema representado pela Eq. (1.2) com o menor custo computacional (tempo de CPU) e solução mais próxima da solução exata. Métodos diretos, como a eliminação de Gauss, resolvem razoavelmente bem sistemas lineares de pequeno porte (da ordem de 10^2 elementos), mas na prática não são recomendados, visto que, a matriz A apresenta ordem elevada em problemas de interesse na Engenharia, e o custo da inversão é alto (Golub e Van Loan, 2013). Para sistemas de grande porte, métodos iterativos são mais adequados (Burden e Faires, 2008), pois possuem custo computacional da ordem de $O(N^2/2)$, valor consideravelmente menor que o custo da ordem de $O(N^3)$ dos métodos diretos, onde N é o número de incógnitas. Neste contexto, os métodos de resolução de sistemas lineares são chamados de *solver*.

Os métodos *multigrid* pertencem à família dos métodos iterativos utilizados para resolver com eficiência equações diferenciais parciais. Estes métodos têm como principal objetivo acelerar a convergência do esquema iterativo em que são aplicados (Pletcher et al., 1997).

Os métodos *multigrid* estão entre as técnicas mais eficientes utilizadas na solução de equações elípticas (Briggs et al., 2000; Trottenberg et al., 2001) porque o número de operações aritméticas que precisam ser realizadas para atingir o nível do erro de discretização é proporcional ao número de incógnitas do sistema de equações a ser resolvido (McBryan, 1990). A relação entre o número de operações aritméticas e o número de incógnitas é uma das virtudes dos métodos *multigrid* e é conhecida como *textbook multigrid efficiency* (TME) (Thomas et al., 2003).

O interesse em resolver problemas de larga escala, mantendo o custo computacional (tempo de processamento) aceitável, conduz à necessidade de constantes melhorias nos algoritmos utilizados. Considerando a ampla disponibilidade de processadores nos sistemas atuais a paralelização do processamento computacional apresenta-se como recurso desejável.

1.2 Motivação

A discretização de equações que modelam certos fenômenos, como o escoamento de fluido em três dimensões, pode resultar em sistemas de equações com elevado número de incógnitas de forma que até mesmo os mais eficientes computadores seriais podem apresentar desempenho inadequado. Possível solução para estes casos, em que até mesmo o emprego do método *multigrid* pode ser insuficiente, é o uso de processamento em paralelo (McBryan, 1990).

Existem vários métodos utilizados para resolver numericamente o escoamento laminar bidimensional em regime transitório de fluido incompressível modelado pelas equações de Navier-Stokes. Bastante populares, os métodos de passo fracionário, frequentemente referidos na literatura como métodos de projeção, podem ser divididos em três classes (Guermond et al., 2006): esquemas de correção na pressão (*pressure-correction schemes*), esquemas de correção na velocidade (*velocity-correction schemes*) e esquemas de divisão consistente (*consistent splitting schemes*).

O primeiro e mais simples método considerado de projeção foi proposto por Chorin (1968) e Temam (1969). Na primeira etapa do algoritmo e utilizando-se o método de Euler na discretização temporal, o algoritmo consiste em criar uma variável auxiliar (também conhecida como velocidade temporária ou ainda preditiva) com a qual se estimam as velocidades no passo de tempo atual a partir do valor das velocidades no passo anterior ignorando a pressão. Na segunda etapa é calculada a pressão a partir da velocidade auxiliar. Nesta etapa, em que é realizada a projeção, são aplicadas as condições de incompressibilidade (no cálculo da pressão). Com o valor da pressão é realizada a correção da velocidade no tempo atual.

O fato de que em cada uma das etapas dos métodos de projeção serem resolvidas equações elípticas para a pressão e velocidades de forma desacoplada representa o aspecto mais atrativo destes métodos. No entanto, as etapas realizadas nos métodos de projeção são inherentemente sequenciais no tempo o que conduz à necessidade de paralelizar os métodos numéricos utilizados nas resoluções dos sistemas lineares resultantes, neste trabalho os métodos *multigrid*. Além da paralelização que pode ser empregada na solução dos sistemas lineares (utilizando o *multigrid*), propõe-se neste trabalho também a paralelização temporal dos métodos de projeção, que representa uma grande oportunidade de redução no tempo total de processamento.

1.3 Revisão bibliográfica

1.3.1 Métodos de projeção para resolução das equações de Navier-Stokes

Após os trabalhos de Chorin (1968) e Temam (1969), Goda (1979) observou que adicionando o valor do gradiente de pressão de uma iteração anterior à primeira etapa do algoritmo de Chorin (1968), que ignora a pressão completamente na primeira etapa, a acurácia do algoritmo aumenta. Esta observação foi utilizada por van Kan (1986) na formulação de um método de correção na pressão de segunda ordem. O método em questão é conhecido como esquema de correção incremental na pressão na forma padrão.

Em Bell et al. (1989) encontra-se um método de projeção bastante conhecido, similar ao de van Kan (1986), ou seja, consiste em um esquema de correção incremental na pressão na forma padrão. O esquema de Bell et al. (1989) tem sido bastante utilizado na literatura e pode ser encontrado em várias variantes, normalmente utilizadas para resolver problemas mais complexos, tais como escoamentos reativos, discutidos em Almgren et al. (1993, 1996) e escoamentos multifásicos, detalhados em Villar (2007).

Assim como no caso do método de Chorin (1968), os métodos de Goda (1979) e van Kan (1986) também possuem sua acurácia afetada devido a condições de contorno impostas à pressão. Na tentativa de superar esta dificuldade Timmermans et al. (1996) propuseram adicionar o divergente das velocidades auxiliares na segunda etapa do algoritmo de van Kan (1986). Esta pequena modificação conduz a melhorias na taxa de convergência da pressão, principalmente porque impõe sobre a pressão, condições de contorno consistentes. No algoritmo de Timmermans et al. (1996) as condições de contorno da pressão devem satisfazer o rotacional das velocidades nos contornos; devido a esta propriedade o método é chamado de correção incremental na pressão na forma rotacional.

Em Kim e Moin (1985) encontra-se um método de projeção que, como o método de Chorin (1968), ignora a pressão na primeira etapa do algoritmo e não realiza a correção da pressão na segunda etapa, no entanto apresenta taxas de convergência similares às obtidas pelo esquema de correção incremental da pressão na forma rotacional, ou seja, o esquema de Timmermans et al. (1996). Pelo fato de ignorar a pressão na primeira etapa e não realizar a correção da mesma na segunda etapa, Brown et al. (2001) referem-se ao esquema de Kim e Moin (1985) como *pressure-free projection method*. Guermond et al. (2006), através de mudanças de variáveis apropriadas, provam que o esquema de Kim e Moin (1985) é equivalente ao esquema de Timmermans et al. (1996), explicando assim as taxas de convergência observadas.

Em Orszag et al. (1986) e Karniadakis et al. (1991) é apresentada uma nova classe de métodos de projeção conhecidos como esquemas de correção na velocidade. Estes métodos, basicamente, invertem as etapas dos métodos com correção na pressão. De forma similar ao caso da correção na pressão, os métodos de correção na velocidade também possuem uma versão com correção incremental na forma padrão e, introduzida por Guermond e Shen (2003), na forma rotacional.

Guermond et al. (2006) fazem uma ampla revisão dos métodos de projeção disponíveis na literatura. Apresentam os principais métodos de projeção, as taxas de convergência e a relação destes com outros métodos de projeção.

Kadioglu et al. (2008) apresentaram um método numérico de quarta ordem para escoamentos compressíveis para dinâmica de gases com número de Mach tendendo a zero. O método de projeção proposto consiste na extensão do caso de escoamentos incompressíveis para o caso de escoamentos compressíveis (Oseledets, 1989). A discretização espacial de quarta ordem é feita utilizando o método dos volumes finitos e a discretização temporal é baseada no método das correções espectrais adiadas.

Guermond e Salgado (2009) apresentaram um novo método de separação consistente de segunda ordem para resolução das equações de Navier-Stokes com densidade variável. O método tem custo computacional reduzido ao resolver apenas uma equação por passo de tempo. A estabilidade do método é provada analiticamente e o desempenho do método é testado.

Griffith (2009) resolveu as equações de Navier-Stokes para escoamentos incompressíveis utilizando uma metodologia de segunda ordem empregando malhas desencontradas. A metodologia consiste no emprego de um método de projeção como precondicionador do sistema linear resultante, este resolvido com o método dos gradientes conjugados precondicionado com o método *multigrid* geométrico. Validou a metodologia proposta para vários problemas incluindo

o problema da cavidade com tampa deslizante (Ghia et al., 1982). Apresentou resultados para números de Reynolds variando de 1 a 30000.

Astorino e Grandmont (2010) apresentaram a análise de convergência de um método de projeção com discretização espacial pelo método dos elementos finitos e discretização temporal semi-implícita. Apresentaram analiticamente uma estimativa do erro associado à discretização temporal e apresentaram resultados numéricos para corroborar a solução analítica.

Almgren et al. (2013) resolveram as equações de Navier-Stokes utilizando um método de quarta ordem tanto na discretização espacial quanto temporal. Similar ao trabalho de Kadioglu et al. (2008), o método de projeção proposto consiste na extensão do caso de escoamentos incompressíveis para o caso de escoamentos compressíveis (Oseledets, 1989). Utilizou o método dos volumes finitos na discretização espacial e o método das correções espectrais adiadas na discretização temporal. O trabalho avaliou a relação entre o método apresentado e a resolução de malha necessária para adequadamente resolver problemas de escoamento turbulento. Compararam o método de quarta ordem apresentado com versões de segunda ordem e demonstraram que para determinado número de Reynolds a resolução de malha necessária para resolver adequadamente escoamentos turbulentos é muito menor que a necessária para métodos de segunda ordem. Além disso, mostram que para dado Reynolds é possível prever a resolução de malha necessária.

Zhang (2014) resolveu as equações de Navier-Stokes para escoamentos incompressíveis utilizando um método de projeção de quarta ordem de acurácia tanto na discretização espacial quanto temporal. Empregou volumes finitos para a discretização espacial e o método de Runge-Kutta semi-implícito para a discretização temporal. Utiliza malhas estruturadas com refinamento adaptativo. A condição de divergente nulo é satisfeita empregando um operador de projeção aproximado em que o autor apresentou detalhada análise de estabilidade e acurácia.

1.3.2 Paralelização do método *multigrid*

Visto que as etapas realizadas nos métodos de projeção são inherentemente sequenciais no tempo é bastante comum paralelizar os métodos numéricos utilizados nas resoluções dos sistemas lineares resultantes, como neste trabalho são utilizados os métodos *multigrid* apresentam-se a seguir trabalhos que abordam a paralelização do mesmo.

Vários autores (Thole, 1985; Chan e Schreiber, 1985; Chan e Saad, 1986; Hempel e Schüller, 1988; Herbin et al., 1988; McBryan, 1990) estudaram a paralelização dos métodos *multigrid*.

Contudo, segundo Trottenberg et al. (2001), o *multigrid* padrão não é completamente paralelizável, pois as malhas são processadas sequencialmente e o grau de paralelismo é diferente para cada malha considerada (menor para malhas mais grossas).

Visto que o custo de comunicação entre os processadores não é desprezível, várias versões do método *multigrid* paralelo foram criadas. Uma classe de algoritmos *multigrid* que processam vários (ou todos) níveis simultaneamente, ou seja, as malhas não são processadas sequencialmente (conhecida como versão aditiva), foi estudada por autores como Gannon e Rosendale (1986), Bramble et al. (1990), Greenbaum (1986) e Xu (1992).

Autores como Frederickson e McBryan (1987b,a, 1990) desenvolveram o algoritmo Método *Multigrid* Paralelo Superconvergente (*Parallel Superconvergent Multigrid Method*) que trata de paralelismo massivo (emprego de uma grande quantidade de processadores em paralelo) e consiste em resolver várias malhas grossas simultaneamente com objetivo de acelerar a convergência do método *multigrid*.

Na tentativa de evitar o custo de comunicação crescente nas malhas mais grossas, Hempel e Schüller (1988) sugeriram uma técnica de aglomeração: nas malhas mais grossas podem-se aglomerar vários pontos de malha em um único processador para evitar comunicação em excesso.

A Decomposição do Domínio (DD) foi estudada por autores como Quarteroni e Valli (1999) e Smith et al. (1996). Esta técnica consiste em decompor o problema em vários outros problemas com domínios próprios, com ou sem a sobreposição dos mesmos.

Com a popularização dos sistemas com memória compartilhada surge o interesse em estudar o desempenho do método *multigrid* neste tipo de máquina.

Baker et al. (2011) observam que o método *multigrid* algébrico (AMG) apresenta bom desempenho em sistemas com memória distribuída, no entanto, enfrenta desafios que podem prejudicar sua eficiência quando executados em arquiteturas com memória compartilhada: sistema com vários processadores, processadores com vários núcleos, acesso não uniforme à memória, compartilhamento de memória *cache*, que é um tipo de memória presente dentro dos processadores com o propósito de diminuir a diferença com que os processadores executam instruções e a velocidade com que os dados são acessados na memória principal. Em seu trabalho resolvem um problema representado por Laplace 3D discretizado com diferenças finitas em um cubo com 1000^3 pontos de malha. Foca no estudo da utilização das APIs (*Application Programming Interface*) OpenMP e MPI, além de uma aplicação híbrida destas. Apresentam melhorias no uso da API OpenMP em sistemas com acesso não uniforme à memória. O sistema testado é o Hera: um *cluster* com 864 nós sendo estes equipados com 4 AMD Quadcore 8356 rodando a uma velocidade de 2,3 GHz, ou seja, um sistema com memória distribuída cujos nós são compostos por sistemas com memória compartilhada.

Courtecuisse e Allard (2009) estudam o desempenho do método Gauss-Seidel utilizando várias estratégias com objetivo de expor o paralelismo (processos que podem ser executados em paralelo). As estratégias testadas consistem em dividir a matriz do sistema de equações em linhas, colunas e blocos de forma que cada um destes possa ser processado em paralelo. Para os testes empregam sistemas com memória compartilhada com 2, 4, 8 e 16 núcleos, além disso, testam o desempenho das técnicas discutidas para o caso de um GPU (NVIDIA GeForce GTX 280) com 240 núcleos. Demonstram que a implementação do Gauss-Seidel utilizando as técnicas propostas é capaz de utilizar todos os núcleos disponíveis.

Wallin et al. (2006) resolvem um problema representado por Poisson 3D utilizando os métodos Gauss-Seidel *red-black*, *multigrid* e, proposta no trabalho, *parallel temporally blocked* Gauss-Seidel (TBGS). O sistema utilizado é o UltraSPARC IV com memória compartilhada e um total de 8 núcleos. Para os testes são empregadas malhas com 129^3 , 257^3 e 513^3 . O número de suavizações utilizado no método *multigrid* varia de 1 a 7, mas iguais na pré e pós-suavização. Demonstram que a versão do método Gauss-Seidel proposta no trabalho (TBGS) conduz a melhores resultados que a versão *red-black* convencional.

Em Kowarschik et al. (2000) é resolvido um problema representado por Poisson 2D utilizando o método Gauss-Seidel *red-black*. O trabalho apresenta possíveis modificações do método Gauss-Seidel *red-black* de forma que este faça uso intensivo da memória *cache*. Discute a técnica *loop fusion* na qual se faz uma única varredura do domínio atualizando nós *black* e *red* alternativamente, ou seja, reduzindo pela metade (em relação ao *red-black* convencional) o número de vezes que as informações das malhas são escritas no *cache*. Discute a técnica *loop blocking* (ou *loop tiling*), na qual o domínio é dividido em blocos com tamanho apropriado à quantidade de memória *cache* disponível nos processadores; nesta técnica os blocos devem ser reutilizados o máximo possível para se beneficiar da velocidade superior da memória *cache*, no entanto é prejudicada pela dependência local (nós vizinhos) dos dados. Introduz a técnica

Windshield Wiper, que consiste em utilizar um pequeno (apropriado ao nível da memória *cache* que se deseja otimizar) bloco bidimensional que atualiza o domínio fazendo uso intensivo da informação armazenada na memória *cache*, além disso, beneficia-se dos diferentes níveis (L1, L2, L3, registradores, etc) que compõem esta. Demonstra que a otimização do uso da memória *cache* conduz a bons resultados.

McAdams et al. (2010) resolvem um problema modelado por Poisson 3D com condições de Dirichlet e Neumann. Utilizam o método *multigrid* como precondicionador da aplicação de um método de Krylov. O tipo de ciclo utilizado é o V e o suavizador escolhido é o Jacobi ponderado com fator de ponderação $\omega = 2/3$. Os testes são realizados em um servidor Intel Xeon X7350 e são obtidos bons resultados.

Apesar da paralelização que pode ser empregada na solução dos sistemas lineares utilizando o método *multigrid*, neste trabalho optou-se pela modificação dos métodos de projeção de forma que estes permitam paralelização temporal, principalmente porque que estes métodos são inherentemente sequenciais no tempo e a paralelização proposta representa uma grande oportunidade de redução no tempo total de processamento. Além disso deve-se observar a escassez de trabalhos com estudos semelhantes à proposta deste trabalho.

1.4 Objetivos

O objetivo geral deste trabalho é modificar os métodos de projeção sequenciais no tempo (Guermond et al., 2006) para que permitam processamento temporal em paralelo. Na resolução dos sistemas lineares será utilizado o método *multigrid*.

Os objetivos específicos são:

1. Unificar os campos de velocidades temporária (ou auxiliar) e real, onde a velocidade temporária é a velocidade utilizada para calcular a pressão e velocidade do fluido, sendo esta última a velocidade “real”.
2. Modificar os métodos de projeção para que cada passo de tempo possa ser processado em paralelo.
3. Modificar os métodos de projeção para obter aproximações das componentes da velocidades a cada passo de tempo.
4. Modificar os métodos de projeção para obter aproximações da correção da pressão a cada passo de tempo.
5. Obter as aproximações das velocidades com o emprego do método *multigrid* para todos os passos de tempo simultaneamente.
6. Obter as aproximações da correção da pressão com o emprego do método *multigrid* para cada passo de tempo de forma individual.
7. Paralelizar a atualização da pressão utilizando o particionamento do domínio.

2 Fundamentação

Neste capítulo são apresentados os fundamentos básicos do método *multigrid* e discutidos os operadores de transferência entre malhas (restrição e prolongação) específicos para cada variável (velocidades e pressão). Além disso são discutidos a ordenação lexicográfica adotada no trabalho e conceitos básicos de paralelização.

2.1 Método *multigrid*

2.1.1 Considerações sobre o método *multigrid*

O método *multigrid* consiste, basicamente, em explorar a propriedade de suavização (redução de componentes de erro oscilatório) dos métodos iterativos básicos. Como exemplos de métodos iterativos básicos podem-se citar os métodos de Jacobi e Gauss-Seidel (Briggs et al., 2000; Trottenberg et al., 2001). As etapas do método *multigrid* podem ser resumidas no esquema de correção para duas malhas descrito a seguir:

1. Aplicação do método iterativo em uma dada malha até que as componentes do erro tornem-se suaves. Neste passo a malha será chamada de malha “fina” e será denotada por Ω^h , ou seja, uma malha com volumes de tamanho h , com $h = h_x = h_y$.
2. Transferência das informações da malha fina para uma malha mais grossa (com menos pontos que a original). Neste passo a malha será chamada de malha “grossa” e será denotada por Ω^H , ou seja, uma malha com volumes de tamanho H , com $H > h$. Nesta malha os erros suavizados no passo anterior tornam-se mais oscilatórios e, portanto, a aplicação dos métodos iterativos é muito mais vantajosa que sua aplicação na malha fina.
3. Transferência das informações da malha mais grossa para a malha mais fina. A solução da malha fina é melhorada com o erro suavizado da malha grossa.

O esquema de correção para duas malhas apresentado pode ser facilmente estendido para diversas malhas, ou seja, pode ser aplicado de Ω^h para Ω^{2h} , de Ω^{2h} para Ω^{4h} , etc. O procedimento por ser repetido até que a malha mais grossa possível, ou outra malha conveniente, seja atingida.

Considerando malhas uniformes (volumes da malha com mesmo tamanho), a razão de engrossamento é definida como $r = H/h$, onde h representa o tamanho dos volumes da malha fina Ω^h e H o tamanho dos volumes da malha grossa Ω^H . Neste trabalho será utilizado $r = 2$ (Briggs et al., 2000; Trottenberg et al., 2001).

O número de malhas (também chamado de níveis) $1 \leq L \leq L_{\max}$ que podem ser visitadas durante o processo de engrossamento depende do problema considerado e da razão

de engrossamento utilizada. A malha mais grossa possível é a malha com 2×2 volumes. A quantidade de níveis até atingir essa malha, chamada número máximo de níveis, será aqui denotado por L_{\max} . Não é obrigatório engrossar as malhas até o nível máximo, mas Pinto e Marchi (2007) e Oliveira et al. (2012) recomendam utilizar todos os níveis.

A Fig. 2.1 mostra o processo de engrossamento de uma malha de 32×32 volumes de tamanho h (em cada direção coordenada) e razão de engrossamento $r = 2$, ou seja, $H = 2h$. Neste caso o processo de engrossamento inicia na malha de 32×32 volumes (o primeiro nível) e termina no quinto nível, ou seja, na malha mais grossa possível (2×2 volumes), totalizando $L = L_{\max} = 5$ níveis. A seta indica a direção de engrossamento da malha.

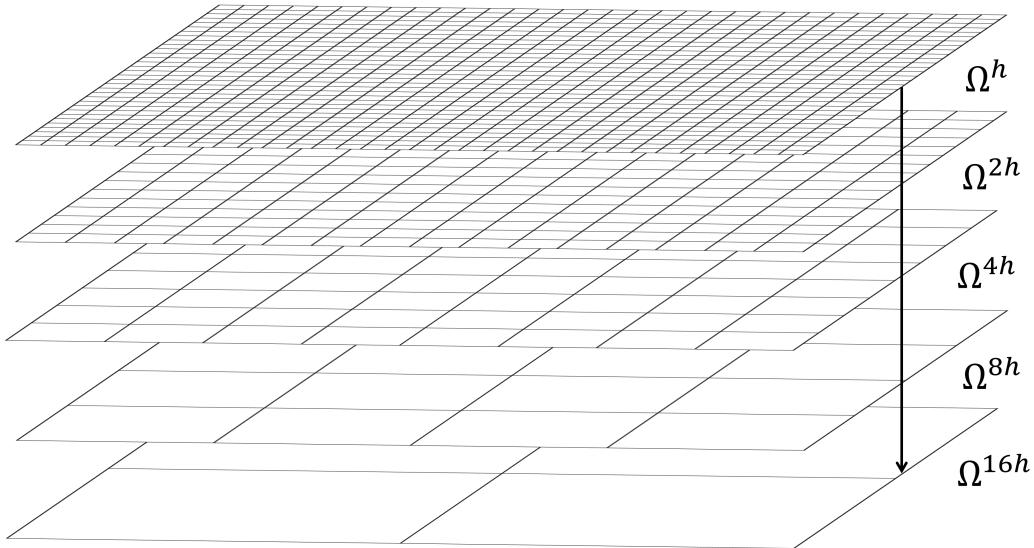


Figura 2.1: Processo de engrossamento de uma malha uniforme e razão de engrossamento $r = 2$.

A ordem na qual as malhas são visitadas é chamada de ciclo *multigrid*. Existem vários tipos de ciclos *multigrid* tais como o ciclo V, o ciclo W, o ciclo dente-de-serra, entre outros (Briggs et al., 2000; Trottenberg et al., 2001; Wesseling, 2004). Neste trabalho será utilizado exclusivamente o ciclo V, pois o ciclo W é cerca de 50% mais caro em relação ao número de operações envolvidas (Hirsch, 1991). A Fig. 2.2 apresenta um exemplo de ciclo V para $L = 5$.

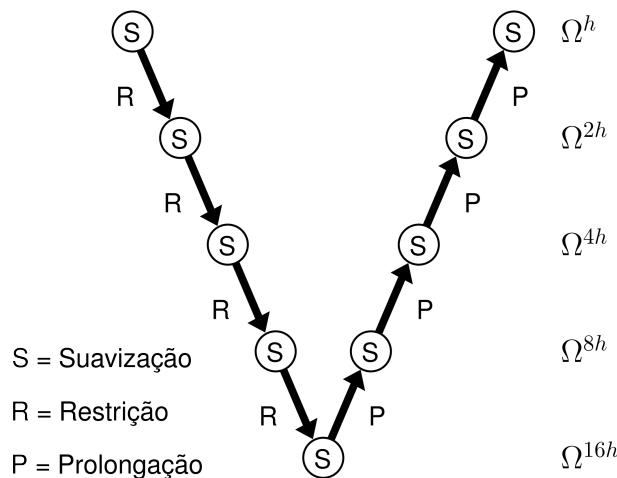


Figura 2.2: Diagrama do ciclo V para $L = 5$.

Não existe restrição ao valor da razão de engrossamento utilizado, que pode ser $r = 2$, $r = 3$, $r = 4$ entre outros valores, mas Brandt (1977) recomenda utilizar $r = 2$ por ser o valor mais próximo da razão de engrossamento ótima e Briggs et al. (2000) observa que utilizar $r \neq 2$ sem considerar as características do problema, não traz vantagens.

Tendo-se estabelecido o papel da malha grossa torna-se necessário definir um procedimento para transferir informações da malha fina para esta malha (processo de restrição) e desta para a malha fina (processo de prolongação). Estes processos de transferência são os temas das seções 2.1.3 e 2.1.4.

2.1.2 Ordenação lexicográfica

A implementação (em FORTRAN) dos conceitos desenvolvidos neste trabalho segue as recomendações de Briggs et al. (2000), ou seja, o vetor solução de cada nível de malha ($\mathbf{u}^h, \mathbf{u}^{2h}, \dots, \mathbf{u}^{Lh}$), onde foi assumido que $Lh = 2^{l-1}h$ para algum $l > 1$, e o vetor de termos independentes ($\mathbf{b}^h, \mathbf{b}^{2h}, \dots, \mathbf{b}^{Lh}$) são armazenados na memória em um único vetor (*array 1D*) de forma adjacente, como ilustrado na Fig. 2.3.

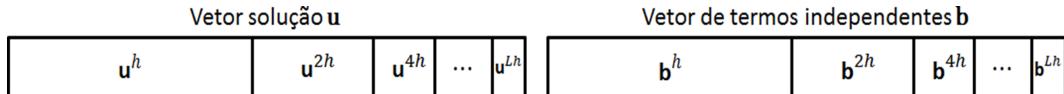


Figura 2.3: Armazenamento adjacente na memória.

A escolha pela ordenação lexicográfica dos volumes do domínio deve-se à forma de armazenamento ilustrada na Fig. 2.3.

A relação entre as coordenadas lexicográficas das variáveis pressão, pp , componentes u (horizontal), pu , e v (vertical), pv , da velocidade; e (i, j) são dadas por

$$\begin{aligned} pp &= i + (j - 1)Nxp \\ pv &= i + (j - 1)N xv, \\ pu &= i + (j - 1)N xu \end{aligned} \quad (2.1)$$

onde Nxp , Nxu e $N xv$ representam o número de volumes, para a pressão, e número de faces, para as componentes da velocidade, de cada variável na direção x .

Na Fig. 2.4(a) é apresentado o recorte de uma malha com as posições (i, j) da pressão P (círculo cheio), da velocidade v (quadrado cheio) e velocidade u (quadrado vazio). Deve ficar claro que cada variável (pressão e velocidades) possui seu próprio índice (i, j) , cujas posições são mostradas na Fig. 2.4(a). Na Fig. 2.4(b) é apresentado o recorte de uma malha com os índices (i, j) , $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ e $(i, j + 1)$ da variável pressão. De forma similar ao caso da Fig. 2.4(b), nas figuras 2.4(c) e 2.4(d) são mostrados os índices das velocidades u e v , respectivamente.

Na Fig. 2.5 são apresentadas as posições de todas as variáveis; da pressão P (círculo cheio), da velocidade v (quadrado cheio) e velocidade u (quadrado vazio); na ordem lexicográfica calculados com a Eq. (2.1).

2.1.3 Processo de restrição

Os operadores que transferem informações da malha fina Ω^h para a malha grossa Ω^{2h} são chamados de operadores de restrição e são denotados por I_h^{2h} . Estes operadores transformam vetores da malha fina \mathbf{v}^h em vetores da malha grossa \mathbf{v}^{2h} de acordo com a regra $I_h^{2h}\mathbf{v}^h = \mathbf{v}^{2h}$.

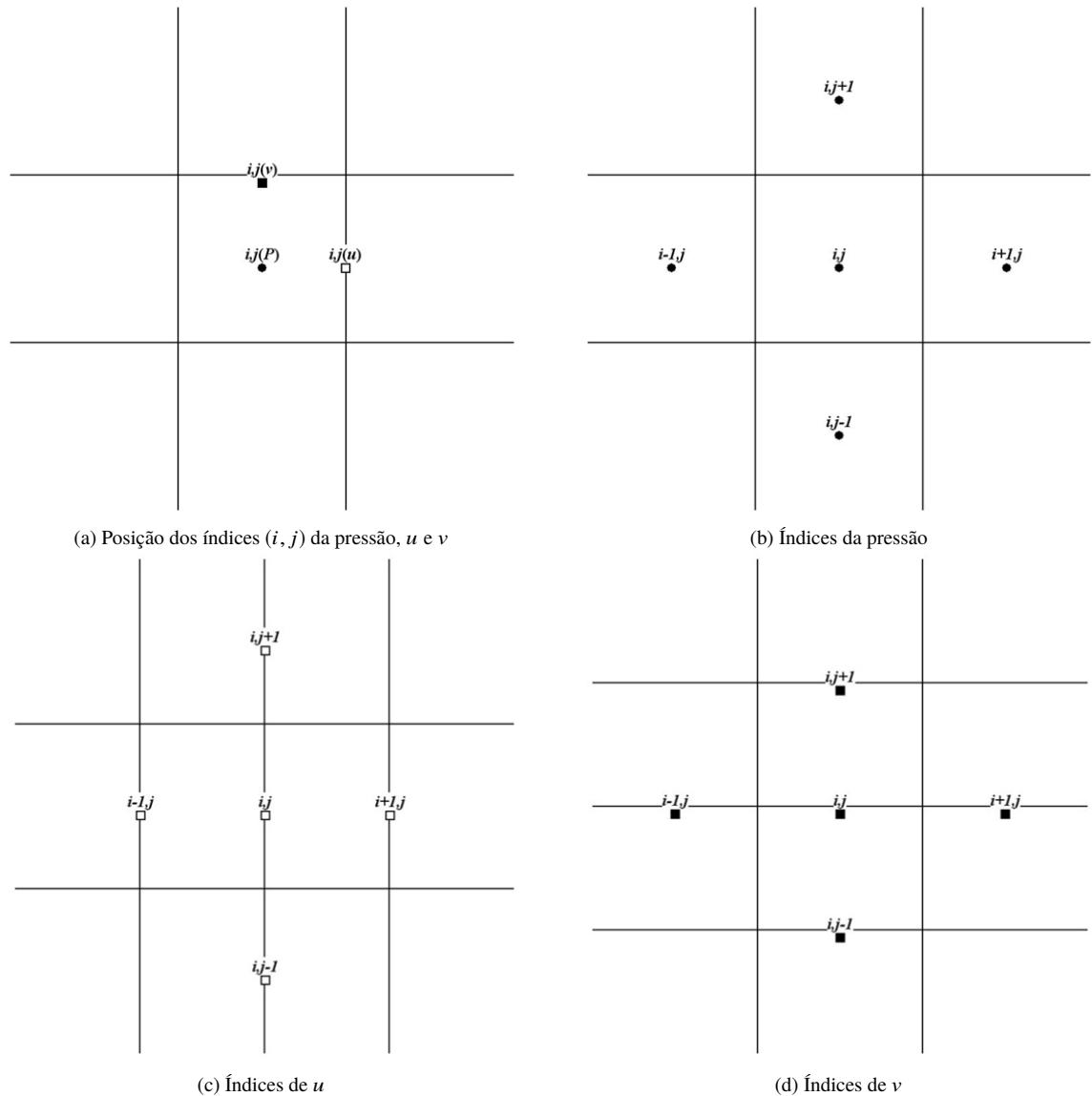


Figura 2.4: Nomenclatura dos vizinhos genéricos dos volumes da malha bidimensional uniforme.

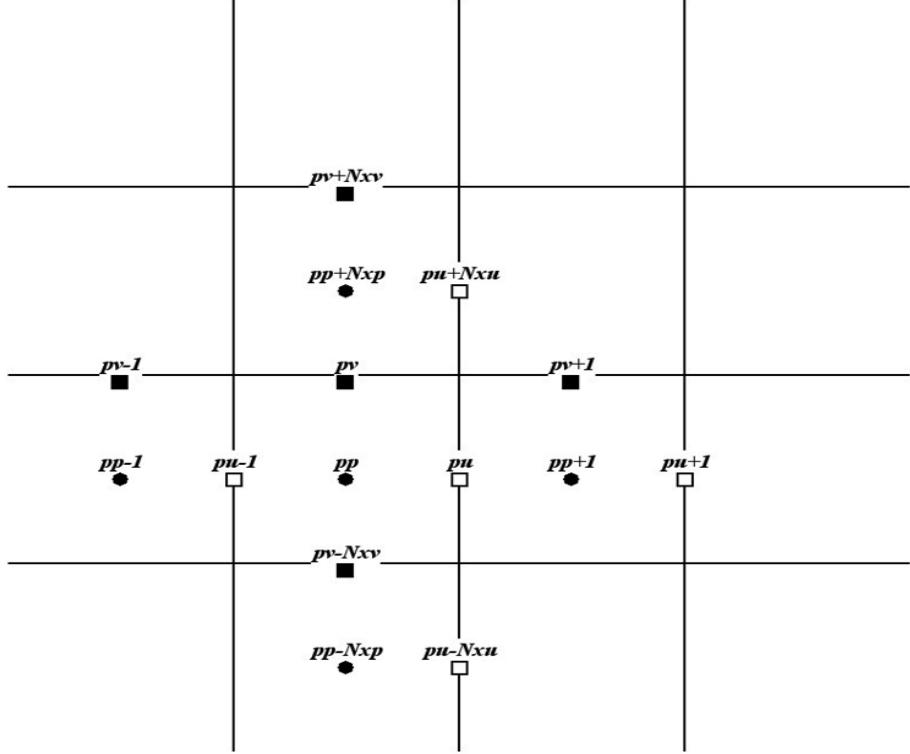


Figura 2.5: Posições da pressão P e velocidades u e v na ordem lexicográfica.

Um operador muito conhecido e utilizado é o de restrição por ponderação completa (Trottenberg et al., 2001), que para o caso da pressão é dado por

$$\phi_{pp}^{2h} = \frac{(\phi_{pp}^h + \phi_{pp+1}^h + \phi_{pp+Nxp}^h + \phi_{pp+Nxp+1}^h)}{4}, \quad (2.2)$$

onde ϕ_{pp}^{2h} é o valor da pressão na malha grossa Ω^{2h} na posição pp (pp da malha grossa). De forma similar ϕ_{pp}^h é o valor da pressão na malha fina na posição pp (pp da malha fina). A Fig. 2.6 ilustra como o processo é realizado.

Para o caso das velocidades u e v o procedimento é semelhante ao utilizado por Trottenberg et al. (2001) apresentado na Eq. (2.3)

$$\begin{aligned} u_{pu}^{2h} &= \frac{(u_{pu}^h + u_{pu+Nxu}^h)}{2} \\ v_{pv}^{2h} &= \frac{(v_{pv}^h + v_{pv+1}^h)}{2} \end{aligned} \quad (2.3)$$

A Fig. 2.7 ilustra o processo de restrição de variáveis deslocadas apresentado na Eq. (2.3). Nela é mostrado como a informação contida nas faces dos volumes da malha fina Ω^h é transferida para a malha grossa Ω^{2h} .

2.1.4 Processo de prolongação

Os operadores que transferem informações da malha grossa Ω^{2h} para a malha fina Ω^h são chamados de operadores de prolongação, também conhecidos como interpolação, e são denotados por I_{2h}^h . Estes operadores transformam vetores da malha grossa \mathbf{v}^{2h} em vetores da malha fina \mathbf{v}^h de acordo com a regra $I_{2h}^h \mathbf{v}^{2h} = \mathbf{v}^h$.

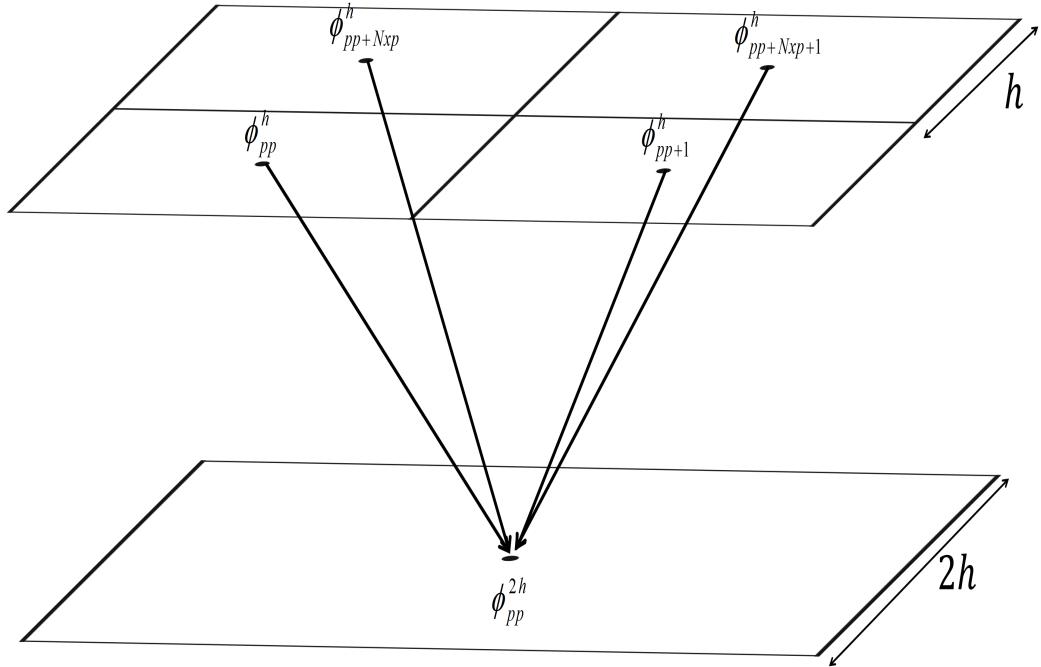
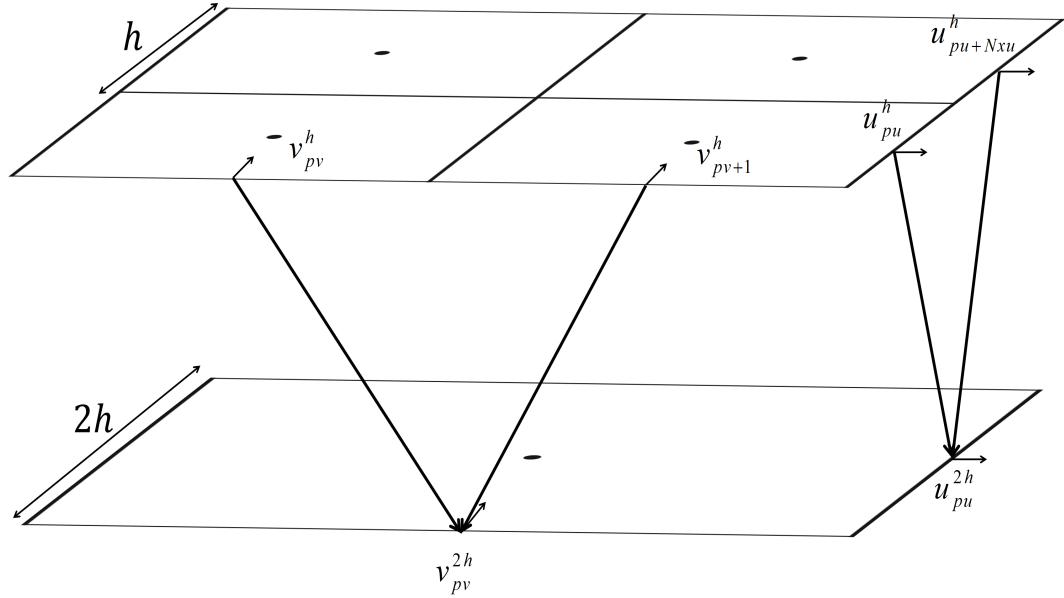


Figura 2.6: Processo de restrição para variável centrada (pressão).

Figura 2.7: Processo de restrição para variáveis deslocadas (u e v).

Entre os operadores de prolongação conhecidos, um dos mais utilizados é o operador de interpolação bilinear. Para o caso da pressão o procedimento é similar ao utilizado por Villar (2007) e seguindo a nomenclatura convencionada para ordenação lexicográfica, tem-se

$$\left\{ \begin{array}{l} \phi_{pp}^h = \frac{9\phi_{pp}^{2h} + 3\phi_{pp-1}^{2h} + 3\phi_{pp-Nxp}^{2h} + \phi_{pp-Nxp-1}^{2h}}{16} \\ \phi_{pp+1}^h = \frac{9\phi_{pp}^{2h} + 3\phi_{pp+1}^{2h} + 3\phi_{pp-Nxp}^{2h} + \phi_{pp-Nxp+1}^{2h}}{16} \\ \phi_{pp+Nxp}^h = \frac{9\phi_{pp}^{2h} + 3\phi_{pp-1}^{2h} + 3\phi_{pp+Nxp}^{2h} + \phi_{pp+Nxp-1}^{2h}}{16} \\ \phi_{pp+Nxp+1}^h = \frac{9\phi_{pp}^{2h} + 3\phi_{pp+1}^{2h} + 3\phi_{pp+Nxp}^{2h} + \phi_{pp+Nxp+1}^{2h}}{16} \end{array} \right. . \quad (2.4)$$

A Fig. 2.8 ilustra o processo de prolongação da pressão apresentado na Eq. (2.4) (mais especificamente a última equação). Nela é mostrado como a informação dos volumes da malha grossa Ω^{2h} é transferida para os volumes da malha fina Ω^h .

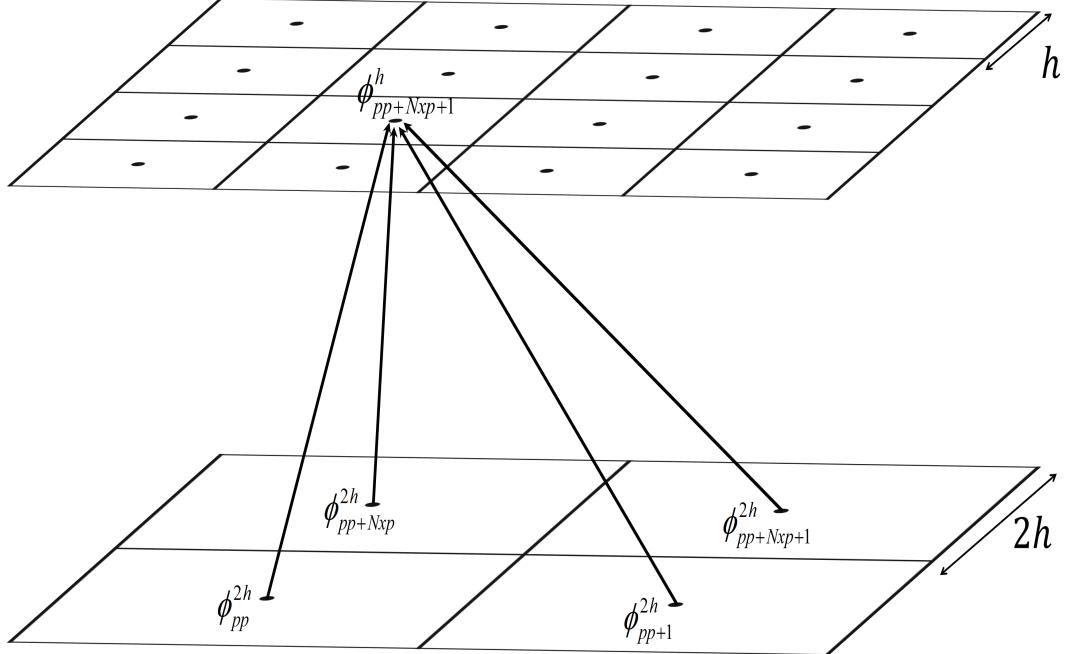


Figura 2.8: Processo de prolongação para a variável pressão.

Para o caso da variável u as equações de interpolação seguem o procedimento utilizado por Villar (2007) e utilizando a nomenclatura convencionada para a ordenação lexicográfica, tem-se:

$$\left\{ \begin{array}{l} u_{pu}^h = \frac{3u_{pu}^{2h} + u_{pu-Nxu}^{2h}}{4} \\ u_{pu-1}^h = \frac{3u_{pu}^{2h} + 3u_{pu-1}^{2h} + u_{pu-Nxu}^{2h} + u_{pu-Nxu-1}^{2h}}{8} \\ u_{pu+1}^h = \frac{3u_{pu}^{2h} + 3u_{pu+1}^{2h} + u_{pu-Nxu}^{2h} + u_{pu-Nxu+1}^{2h}}{8} \\ u_{pu+Nxu}^h = \frac{3u_{pu}^{2h} + u_{pu+Nxu}^{2h}}{4} \\ u_{pu+Nxu-1}^h = \frac{3u_{pu}^{2h} + 3u_{pu-1}^{2h} + u_{pu+Nxu}^{2h} + u_{pu+Nxu-1}^{2h}}{8} \\ u_{pu+Nxu+1}^h = \frac{3u_{pu}^{2h} + 3u_{pu+1}^{2h} + u_{pu+Nxu}^{2h} + u_{pu+Nxu+1}^{2h}}{8} \end{array} \right. . \quad (2.5)$$

A Fig. 2.9 ilustra o processo de prolongação da variável u apresentado na Eq. (2.5) (mais especificamente as três últimas equações). Nela é mostrado como a informação das faces da malha grossa Ω^{2h} é transferida para as faces da malha fina Ω^h .

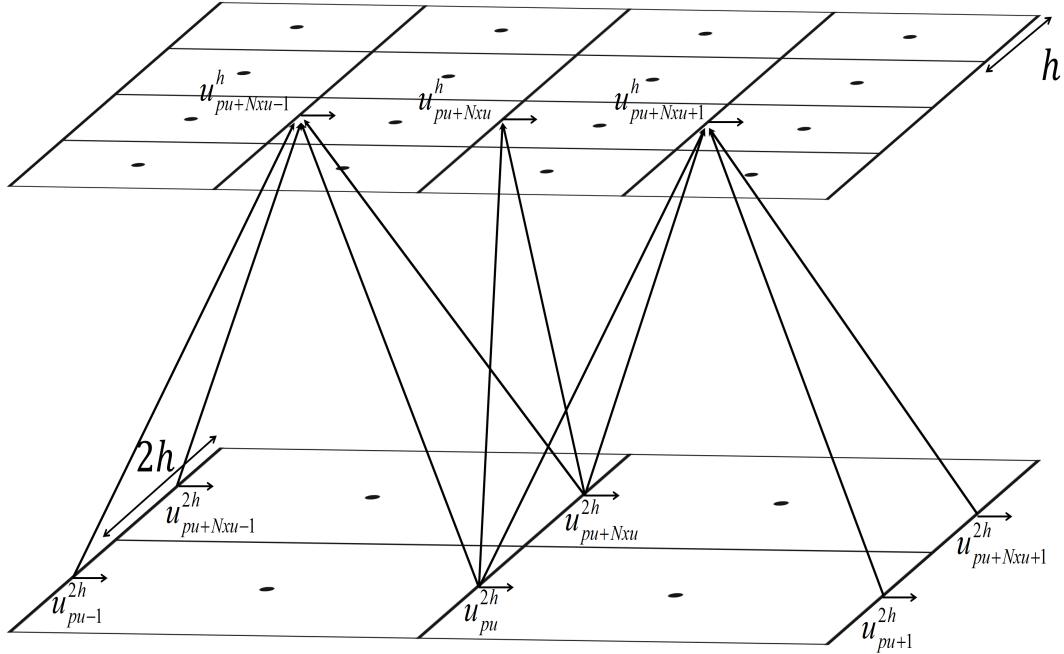


Figura 2.9: Processo de prolongação para variável deslocada u .

De forma similar, para a variável v as equações de interpolação podem ser escritas, seguindo Villar (2007) e utilizando a nomenclatura convencionada para a ordenação lixicográfica, tem-se

$$\left\{ \begin{array}{l} v_{pv}^h = \frac{3v_{pv}^{2h} + v_{pv-1}^{2h}}{4} \\ v_{pv+Nxv}^h = \frac{3v_{pv}^{2h} + 3v_{pv+Nxv}^{2h} + v_{pv-1}^{2h} + v_{pv+Nxv-1}^{2h}}{8} \\ v_{pv-Nxv}^h = \frac{3v_{pv}^{2h} + 3v_{pv-Nxv}^{2h} + v_{pv-1}^{2h} + v_{pv-Nxv-1}^{2h}}{8} \\ v_{pv+1}^h = \frac{3v_{pv}^{2h} + v_{pv+1}^{2h}}{4} \\ v_{pv-Nxv+1}^h = \frac{3v_{pv}^{2h} + 3v_{pv-Nxv}^{2h} + v_{pv+1}^{2h} + v_{pv-Nxv+1}^{2h}}{8} \\ v_{pv+Nxv+1}^h = \frac{3v_{pv}^{2h} + 3v_{pv+Nxv}^{2h} + v_{pv+1}^{2h} + v_{pv+Nxv+1}^{2h}}{8} \end{array} \right. . \quad (2.6)$$

A Fig. 2.10 ilustra o processo de prolongação da variável v apresentado na Eq. (2.6) (mais especificamente as três últimas equações). Nela é mostrado como a informação das faces da malha grossa Ω^{2h} é transferida para as faces da malha fina Ω^h .

2.1.5 Esquema de correção para duas malhas

Como será discutido em detalhes no capítulo 4 os métodos de projeção utilizados neste trabalho consistem em calcular uma velocidade auxiliar (também conhecida como temporária ou preditiva). Uma consequência vantajosa desta abordagem é que os termos não-lineares das equações de Navier-Stokes passam a fazer parte do termo fonte dos sistemas lineares obtidos, desta forma pode-se utilizar o esquema de correção (*Correction Scheme*, CS) de duas malhas que consiste em um processo de “correção residual” ou “refinamento iterativo” exemplificado nos passos a seguir:

1. Resolver $A\mathbf{u} = \mathbf{b}$ (com um método iterativo) e obter \mathbf{v} ;

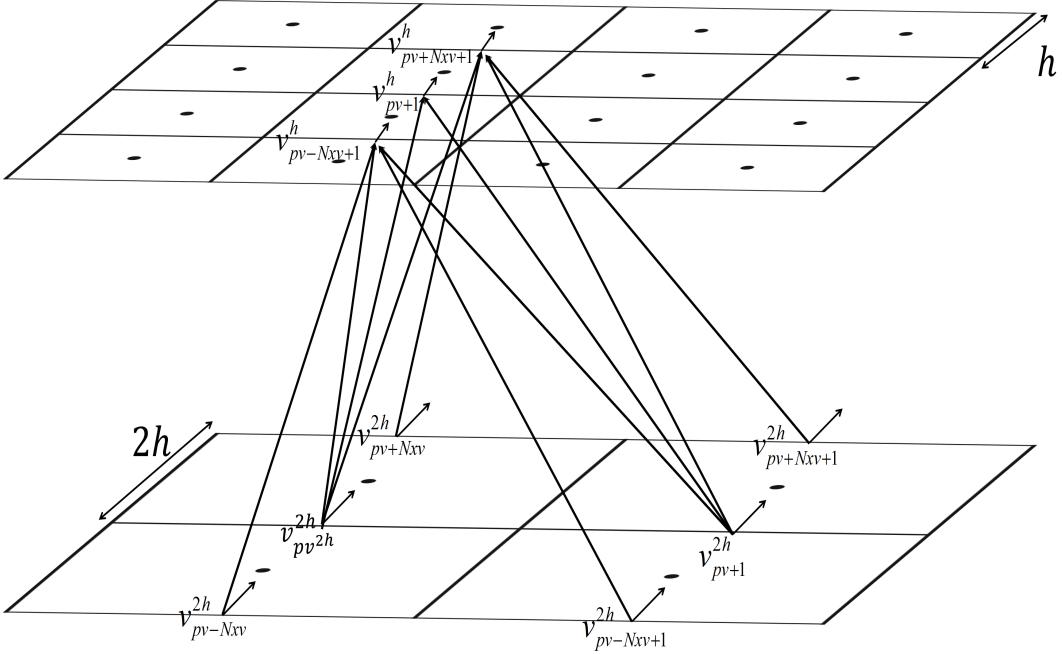


Figura 2.10: Processo de prolongação para variável deslocada v .

2. Calcular o resíduo com $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{v}$;
3. Resolver $A\mathbf{e} = \mathbf{r}$ (equação residual) e obter \mathbf{e} ;
4. Usando a definição de erro, corrigir \mathbf{u} com $\mathbf{u} = \mathbf{v} + \mathbf{e}$.

No método *multigrid*, o passo 3) da “correção residual” é realizado na malha grossa, ou seja, o resíduo é transferido para a malha grossa, resolve-se a equação residual, transfere-se o valor do erro para a malha fina e corrigir-se a solução anterior.

Utilizando os conceitos desenvolvidos até agora pode-se formalizar a discussão anterior no algoritmo a seguir

Algoritmo 1 Esquema de correção para duas malhas (adaptado de Briggs et al. (2000)).

1. Iterar $A^h \mathbf{u}^h = \mathbf{b}^h$ ν_1 vezes em Ω^h com estimativa inicial \mathbf{v}^h ;
 2. Calcular $\mathbf{r}^h = \mathbf{b}^h - A^h \mathbf{v}^h$;
 3. Restringir o resíduo da malha fina Ω^h para a malha grossa Ω^{2h} : $I_h^{2h} \mathbf{r}^h = \mathbf{r}^{2h}$;
 - (a) Iterar $A^{2h} \mathbf{e}^{2h} = \mathbf{r}^{2h}$ ν_1 vezes em Ω^{2h} com estimativa inicial $\mathbf{e}^{2h} = \mathbf{0}$;
 - (b) Interpolar \mathbf{e}^{2h} da malha grossa Ω^{2h} para a malha fina Ω^h : $I_{2h}^h \mathbf{e}^{2h} = \mathbf{e}^h$;
 4. Corrigir $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h$;
 5. Iterar $A^h \mathbf{u}^h = \mathbf{b}^h$ ν_2 vezes em Ω^h com estimativa inicial \mathbf{v}^h .
-

Assim como a matriz A^h foi obtida a partir da discretização do problema, a matriz A^{2h} pode ser obtida pela rediscritização do problema na malha grossa Ω^{2h} . Os parâmetros ν_1 e ν_2

são conhecidos como pré e pós-suavização, respectivamente. Estes parâmetros são importantes porque impactam no desempenho do método *multigrid*. Estudos dos valores ótimos destes parâmetros podem ser consultados em Oliveira et al. (2012) e Pinto e Marchi (2007) para a equação de Laplace, Oliveira et al. (2012) para a equação de Poisson.

Aplicando o Algoritmo 1 sucessivamente a cada duas malhas e seguindo a ordem estabelecida na Fig. 2.2 (ciclo V), pode-se formalizar o algoritmo *multigrid* empregando ciclo V no Algoritmo 2.

Para facilitar este procedimento foram feitas as seguintes modificações na notação do Algoritmo 1: na equação residual o vetor \mathbf{r}^{2h} , foi substituído por \mathbf{b}^{2h} , e o vetor \mathbf{e}^{2h} , por \mathbf{u}^{2h} . Desta forma os vetores do lado esquerdo ($\mathbf{u}^h, \mathbf{u}^{2h}, \dots, \mathbf{u}^{Lh}$) sempre representam o vetor solução de um sistema de equações $A\mathbf{u} = \mathbf{b}$. De forma semelhante, os vetores do lado direito ($\mathbf{b}^h, \mathbf{b}^{2h}, \dots, \mathbf{b}^{Lh}$) sempre representam o vetor dos termos independentes. Nesta notação as equações residuais são interpretadas como um sistema de equações $A\mathbf{u} = \mathbf{b}$, onde o resíduo assume o papel do vetor de termos independentes. O Algoritmo 2, que utiliza a notação discutida, é uma adaptação do esquema de correção para duas malhas para o caso de várias malhas.

Algoritmo 2 Ciclo V com esquema CS para várias malhas (adaptado de Briggs et al. (2000)).

1. Iterar $A^h\mathbf{u}^h = \mathbf{b}^h$ ν_1 vezes em Ω^h com estimativa inicial \mathbf{v}^h ;
 2. Calcular $\mathbf{r}^h = \mathbf{b}^h - A^h\mathbf{v}^h$;
 3. Restringir o resíduo da malha Ω^h para a malha Ω^{2h} : $I_h^{2h}\mathbf{r}^h = \mathbf{b}^{2h}$;
 - (a) Iterar $A^{2h}\mathbf{u}^{2h} = \mathbf{b}^{2h}$ ν_1 vezes em Ω^{2h} com estimativa inicial $\mathbf{v}^{2h} = \mathbf{0}$;
 - (b) Calcular $\mathbf{r}^{2h} = \mathbf{b}^{2h} - A^{2h}\mathbf{v}^{2h}$;
 - (c) Restringir o resíduo da malha Ω^{2h} para a malha Ω^{4h} : $I_{2h}^{4h}\mathbf{r}^{2h} = \mathbf{b}^{4h}$;
 - i. Iterar $A^{4h}\mathbf{u}^{4h} = \mathbf{b}^{4h}$ ν_1 vezes em Ω^{4h} com estimativa inicial $\mathbf{v}^{4h} = \mathbf{0}$;
 - ii. Calcular $\mathbf{r}^{4h} = \mathbf{b}^{4h} - A^{4h}\mathbf{v}^{4h}$;
 - iii. Restringir o resíduo da malha Ω^{4h} para a malha Ω^{8h} : $I_{4h}^{8h}\mathbf{r}^{4h} = \mathbf{b}^{8h}$;
 - .
 - .
 - .
 - iv. Corrigir $\mathbf{v}^{4h} \leftarrow \mathbf{v}^{4h} + I_{8h}^{4h}\mathbf{v}^{8h}$;
 - v. Iterar $A^{4h}\mathbf{u}^{4h} = \mathbf{b}^{4h}$ ν_2 vezes em Ω^{4h} com estimativa inicial \mathbf{v}^{4h} ;
 - (d) Corrigir $\mathbf{v}^{2h} \leftarrow \mathbf{v}^{2h} + I_{4h}^{2h}\mathbf{v}^{4h}$;
 - (e) Iterar $A^{2h}\mathbf{u}^{2h} = \mathbf{b}^{2h}$ ν_2 vezes em Ω^{2h} com estimativa inicial \mathbf{v}^{2h} ;
4. Corrigir $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$;
 5. Iterar $A^h\mathbf{u}^h = \mathbf{b}^h$ ν_2 vezes em Ω^h com estimativa inicial \mathbf{v}^h .
-

2.2 Conceitos de paralelização

Neste trabalho duas formas de paralelização foram adotadas: de processos e de dados. A paralelização temporal dos métodos de projeção foi realizada através de modificação para que permitam que processos sejam paralelizados. Como a pressão nestes métodos não permite paralelização de processos optou-se por paralelização de dados através do particionamento do domínio em subdomínios, ou seja, considera-se que cada subdomínio pode ser resolvido com o mesmo algoritmo. O modelo de paralelização que aplica o mesmo programa em cada um dos subdomínios é conhecido como SPMD (*Single Program Multiple Data*) (Chapman et al., 2007).

O particionamento do domínio do problema consiste em dividir o domínio em subdomínios de forma que cada *thread* (processador físico ou lógico) resolva uma parcela do sistema de equações que define o problema em um menor intervalo de tempo.

Não existe restrição na forma em que o domínio é dividido, mas algumas demandas precisam ser respeitadas (Al-Nasra e Nguyen, 1991):

1. Distribuir a carga entre os processadores de forma balanceada. Cada processador deve receber carga proporcional à sua capacidade de processamento;
2. Minimizar o tempo de comunicação entre os processadores reduzindo as dimensões dos contornos dos subdomínios;
3. O tempo gasto no processo de partição do domínio deve ser pequeno em relação ao tempo gasto na solução do problema;
4. O algoritmo deve ser capaz de tratar geometrias irregulares e discretizações arbitrárias.

O problema do particionamento do domínio sujeito às restrições 1) e 2) pode ser classificado como problema MPE (*Minimum Perimeter Equi-partition*): uma malha retangular $M \times N$ deve ser dividida em \bar{T} (*threads* ou processadores) subdomínios com o mesmo número de volumes e com o menor perímetro (contorno dos subdomínios) possível (Christou e Meyer, 1996; Martin, 1998). Muitos dos subdomínios resultantes do MPE, que satisfazem 1) e 2), são figuras irregulares e, normalmente, representadas por uma união de retângulos (Yackel e Meyer, 1992), como mostram as figuras 2.11(a) e 2.11(b) extraídas de Yackel e Meyer (1992) e Martin (1998), respectivamente, o que pode resultar em dificuldades na implementação do método *multigrid*.

Pode-se particionar as malhas decompondo estas em subdomínios retangulares, como mostra a Fig. 2.11(c) extraída de Trottenberg et al. (2001) onde uma malha com 16×16 volumes é particionada em subdomínios de 1×16 (1D) volumes e 4×4 (2D) volumes. Subdomínios retangulares implicam em menor dificuldade comparada aos casos citados por Yackel e Meyer (1992), Christou e Meyer (1996), e Martin (1998). No entanto a diferença entre o número de volumes por processador (desbalanceamento de carga) não pode ser evitada, o que pode resultar em desempenho insatisfatório da paralelização (Trottenberg et al., 2001).

O algoritmo PERIX-GA de Christou e Meyer (1996) satisfaz o balanceamento de carga e minimiza o tempo de comunicação entre os processadores, contudo torna-se inviável devido ao aumento expressivo no tempo necessário para encontrar as formas ótimas dos subdomínios (Martin, 1998).

O algoritmo MSP (*Minimum Striped Perimeter*) de Martin (1998) apresenta tempo de processamento significativamente menor que o de Christou e Meyer (1996), todavia, o tempo de processamento cresce linearmente com N .

Donaldson (2000) apresenta revisões de outras metodologias, entretanto considera que as metodologias de Christou e Meyer (1996) e Martin (1998) são as que apresentam os melhores resultados.

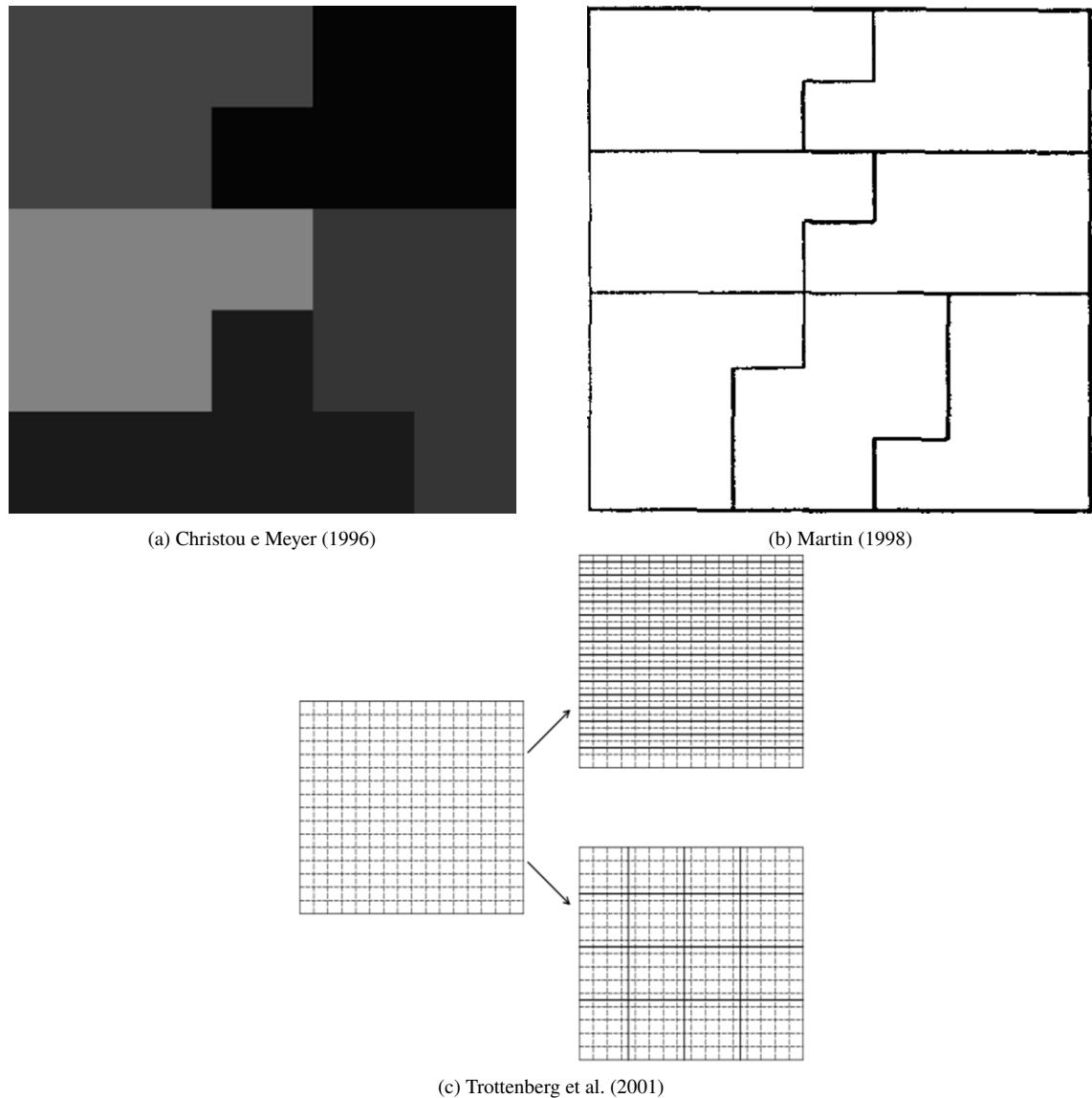


Figura 2.11: Particionamento da malha em subdomínios.

Neste trabalho são tratados com especial atenção o problema do balanceamento de carga e o problema do tempo gasto no processo de partição do domínio. Como foram utilizadas apenas malhas estruturadas e discretização pelo método dos volumes finitos, o item 4) é automaticamente satisfeito.

A ausência da restrição 2), ignorada neste do trabalho, implica em encontrar subdomínios com balanceamento de carga ótimo. Desta forma, representa um problema muito mais simples que os estudados por Yackel e Meyer (1992), Christou e Meyer (1996), e Martin (1998).

O presente trabalho utiliza uma metodologia de particionamento do domínio com baixo custo computacional, que não depende do tamanho da malha utilizada e apresenta balanceamento de carga ótimo (no máximo um volume de diferença entre os processadores). Diferentemente dos casos apresentados na Fig. 2.11, os subdomínios resultantes da metodologia utilizada aqui possuem forma não retangular e, em geral, com forma geométrica desconhecida. No entanto se ajustam perfeitamente aos algoritmos de paralelização que serão apresentados.

2.2.1 Subdomínios e balanceamento de carga

O modelo de paralelização SPMD consiste na aplicação do mesmo programa (mesmo conjunto de instruções sequenciais) em cada subdomínio. Logo, pode-se concluir que, para o caso de processadores com as mesmas especificações (desempenho), o balanceamento de carga pode ser atingido através de uma divisão do domínio em subdomínios de mesma dimensão.

De acordo com a conclusão acima, o número de volumes de cada subdomínio pode ser calculado como

$$N_{sub} = \left\lfloor \frac{N_x N_y}{\bar{T}} \right\rfloor \quad (2.7)$$

onde \bar{T} é o número total de *threads* disponíveis na máquina. O símbolo $\lfloor \rfloor$ refere-se à função piso definida por Graham et al. (1994)

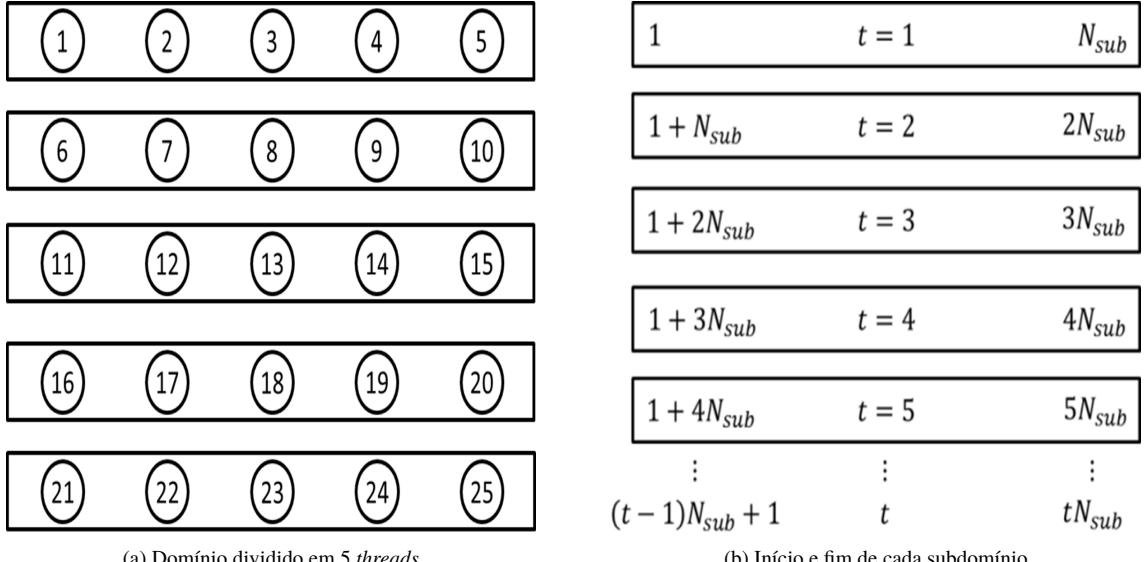
$$\text{piso} : R \rightarrow Z, \text{ com } x \mapsto \text{piso}(x) = \max\{z \in Z \mid z \leq x\}. \quad (2.8)$$

A Eq. (2.7) distribui o número de volumes no interior do domínio entre os *threads* disponíveis. Uma consequência importante da Eq. (2.7) é que o número máximo de volumes que podem “sobrar” na divisão é $\bar{T} - 1$, e estes podem ser redistribuídos entre os *threads* de forma que a diferença entre o número de volumes entre os subdomínios não seja maior que um, técnica que será apresentada a seguir.

A Fig. 2.12(a) mostra a divisão exata de um domínio com 25 volumes distribuídos entre 5 *threads* de forma que $N_{sub} = 5$. Na Fig. 2.12(b), $t = 1, 2, \dots, \bar{T}$ é o número que identifica o *thread* t , e, portanto, o subdomínio do qual este vai se encarregar. Ainda na Fig. 2.12(b), pode-se notar como os índices do primeiro e do último volume de cada subdomínio estão relacionados com o *thread* t .

O caso geral (onde a divisão na Eq. (2.7) não é exata) é apresentado na Fig. 2.13(a). Nota-se que existem volumes que precisam ser redistribuídos. Neste exemplo, a Fig. 2.13(a), tem-se um domínio com 28 volumes que é distribuído entre 5 *threads* com a Eq. (2.7), de forma que 3 volumes precisam ser redistribuídos. A escolha de quais *threads* receberão os volumes que “sobram” é arbitrária, desde que cada *thread* receba apenas 1 volume extra. Neste trabalho, optou-se por redistribuir os volumes entre os *threads* $t \geq t_c$, onde

$$t_c = \bar{T} + 1 - \left[N_x N_y - \bar{T} N_{sub} \right]. \quad (2.9)$$



(a) Domínio dividido em 5 threads

(b) Início e fim de cada subdomínio

Figura 2.12: Divisão exata de um subdomínio.

Cada *thread* a partir de t_c (este *thread* incluso) receberá 1 volume dos que sobraram na divisão do domínio com a Eq. (2.7). No exemplo da Fig. 2.13 $t_c = 3$, ou seja, todos os *threads* a partir do terceiro, $t = 3, 4$, e 5 , receberão 1 volume extra. Na Eq. (2.9), pode-se notar que $\bar{T} - t_c + 1$ é igual ao número de volumes que sobram na divisão da Eq. (2.7).

Na Fig. 2.13(b) cada *thread* $t \geq t_c$ recebe +1 volume (colocado fora do subdomínio para ressaltar que a escolha é arbitrária) em seu subdomínio de forma que os índices do primeiro e último subdomínios precisam ser corrigidos com o acréscimo dos termos $C(t - t_c)$ e $C(t - t_c + 1)$, respectivamente. O valor C é definido por

$$C = \begin{cases} 1, & \text{se } t \geq t_c \\ 0, & \text{se } t < t_c \end{cases} \quad (2.10)$$

Da Fig. 2.13(b) pode-se concluir que os índices que partitionam o domínio em subdomínios, na ordenação lexicográfica, são dados por

$$i_t = (t - 1)N_{sub} + 1 + C(t - t_c) \quad (2.11)$$

$$f_t = tN_{sub} + C(t - t_c + 1),$$

em que i_t representa o início e f_t representa o fim de cada subdomínio. O termo C é responsável por “ativar” a redistribuição dos volumes que “sobraram” na Eq. (2.7).

A Eq. (2.11) resulta no particionamento do domínio com ótimo balanceamento de carga, pois implica que um *thread* vai resolver, no máximo, uma equação a mais que os outros. Além disso, implica também que o número de subdomínios sempre será igual ao número de *threads* disponíveis, evitando que processadores fiquem ociosos.

A Fig. 2.14 ilustra um domínio com 49 volumes distribuídos, utilizando a Eq. (2.11), entre: (a) 4 *threads* e (b) 8 *threads*. As formas geométricas representam os subdomínios. Pode-se observar nas duas situações, (a) e (b), que a diferença no número de volumes entre os subdomínios não passa de um.

Como cada domínio (malha utilizada no método *multigrid*) precisa ser particionado apenas uma vez, o custo computacional da Eq. (2.11) pode ser considerado pequeno em relação ao custo de resolver um sistema de equações com muitas incógnitas.

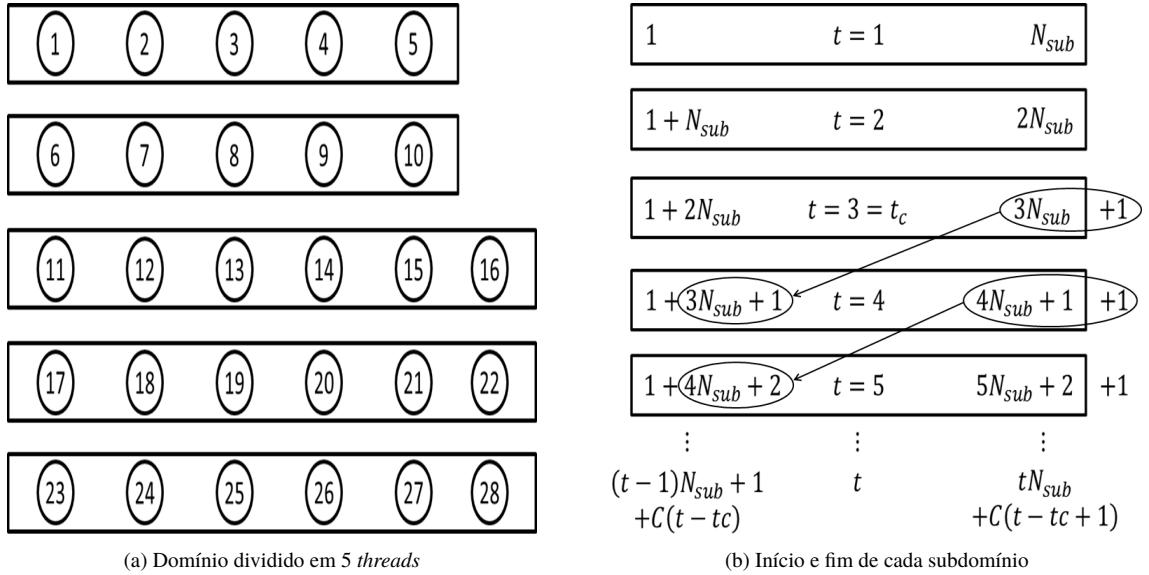


Figura 2.13: Divisão não exata de um subdomínio.

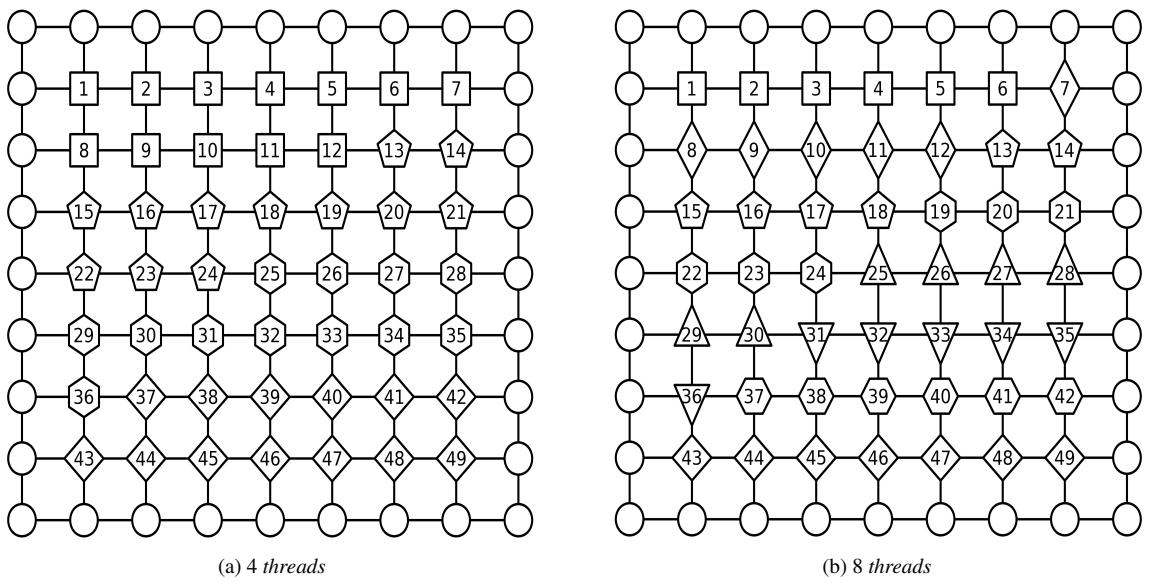


Figura 2.14: Domínio com 49 volumes partitionado em (a) 4 subdomínios e (b) 8 subdomínios.

2.2.2 Lei de Amdahl e métricas de desempenho

O objetivo principal da paralelização de uma aplicação é reduzir o tempo de CPU, ou seja, melhorar o desempenho desta. Uma das principais formas de avaliar o ganho de desempenho obtido com a paralelização é através do *speedup*.

O *speedup* é uma medida empregada para avaliar o aumento de velocidade obtido durante a execução de um programa utilizando um algoritmo “A” em relação a sua execução utilizando um algoritmo “B” (Trottenberg et al., 2001)

$$S_P = \frac{t_{CPU}(\text{Algoritmo A})}{t_{CPU}(\text{Algoritmo B})}. \quad (2.12)$$

No contexto do paralelismo, o *speedup* é empregado para avaliar o aumento de velocidade, ou “aceleração”, obtido durante a execução de um programa utilizando um algoritmo “A”, em apenas um processador, em relação a sua execução em n processadores (Chapman et al., 2007; Trottenberg et al., 2001). O *speedup*, S_n devido a n processadores é dado pela fórmula

$$S_n = \frac{t_1(\text{Algoritmo A})}{t_n(\text{Algoritmo A})}. \quad (2.13)$$

onde t_1 é o tempo de CPU utilizando apenas 1 processador (tempo de execução serial do programa) e t_n é o tempo de CPU utilizando n processadores (tempo de execução em paralelo do programa).

Quanto maior o valor de S_n , mais rápido se encontra a versão em paralelo do programa. Porém, devido à execução sequencial de regiões do programa, o valor máximo que S_n pode atingir é limitado pelo tempo que o algoritmo “A” utiliza realizando o processamento serial. Este fenômeno é conhecido como “lei de Amdahl” (Amdahl, 1967).

Seguem abaixo as premissas da lei de Amdahl:

1. O tempo de execução T' , de um programa utilizando um algoritmo “A” é a soma do tempo T_S que o programa emprega realizando processamento serial, com o tempo T_P que o programa utiliza realizando processamento em paralelo, ou seja, $T' = T_S + T_P$;
2. Apenas o tempo de processamento realizado em paralelo, T_P , é afetado pela adição de processadores ao sistema $\left(T_P, \frac{T_P}{2}, \frac{T_P}{3}, \dots\right)$. O tempo de processamento serial não pode ser reduzido pela adição de processadores;
3. Para n processadores, o tempo de CPU é dado por:

$$t_n = T_S + \frac{T_P}{n}; \quad (2.14)$$

4. Teoricamente, com a adição ilimitada de processadores, o tempo de execução do programa tende para o tempo gasto no processamento serial do mesmo.

Na Fig. 2.15 estão ilustradas tais premissas.

A fórmula matemática que descreve a lei de Amdahl pode ser obtida substituindo a Eq. (2.14) na Eq. (2.13), ou seja, o *speedup* devido a n processadores pode ser calculado com:

$$S_n = \frac{t_1}{t_n} = \frac{T_S + T_P}{T_S + \frac{T_P}{n}} = \frac{1}{T'_S + \frac{T'_P}{n}} = \frac{1}{(1 - T'_P) + \frac{T'_P}{n}}, \quad (2.15)$$

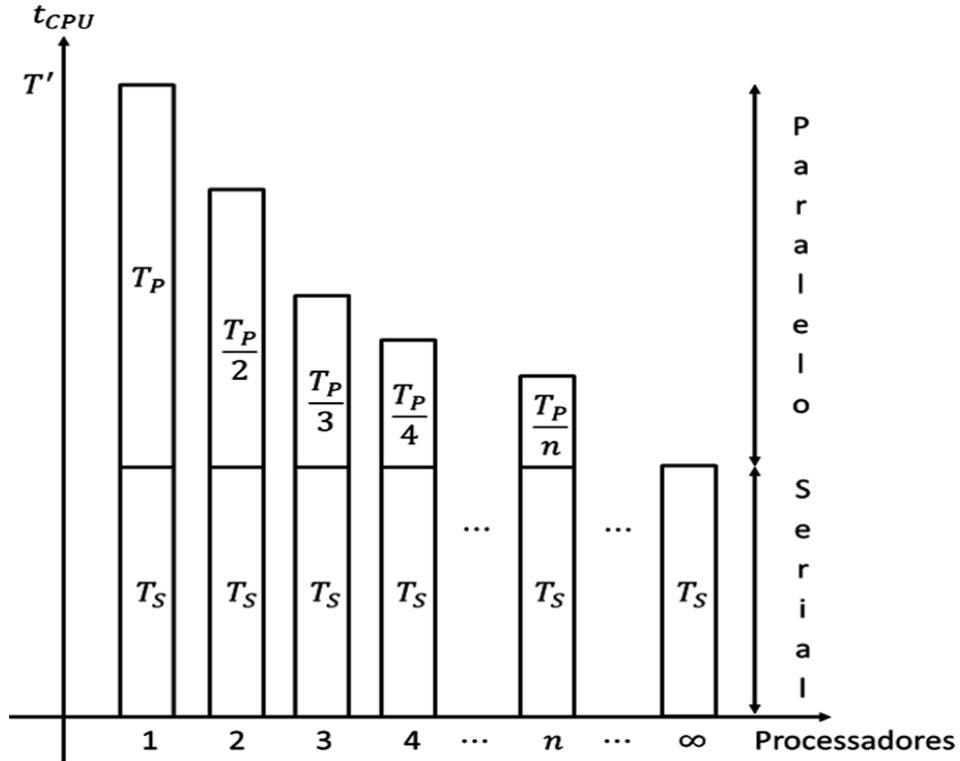


Figura 2.15: Premissas da lei de Amdahl.

onde $T'_S = \frac{T_S}{T'}$ é a fração do tempo T' de execução (do programa) devido ao processamento sequencial, e $T'_P = \frac{T_P}{T'}$ é a fração do tempo empregado no processamento paralelo. Por simplicidade pode-se adimensionalizar tal medida e com isso, considerar $T' = 1$. Assim

$$S_n = \frac{1}{(1 - T_P) + \frac{T_P}{n}}. \quad (2.16)$$

Pode-se notar que se $n \rightarrow \infty$, então $S_n \rightarrow 1/T_S$, ou seja, o valor do *speedup* é limitado pela parcela serial do processamento.

Na literatura (Sun e Chen, 2010) é comum encontrar a lei de Amdahl em sua forma mais genérica

$$S_{\text{Amdahl}} = \frac{1}{(1 - f) + \frac{f}{n}}, \quad (2.17)$$

onde f representa a parcela do processamento que pode ser melhorada por um fator m e o restante, $(1 - f)$, a parcela que não pode ser melhorada. No contexto do paralelismo, f representa a parcela do trabalho que pode ser paralelizada e m , o número de processadores utilizados.

A Fig. 2.16 ilustra o *speedup* teórico máximo previsto, pela Eq. (2.17), para programas que têm, hipoteticamente, frações paralelas f de 50%, 80%, 90% e 95%.

O *speedup* teórico calculado pela Eq. (2.17) representa o valor ideal desta métrica, pois considera que f é perfeitamente paralelizável, ou seja, ignora custos como a comunicação e sincronização dos processadores, tempo de acesso à memória, entre outros.

É importante observar que o *speedup* previsto pela Eq. (2.17) é válido apenas para o caso em que a “carga de trabalho” é fixa (Gustafson, 1988; Sun e Chen, 2010). Neste trabalho, pode-se interpretar “carga de trabalho” como o número de incógnitas do sistema de equações resultante das discretizações.

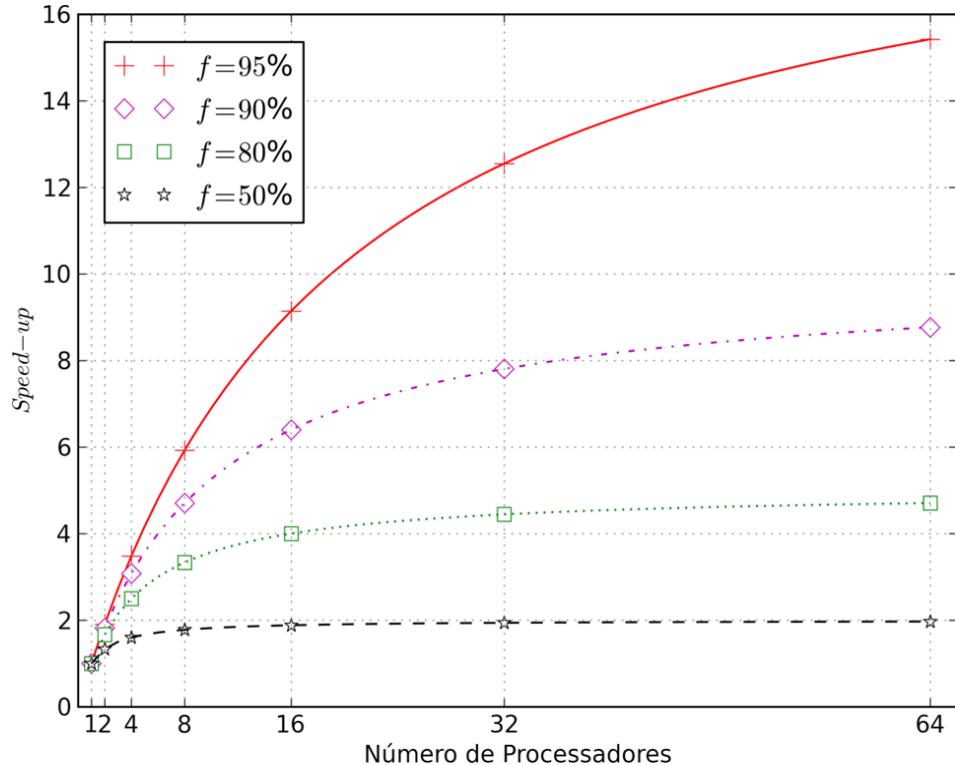


Figura 2.16: Limite superior de *speedup* para programas com frações paralelas entre 50% a 95%.

A Eq. (2.17) é importante no contexto do paralelismo, pois modela a escalabilidade do sistema (considerando uma carga de trabalho fixa). A escalabilidade de um sistema paralelo pode ser interpretada como uma medida de sua capacidade de aumentar o fator *speedup* em proporção ao número de elementos de processamento. Desta forma, a escalabilidade reflete a capacidade de empregar eficientemente novos elementos de processamento introduzidos no sistema (Gramma et al., 2003).

Existem outras leis de escalabilidade, como a lei de Gustafson (Gustafson, 1988) e a lei de Sun e Ni (Sun e Ni, 1993). Esta última é a generalização da lei de Amdahl e Gustafson. No contexto destas leis, a carga de trabalho não é fixa (a carga de trabalho aumenta com o número de processadores), portanto a escalabilidade é diferente da prevista pela lei de Amdahl.

Os problemas estudados neste trabalho são governados pela lei de Amdahl, pois se tratam de problemas cuja carga de trabalho é fixa, ou seja, os mesmos sistemas de equações são resolvidos com diferentes números de processadores.

Outra métrica utilizada na avaliação do desempenho da paralelização é conhecida como eficiência e é definida como

$$E = \frac{S_n}{n}, \quad (2.18)$$

onde S_n é o *speedup* calculado com a Eq. (2.13) e n o número de processadores.

O valor da eficiência, calculada pela Eq. (2.18), encontra-se entre 0 e 1. Esta métrica pode ser interpretada como uma medida do tempo em que os processadores são efetivamente empregados no processamento paralelo (Gramma et al., 2003).

3 Modelo matemático

Neste capítulo apresentam-se o modelo matemático e os problemas estudados. São apresentados os domínios, condições de contorno e são discutidos os propósitos de cada problema.

O modelo matemático considerado refere-se ao escoamento laminar bidimensional em regime transitório sem forças externas de fluido incompressível modelado pelas equações de Navier-Stokes (Pritchard e Mitchell, 2015)

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

$$\rho \frac{\partial \hat{\mathbf{u}}}{\partial \hat{t}} + \rho (\hat{\mathbf{u}} \cdot \nabla \hat{\mathbf{u}}) = -\nabla \hat{p} + \mu \nabla^2 \hat{\mathbf{u}} \quad (3.2)$$

onde

- \hat{t} é a coordenada temporal (variável independente) (s);
- $\hat{\mathbf{u}}$ é o vetor velocidade (m/s);
- \hat{p} é a pressão estática do fluido (Pa);
- ρ é a massa específica (Kg/m^3); e
- μ a viscosidade absoluta do fluido (Pa s).

Visto que ρ e μ são constantes homogêneas e considerando-se as grandezas de referência L_∞ (m) e U_∞ (m/s) as seguintes variáveis adimensionais podem ser escritas (Pritchard e Mitchell, 2015)

$$x = \frac{\hat{x}}{L_\infty} \quad (3.3)$$

$$y = \frac{\hat{y}}{L_\infty} \quad (3.4)$$

$$\mathbf{u} = \frac{\hat{\mathbf{u}}}{U_\infty} \quad (3.5)$$

$$t = \frac{\hat{t} U_\infty}{L_\infty} \quad (3.6)$$

$$P = \frac{\hat{p}}{\rho U_\infty^2} \quad (3.7)$$

onde \hat{x} e \hat{y} são as coordenadas espaciais (variáveis independentes) (m).

Substituindo as Eqs. (3.3-3.7) nas Eqs. (3.1) e (3.2) obtem-se as equações de Navier-Stokes na forma adimensional (Kim e Moin, 1985):

$$\nabla \cdot \mathbf{u} = 0, \quad (3.8)$$

$$\frac{\partial u}{\partial t} + \mathbf{u} \cdot \nabla u = -\frac{\partial P}{\partial x} + \frac{1}{\text{Re}} \nabla^2 u, \quad (3.9)$$

$$\frac{\partial v}{\partial t} + \mathbf{u} \cdot \nabla v = -\frac{\partial P}{\partial y} + \frac{1}{\text{Re}} \nabla^2 v, \quad (3.10)$$

onde u e v são as componentes do vetor velocidade \mathbf{u} nas direções x e y , respectivamente, e

$$\text{Re} = \frac{\rho U_\infty L_\infty}{\mu} \quad (3.11)$$

é o número de Reynolds.

O número de Reynolds expressa a razão entre as características inerciais $\rho U_\infty L_\infty$ e viscosas μ do fluido e é utilizado na caracterização do comportamento do escoamento, sendo este classificado em laminar, turbulento ou transitório. Escoamentos abaixo de um valor crítico $\text{Re}_{\text{crítico}}$ são classificados como laminares e as forças viscosas predominam sobre as inerciais. Acima de $\text{Re}_{\text{crítico}}$ o comportamento é chamado de turbulento e as forças inerciais são predominantes sobre as viscosas. No problema da cavidade quadrada, a ser apresentado neste capítulo, Erturk (2009) observou que para $\text{Re} < 6000$ o escoamento pode ser considerado laminar, para $6000 < \text{Re} < 8000$ o comportamento é transitório pois começa a apresentar sinais de turbulência e para $\text{Re} > 8000$ o escoamento é considerado turbulento.

Como será discutido no capítulo 4 é necessário reescrever as equações de Navier-Stokes com os termos não lineares na forma conservativa:

$$\frac{\partial u}{\partial t} + \left(u^2 \right)_x + (uv)_y = -\frac{\partial P}{\partial x} + \frac{1}{\text{Re}} \nabla^2 u, \quad (3.12)$$

$$\frac{\partial v}{\partial t} + (uv)_x + \left(v^2 \right)_y = -\frac{\partial P}{\partial y} + \frac{1}{\text{Re}} \nabla^2 v. \quad (3.13)$$

Para testar a metodologia numérica foram resolvidos 2 problemas:

Problema 1: Vórtices de Taylor-Green (Pearson, 1965)

Domínio dado por $\{(x, y) \in R^2 : 0 \leq x \leq \pi \text{ e } 0 \leq y \leq \pi\}$ e soluções analíticas

$$\begin{aligned} u &= -(\cos x \cdot \sin y) e^{-2(t+Ti)} \\ v &= (\sin x \cdot \cos y) e^{2(t+Ti)} \\ p &= -\frac{1}{4} (\cos 2x + \sin 2y) e^{-4(t+Ti)} \end{aligned} \quad (3.14)$$

Na Eq. (3.14) Ti é o tempo inicial da simulação. As condições iniciais e de contorno são obtidas diretamente da solução analítica.

O propósito deste problema é validar a implementação do código bem como a acurácia das soluções numéricas pois, visto que possui solução analítica, permite que a ordem do erro numérico seja calculada. Além disso é possível avaliar a discretização temporal comparando-se os resultados com estimativas teóricas apresentadas em Guermond et al. (2006).

Problema 2: Cavidade quadrada com tampa deslizante (Ghia et al., 1982)

Domínio dado por $\{(x, y) \in R^2 : 0 \leq x \leq 1 \text{ e } 0 \leq y \leq 1\}$ cujas condições de contorno são mostradas na Fig. 3.1 e apresentadas na Eq. (3.15):

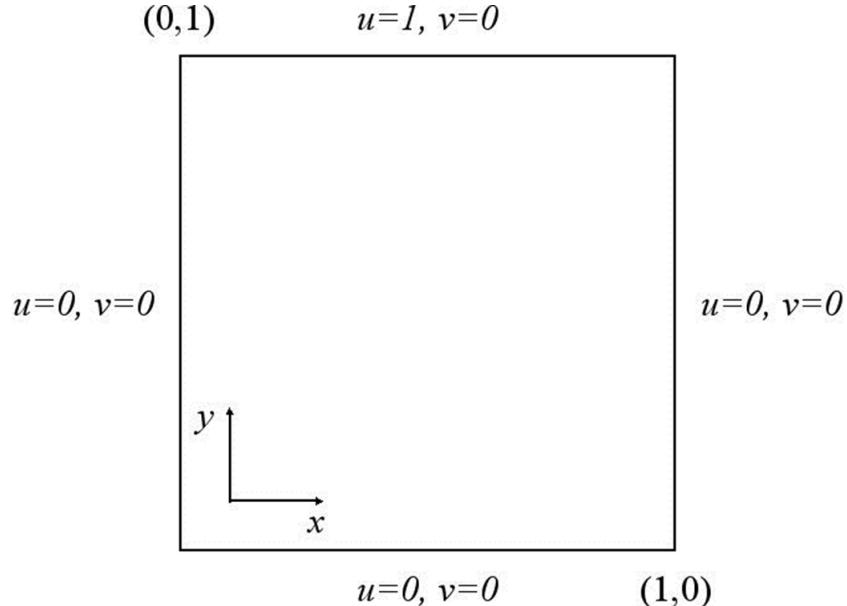


Figura 3.1: Domínio e condições de contorno.

$$\begin{aligned}
 u(x, 1) &= 1 \\
 u(0, y) &= u(1, y) = u(x, 0) = 0 \\
 v(0, y) &= v(1, y) = v(x, 0) = v(x, 1) = 0
 \end{aligned} \tag{3.15}$$

O problema 2 não possui solução analítica, no entanto as soluções numéricas apresentadas por Ghia et al. (1982) são amplamente utilizadas na literatura como referência na comparação dos resultados numéricos.

Visto que a solução analítica do problema 1 é válida apenas para $Re = 1$ o problema 2 é utilizado principalmente para testar o código implementado considerando vários valores de Reynolds para escoamentos com comportamento laminar, isto é, $Re < 6000$ (Erturk, 2009).

4 Metodologia numérica

Neste capítulo é discutida a metodologia numérica utilizada no trabalho. São apresentados o tipo de malha utilizada, a discretização das equações de Navier-Stokes, os métodos de projeção sequenciais no tempo, os coeficientes das velocidades de pressão, as restrições no passo de tempo e condição de integrabilidade.

4.1 Tipo de malha utilizada

Neste trabalho são utilizadas malhas desencontradas do tipo Harlow e Welch (1965) para o domínio espacial, também utilizadas por Maliska (2004), em que as componentes da velocidade são posicionadas nas faces e a pressão no centro dos volumes. A razão disso é para evitar instabilidades numéricas na solução da pressão (Trottenberg et al., 2001).

Na Fig. 4.1 os índices pu , pv e pp representam as posições (em um volume genérico) das variáveis u , v e P respectivamente.

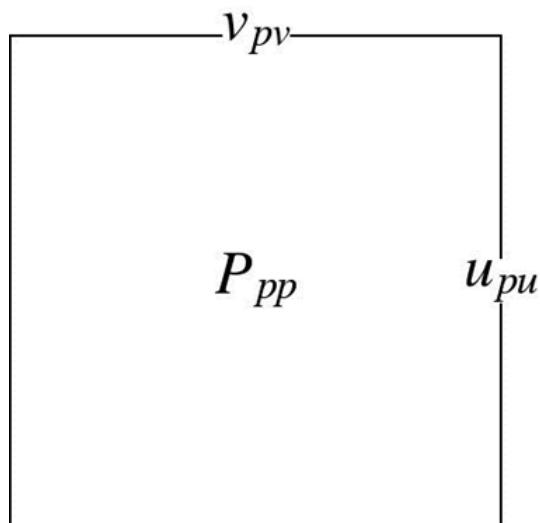


Figura 4.1: Disposição das variáveis: velocidades nas faces e pressão no centro dos volumes.

A Fig. 4.2 ilustra uma malha de 8 por 8 volumes, nela é apresentada a ordenação que será adotada ao longo do trabalho. Nesta mesma figura os volumes com linha tracejada são chamados de volumes fictícios e não pertencem ao domínio físico do problema. Além disso, pode-se observar que utilizando a malha desencontrada, as condições de contorno para a variável u são automaticamente prescritas nos contornos oeste e leste. Nos contornos norte e sul é necessário algum tipo de procedimento para que os volumes fictícios levem em consideração a informação dos contornos; neste trabalho utilizou-se interpolação linear. De forma similar ao caso do variável u , nota-se, na Fig. 4.2, que as condições de contorno na variável v são

automaticamente prescritas nos contornos sul e norte, e nos contornos oeste e leste, extração linear.

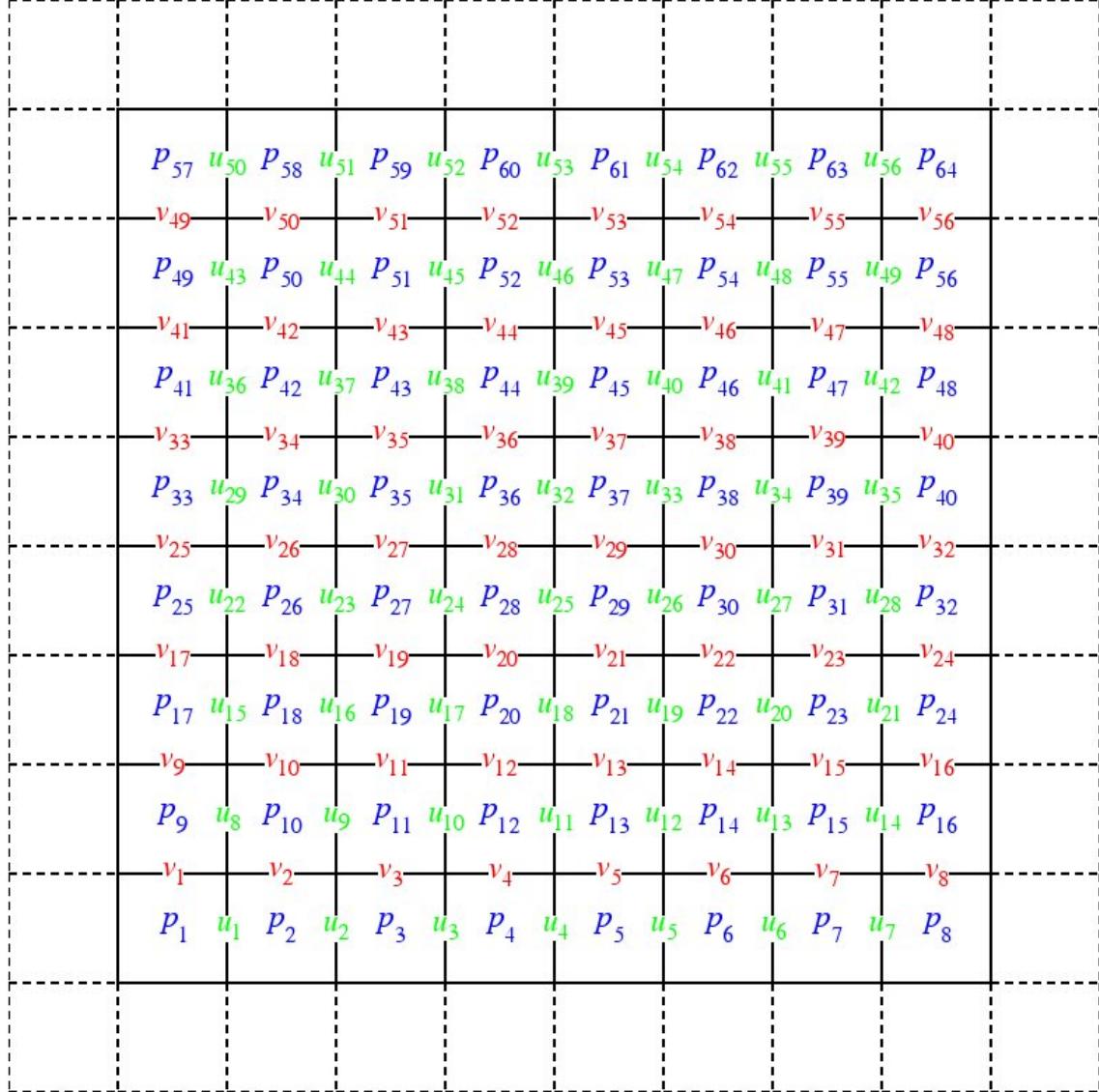


Figura 4.2: Malha e ordenação lexicográfica das variáveis pressão e velocidades.

4.2 Discretização espacial por volumes finitos

As Eqs. (3.8), (3.9) e (3.10) podem ser integradas em cada volume de controle mostrado na Fig. 4.3 e serem reescritas como:

Equação da conservação da massa (massa)

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS = 0. \quad (4.1)$$

Equação da conservação da quantidade de movimento linear na direção x (QML- x)

$$\frac{\partial}{\partial t} \int_V u dV = - \oint_S u \mathbf{u} \cdot \mathbf{n} dS - \oint_S P n_x dS + \frac{1}{Re} \oint_S \nabla u \cdot \mathbf{n} dS. \quad (4.2)$$

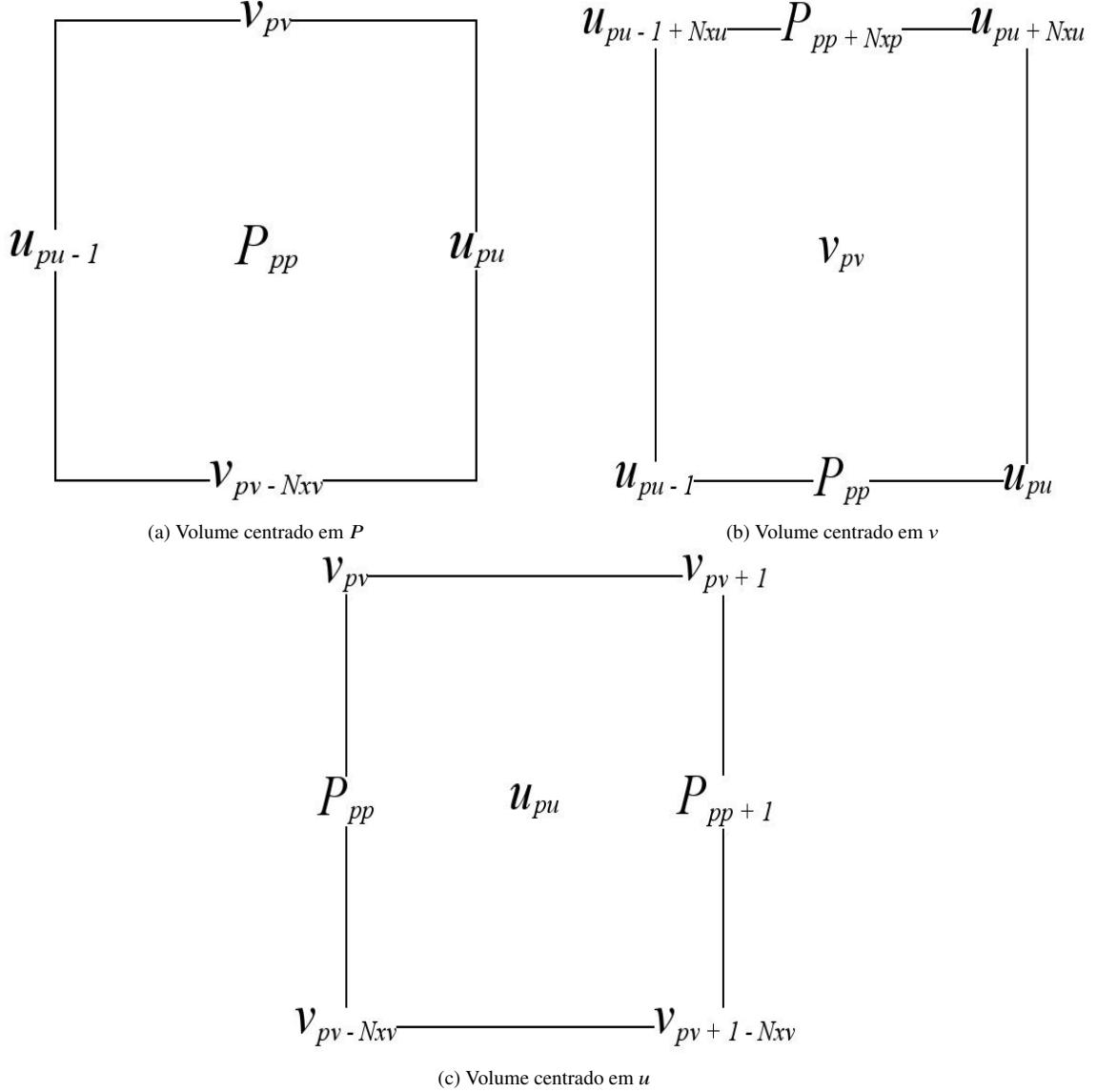


Figura 4.3: Volumes de controle.

Equação da conservação da quantidade de movimento linear na direção y (QML- y)

$$\frac{\partial}{\partial t} \int_V v \, dV = - \oint_S v \mathbf{u} \cdot \mathbf{n} \, dS - \oint_S P n_y \, dS + \frac{1}{\text{Re}} \oint_S \nabla v \cdot \mathbf{n} \, dS. \quad (4.3)$$

Os termos das Eqs. (4.1), (4.2) e (4.3) foram discretizados utilizando o Método dos Volumes Finitos (MVF) empregando malhas desencontradas calculando-se tais integrais sobre os volumes de controle definidos na Fig. 4.3. Por questão de simplicidade foi utilizado o mesmo refinamento espacial em todas as direções, ou seja, $h_x = h_y = h$ (malhas uniformes).

A equação da conservação da massa, Eq. (4.1), é discretizada utilizando a Fig. 4.3(a), ou seja, um volume centrado em P

$$\oint_S \mathbf{u} \cdot \mathbf{n} \, dS \approx u_{pu}^{n+1} - u_{pu-1}^{n+1} + v_{pv}^{n+1} - v_{pv-Nxv}^{n+1} = 0. \quad (4.4)$$

Os termos das velocidades com variação no tempo são discretizados utilizando os volumes definidos nas Figs. 4.3(b) e 4.3(c) de acordo com

$$\begin{aligned}\frac{\partial}{\partial t} \int_V u \, dV &\approx \frac{(u_{pu}^{n+1} - u_{pu}^n)}{h_t} h^2 \\ \frac{\partial}{\partial t} \int_V v \, dV &\approx \frac{(v_{pv}^{n+1} - v_{pv}^n)}{h_t} h^2\end{aligned}. \quad (4.5)$$

Os termos das pressões são discretizados utilizando os volumes definidos nas Figs. 4.3(b) e 4.3(c) de acordo com

$$\begin{aligned}\oint_S P n_x \, dS &\approx (P_{pp+1} - P_{pp})h \\ \oint_S P n_y \, dS &\approx (P_{pp+Nxp} - P_{pp})h\end{aligned}. \quad (4.6)$$

Utilizando os volumes de controle definidos pelas Figs. 4.3(b) e 4.3(c) pode-se discretizar os termos difusivos com

$$\begin{aligned}\oint_S \nabla u \cdot \mathbf{n} \, dS &\approx u_{pu+1}^n + u_{pu-1}^n + u_{pu+Nxu}^n + u_{pu-Nxu}^n - 4u_{pu}^n \\ \oint_S \nabla v \cdot \mathbf{n} \, dS &\approx v_{pv+1}^n + v_{pv-1}^n + v_{pv+Nxv}^n + v_{pv-Nxv}^n - 4v_{pv}^n\end{aligned}. \quad (4.7)$$

Os termos advectivos da componente u são discretizados também utilizando os volumes de controle definidos nas Figs. 4.3(b) e 4.3(c), ou seja, volumes centrados nas componentes das velocidades

$$\oint_S u \mathbf{u} \cdot \mathbf{n} \, dS \approx \left[(u^2)_{pu+1/2}^n + (u^2)_{pu-1/2}^n + (uv)_{pu+Nxu/2}^n - (uv)_{pu-Nxu/2}^n \right] h \quad (4.8)$$

Na Eq. (4.9) são apresentados as aproximações para as componentes das velocidades utilizadas na Eq. (4.8):

$$\begin{aligned}(u^2)_{pu+1/2}^n &= \left[\frac{1}{2} (u_{pu+1}^n + u_{pu}^n) \right]^2 \\ (u^2)_{pu-1/2}^n &= \left[\frac{1}{2} (u_{pu-1}^n + u_{pu}^n) \right]^2 \\ (uv)_{pu+Nxu/2}^n &= \left[\frac{1}{2} (u_{pu}^n + u_{pu+Nxu}^n) \right] \left[\frac{1}{2} (v_{pv}^n + v_{pv+1}^n) \right] \\ (uv)_{pu-Nxu/2}^n &= \left[\frac{1}{2} (u_{pu}^n + u_{pu-Nxu}^n) \right] \left[\frac{1}{2} (v_{pv-Nxv}^n + v_{pv+1-Nxv}^n) \right]\end{aligned}. \quad (4.9)$$

De forma similar para a componente v a discretização é dada por

$$\oint_S v \mathbf{u} \cdot \mathbf{n} \, dS \approx \left[(v^2)_{pv+Nxv/2}^n + (v^2)_{pv-Nxv/2}^n + (uv)_{pv+1/2}^n - (uv)_{pv-1/2}^n \right] h, \quad (4.10)$$

e as aproximações são dadas por

$$\begin{aligned}
 (v^2)_{pv+Nxv/2}^n &= \left[\frac{1}{2} (v_{pv+Nxv}^n + v_{pv}^n) \right]^2 \\
 (v^2)_{pv-Nxv/2}^n &= \left[\frac{1}{2} (v_{pv}^n + v_{pv-Nxv}^n) \right]^2 \\
 (uv)_{pv+1}^n &= \left[\frac{1}{2} (u_{pu}^n + u_{pu+Nxu}^n) \right] \left[\frac{1}{2} (v_{pv}^n + v_{pv+1}^n) \right] \\
 (uv)_{pv-1}^n &= \left[\frac{1}{2} (u_{pu-1}^n + u_{pu-1+Nxu}^n) \right] \left[\frac{1}{2} (v_{pv-1}^n + v_{pv}^n) \right]
 \end{aligned} \quad (4.11)$$

4.3 Métodos de projeção sequenciais no tempo

Os métodos de projeção, ou métodos de passo fracionário, foram introduzidos por Chorin (1968) e Temam (1969) e são caracterizados pela solução das Eqs. (3.8), (3.9) e (3.10) em dois passos. Inicialmente uma velocidade auxiliar \mathbf{u}^t é calculada desprezando-se a condição de incompressibilidade e o termo da pressão. Neste passo é resolvida a equação com a discretização temporal. No segundo passo é calculada a pressão resolvendo-se uma equação de Poisson. Como este passo é equivalente à projeção da velocidade auxiliar \mathbf{u}^t no espaço dos campos vetoriais com divergente nulo, de acordo com o teorema de Decomposição de Hodge, são obtidas condições de contorno de Neumann para a pressão. A propriedade mais atraente dos métodos de projeção é o fato de que a cada passo de tempo resolve-se uma sequência de equações elípticas, desacopladas, para a pressão e velocidades permitindo que simulações numéricas de larga escala sejam realizadas de forma eficiente. No entanto, não é trivial desenvolver e analisar métodos de projeção de altas ordens (Guermond et al., 2006). Há várias formulações para os métodos de projeções diferindo na forma de calcular a velocidade auxiliar \mathbf{u}^t e o passo de projeção. A seguir será apresentada uma derivação do método de Chorin (1968) para a componente u de \mathbf{u} .

Substituindo as Eqs. (4.5), (4.6), (4.7) e (4.8) na Eq. (4.2) tem-se

$$\begin{aligned}
 \frac{(u_{pu}^{n+1} - u_{pu}^n)}{h_t} &= -\frac{1}{h} \left[(u^2)_{pu+1/2}^n + (u^2)_{pu-1/2}^n + (uv)_{pu+Nxu/2}^n - (uv)_{pu-Nxu/2}^n \right] \\
 &\quad - \frac{1}{h} (P_{pp+1} - P_{pp}) \\
 &\quad + \frac{1}{Reh^2} (u_{pu+1}^n + u_{pu-1}^n + u_{pu+Nxu}^n + u_{pu-Nxu}^n - 4u_{pu}^n).
 \end{aligned} \quad (4.12)$$

Substituindo os índices por i, j , a Eq. (4.12) pode ser reescrita como

$$\frac{(u_{i,j}^{n+1} - u_{i,j}^n)}{h_t} = \mathbf{A}_{i,j}^n + \mathbf{D}_{i,j}^n - \nabla P_{i,j}. \quad (4.13)$$

onde

$$\begin{aligned}\mathbf{A}_{i,j}^n &= -\frac{1}{h} \left[(u^2)_{pu+1/2}^n + (u^2)_{pu-1/2}^n + (uv)_{pu+Nxu/2}^n - (uv)_{pu-Nxu/2}^n \right] \\ \nabla P_{i,j} &= -\frac{1}{h} (P_{pp+1} - P_{pp}) \\ \mathbf{D}_{i,j}^n &= \frac{1}{\text{Re}h^2} \left(u_{pu+1}^n + u_{pu-1}^n + u_{pu+Nxu}^n + u_{pu-Nxu}^n - 4u_{pu}^n \right)\end{aligned}. \quad (4.14)$$

Introduzindo a variável auxiliar u^t , a Eq. (4.13) pode ser dividida em duas partes

$$\frac{(u^t - u_{i,j}^n)}{h_t} = \mathbf{A}_{i,j}^n + \mathbf{D}_{i,j}^n \quad (4.15)$$

e

$$\frac{(u_{i,j}^{n+1} - u^t)}{h_t} = -\nabla P_{i,j}. \quad (4.16)$$

Aplicando o divergente na Eq. (4.16) e lembrando que $\nabla \cdot u_{i,j}^{n+1} = 0$ tem-se

$$\nabla^2 P_{i,j} = \frac{1}{h_t} \nabla \cdot u^t. \quad (4.17)$$

Isolando a variável u^t na Eq. (4.15)

$$u^t = u_{i,j}^n + h_t \left(\mathbf{A}_{i,j}^n + \mathbf{D}_{i,j}^n \right). \quad (4.18)$$

As Eqs. (4.16), (4.17) e (4.18) representam o método de projeção originalmente proposto por Chorin (1968) e Temam (1969):

1. Primeiro passo:

$$\begin{aligned}\frac{(u^t - u^n)}{h_t} &= - \left[(u^n)^2 \right]_x - (u^n v^n)_y + \frac{1}{\text{Re}} \nabla^2 u^n, \\ \frac{(v^t - v^n)}{h_t} &= - (u^n v^n)_x - \left[(v^n)^2 \right]_y + \frac{1}{\text{Re}} \nabla^2 v^n, \\ \mathbf{u}^t|_{\partial\Omega} &= \mathbf{b}^n\end{aligned} \quad (4.19)$$

2. Segundo passo:

$$\nabla^2 P^{n+1} = \frac{1}{h_t} \nabla \cdot \mathbf{u}^t, \quad (4.20)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - h_t \nabla P^{n+1}, \quad (4.21)$$

sendo \mathbf{u}^t o campo de velocidades auxiliar cujas componentes são u^t e v^t , \mathbf{u}^n o campo de velocidades no passo de tempo n , h_t o refinamento temporal, P^n a pressão no passo de tempo n e \mathbf{b}^n as condições de contorno no passo de tempo n .

Devido à ausência do gradiente de pressão na Eq. (4.19), à condição de contorno de Neumann imposta na pressão $\nabla P^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = 0$, em que \mathbf{n} é o vetor normal ao contorno $\partial\Omega$, e

erros causados no passo de projeção, o método de Chorin não chega a ser um método de primeira ordem (em relação ao tempo) na pressão na norma l_2 (Guermond et al., 2006). O método de Chorin é conhecido como esquema de correção na pressão não incremental (o significado de “incremental” ficará claro adiante).

4.3.1 Método 1

Godan (1979) observou que adicionando o gradiente de pressão na Eq. (4.19) o algoritmo de Chorin apresenta melhora em termos de acurácia, ideia utilizada por van Kan (1986) para desenvolver um esquema de correção na pressão incremental de segunda ordem que pode ser descrito como

1. Primeiro passo:

$$\frac{(3\mathbf{u}^t - 4\mathbf{u}^n + \mathbf{u}^{n-1})}{2h_t} = \beta g + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^t + \nabla P^n, \quad (4.22)$$

$$\mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n$$

2. Segundo passo:

$$\nabla^2 \phi^{n+1} = \frac{3}{2h_t} \nabla \cdot \mathbf{u}^t, \quad (4.23)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{2h_t}{3} \nabla \phi^{n+1}, \quad (4.24)$$

$$P^{n+1} = P^n + \phi^{n+1}. \quad (4.25)$$

onde ϕ^n é a correção da pressão no passo de tempo n .

A partir de agora os termos convectivos $\mathbf{u} \cdot \nabla \mathbf{u}$ serão tratados como βg onde β é uma constante. A razão desta escolha ficará clara quando a generalização dos esquemas de correção na pressão for apresentada.

Devido à equação $P^{n+1} = P^n + \phi^{n+1}$ que o método acima é conhecido como correção na pressão “incremental” e também conhecido como “forma padrão”. Bell et al. (1989) utilizam este esquema com o termo advectivo calculado na iteração $n + 1/2$ (semi-implícito), ou seja, o termo advectivo é calculado a partir dos termos advectivos na iteração n e $n + 1$. Este esquema é de segunda ordem no tempo para as velocidades nas normas l_2 , l_1 e l_∞ . Para a pressão, o esquema é de segunda ordem nas normas l_2 e l_1 , mas apenas de primeira ordem na norma l_∞ . As provas destes fatos podem ser consultadas em Shen (1996) e E e Liu (1995). Este método encontra-se implementado e será analisado na seção de resultados como Método 1, na referida seção serão apresentadas evidências que corroboram as conclusões sobre as ordens de convergência do método citadas acima. A principal fonte de erro numérico do Método 1 está no fato de que este impõe condições de Neumann artificiais do tipo: $\nabla P^{n+1} \cdot \mathbf{n}|_{\partial\Omega} = \nabla P^n \cdot \mathbf{n}|_{\partial\Omega} = \dots = \nabla P^0 \cdot \mathbf{n}|_{\partial\Omega}$.

As condições de contorno de Neumann são ditas artificiais porque não ocorrem no fenômeno físico. Estas condições são impostas pelo método de projeção e são uma de suas principais fontes de erro numérico.

4.3.2 Método 2

Com objetivo de melhorar a acurácia do Método 1 (correção incremental na pressão na forma padrão), Timmermans et al. (1996) modificaram o Método 1:

1. **Primeiro passo:**

$$\frac{(3\mathbf{u}^t - 4\mathbf{u}^n + \mathbf{u}^{n-1})}{2h_t} = \beta g + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^t + \nabla P^n, \quad (4.26)$$

$$\mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n$$

2. **Segundo passo:**

$$\nabla^2 \phi^{n+1} = \frac{3}{2h_t} \nabla \cdot \mathbf{u}^t, \quad (4.27)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{2h_t}{3} \nabla \phi^{n+1}, \quad (4.28)$$

$$P^{n+1} = P^n + \phi^{n+1} - \frac{1}{\text{Re}} \nabla \cdot \mathbf{u}^t. \quad (4.29)$$

A versão acima tem como condições de contorno na pressão $\partial_n P^{n+1} = \frac{1}{\text{Re}} (\nabla \times \nabla \times \mathbf{u}^{n+1}) \cdot \mathbf{n}|_{\partial\Omega}$, que diferente do Método 1, são condições de contorno consistentes para a pressão (Guermond et al., 2006). Tal método é conhecido (Guermond e Shen, 2001, 2004) como correção incremental na pressão na forma rotacional. Guermond e Shen (2004) mostram que a versão incremental na forma rotacional é de segunda ordem para as velocidades nas normas l_2 , l_1 e l_∞ . Para a pressão o esquema é de segunda ordem nas normas l_2 , l_1 e de ordem 3/2 na norma l_∞ . Este esquema também se encontra implementado e na seção de resultados será tratado como Método 2, nela serão apresentadas evidências das conclusões sobre a convergência das normas em relação ao tempo citadas acima.

4.3.3 Generalização e Método 3

A seguir é apresentada uma generalização dos métodos de projeção com correção na pressão (Guermond et al., 2006):

1. **Primeiro passo:**

$$\frac{1}{h_t} \left(\alpha_q \mathbf{u}^t - \sum_{j=0}^{q-1} \alpha_j \mathbf{u}^{n-j} \right) = \beta g + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^t + \nabla P^{*,n+1}, \quad (4.30)$$

$$\mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n$$

onde os coeficientes α de $\alpha_q \mathbf{u}^t - \sum_{j=0}^{q-1} \alpha_j \mathbf{u}^{n-j}$ são obtidos da expansão em série de Taylor da derivada temporal utilizando diferença regressiva (*Upstream Difference Scheme*, UDS) de ordem q . Em particular, para primeira ($q = 1$) e segunda ($q = 2$) ordens tem-se

$$\alpha_q \mathbf{u}^t - \sum_{j=0}^{q-1} \alpha_j \mathbf{u}^{n-j} = \begin{cases} \mathbf{u}^{n+1} - \mathbf{u}^n & \text{se } q = 1 \\ \frac{3}{2} \mathbf{u}^{n+1} - 2\mathbf{u}^n + \frac{1}{2} \mathbf{u}^{n-1} & \text{se } q = 2 \end{cases}. \quad (4.31)$$

O termo $P^{*,n+1}$ representa uma extrapolação linear de ordem r , em particular, para $r = 0$, $r = 1$ (primeira ordem) e $r = 2$ (segunda ordem) tem-se

$$P^{*,n+1} = \begin{cases} 0 & \text{se } r = 0 \\ P^n & \text{se } r = 1 \\ 2P^n - P^{n-1} & \text{se } r = 2 \end{cases} \quad (4.32)$$

2. Segundo passo:

$$\nabla^2 \phi^{n+1} = \frac{\alpha_q}{h_t} \nabla \cdot \mathbf{u}^t, \quad (4.33)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{h_t}{\alpha_q} \nabla \phi^{n+1}, \quad (4.34)$$

$$P^{n+1} = P^{*,n+1} + \phi^{n+1} - \frac{\chi}{\text{Re}} \nabla \cdot \mathbf{u}^t. \quad (4.35)$$

onde o termo χ é uma constante que assume os valores 0 ou 1 e sua escolha resulta na forma padrão ($\chi = 0$) ou na forma rotacional ($\chi = 1$) do esquema de correção na pressão.

Evidentemente que diferentes combinações de q , r e χ irão gerar diferentes métodos, por exemplo, pode-se citar que a combinação $q = 1$, $r = 0$ e $\chi = 0$ resulta no método original de Chorin.

Para a combinação $q = 2$, $r = 2$ e $\chi = 1$ Guermond et al. (2006) descrevem o método resultante como de segunda ordem tanto nas velocidades como na pressão para todas as normas consideradas l_2 , l_1 e l_∞ . O algoritmo do método em questão encontra-se implementado e será tratado na seção de resultados como Método 3, na seção em questão serão apresentadas evidências numéricas que de fato o algoritmo resultante desta combinação ($q = 2$, $r = 2$ e $\chi = 1$) é de segunda ordem em relação a todas as normas consideradas.

Considerando que os métodos resultantes de $q = 2$ são métodos de três passos de tempo foi utilizada a metodologia apresentada em Villar (2007):

$$\begin{aligned} \frac{(\alpha_2 \mathbf{u}^t + \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1})}{h_t} &= \beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^t + \nabla P^{*,n+1} \\ \nabla^2 \phi^{n+1} &= \frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t \\ \mathbf{u}^{n+1} &= \mathbf{u}^t - \frac{h_t}{\alpha_2} \nabla \phi^{n+1} \\ P^{n+1} &= P^{*,n+1} + \phi^{n+1} - \frac{\chi}{\text{Re}} \nabla \cdot \mathbf{u}^t \end{aligned}, \quad (4.36)$$

onde para o primeiro passo de tempo são escolhidos $\alpha_2 = 1$, $\alpha_1 = -1$, $\alpha_0 = 0$ (equivalente a escolher $q = 1$) e para os demais $\alpha_2 = 3/2$, $\alpha_1 = -2$, $\alpha_0 = 1/2$ (equivalente a escolher $q = 2$), a escolha das constantes β_1 e β_0 vai depender de como se quer tratar os termos convectivos. Para o Método 3, Kim e Moin (1985), foram escolhidos $\beta_1 = 3/2$ e $\beta_0 = -1/2$.

Antes de encerrar esta seção é necessário observar que existem vários outros métodos de projeção. Por exemplo, pode-se inverter a ordem e resolver para pressão no passo 1 dos algoritmos citados anteriormente e resolver as velocidades no espaço dos campos vetoriais com divergente nulo (passo de projeção) dando origem aos métodos de projeção com correção nas

velocidades. Além dos métodos de correção na pressão e velocidades existem outras formas de aplicar a projeção, para mais detalhes sobre estes métodos consultar o trabalho de Guermond et al. (2006).

4.4 Coeficientes das velocidades

Com as discretizações apresentadas na seção 4.2 e os métodos de projeção apresentados nas seções 4.3.1, 4.3.2 e 4.3.3 apresentam-se agora os coeficientes das velocidades. Primeiramente deve-se isolar a velocidade auxiliar, dada pela primeira equação da Eq. (4.36).

$$\alpha_2 \mathbf{u}^t - \frac{ht}{\text{Re}} \nabla^2 \mathbf{u}^t = -\alpha_1 \mathbf{u}^n - \alpha_0 \mathbf{u}^{n-1} + ht \left(\beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \nabla P^{*,n+1} \right) = LD, \quad (4.37)$$

onde LD representa o “lado direito” da Eq. (4.37) e será utilizado como termo fonte no sistema de equações das velocidades. Os coeficientes para as velocidades dos volumes internos e os que fazem vizinhança com os volumes fictícios serão apresentados a seguir.

Deve-se lembrar que os termos βg representam os termos advectivos do tipo $\mathbf{u} \cdot \nabla \mathbf{u}$, conforme Eq. (4.22).

Reescrevendo a Eq. (4.37) no formato da Eq. (4.13)

$$\alpha_2 \mathbf{u}^t - \frac{ht}{\text{Re}} \mathbf{D}_{i,j}^t = -\alpha_1 \mathbf{u}^n - \alpha_0 \mathbf{u}^{n-1} + ht \left(\beta_1 \mathbf{A}_{i,j}^n + \beta_0 \mathbf{A}_{i,j}^{n-1} + \nabla P_{i,j}^{*,n+1} \right) = LD. \quad (4.38)$$

A seguir os termos difusivos $\mathbf{D}_{i,j}^t$ e advectivos $\mathbf{A}_{i,j}^n$ serão reescritos na ordem lexicográfica adotada, ver Fig. 2.5 para mais detalhes.

4.4.1 Termos difusivos na ordem lexicográfica

Reescrevendo a parte difusiva da Eq. (4.14) na ordem lexicográfica para a componente u^t tem-se

$$\mathbf{D}_u^t = \frac{1}{h^2} \left(u_{pu+1}^t + u_{pu-1}^t + u_{pu+Nxu}^t + u_{pu-Nxu}^t - 4u_{pu}^t \right), \quad (4.39)$$

onde pu é o índice lexicográfico da variável u , ver Fig. 2.5, e Nxu é o número de “faces” na direção coordenada x . Evidentemente $pu + Nxu$ representa u ao norte de u_{pu} .

De forma similar, o termo difusivo para a componente vertical da velocidade v^t

$$\mathbf{D}_v^t = \frac{1}{h^2} \left(v_{pv+1}^t + u_{pv-1}^t + u_{pv+Nxv}^t + v_{pv-Nxv}^t - 4u_{pv}^t \right). \quad (4.40)$$

4.4.2 Termos advectivos na ordem lexicográfica

Reescrevendo os termos advectivos $u, A_{i,j}^n$ da Eq. (4.14), como $u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y}$ (Villar, 2007) e utilizando a ordenação lexicográfica adotada, tem-se

$$\mathbf{u} \cdot \nabla u = u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = u_{pu} \frac{(u_{pu+1} - u_{pu-1})}{2h} + \left(\frac{v_{pv} + v_{pv+1} + v_{pv-Nxv} + v_{pv+1-Nxv}}{4} \right) \frac{(u_{pu+Nxu} - u_{pu-Nxu})}{2h}, \quad (4.41)$$

neste ponto sugere-se utilizar a Fig. 4.2 para familiarizar-se com os subíndices. Verificando a Fig. 4.2 observa-se que o único termo que não está definido na expressão $u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y}$ é v , logo recorre-se a uma média de v “sobre” o ponto onde está u_{pu} .

Resumindo as notações utilizadas nas Eqs. (4.22), (4.38) e (4.41) para a advecção, tem-se para a componente u

$$g(\mathbf{u}) = \mathbf{A}_{i,j} = \mathbf{u} \cdot \nabla u = u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y}. \quad (4.42)$$

De forma similar para v tem-se

$$\mathbf{u} \cdot \nabla v = u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \left(\frac{u_{pu} + u_{pu-1} + u_{pu+Nxu} + u_{pu-1+Nxu}}{4} \right) \frac{(v_{pv+1} - v_{pv-1})}{2h} + v_{pv} \frac{(v_{pv+Nxv} - v_{pv-Nxv})}{2h}. \quad (4.43)$$

4.4.3 Gradiente na ordem lexicográfica

O gradiente da pressão, primeira equação da Eq. (4.36), pode ser reescrito na ordem lexicográfica como

$$\begin{aligned} \nabla P^{*,n+1} &= \frac{\partial P^{*,n+1}}{\partial x} = \frac{P_{pp+1}^{*,n+1} - P_{pp}^{*,n+1}}{h} \\ &\quad \text{ou} \\ \nabla P^{*,n+1} &= \frac{\partial P^{*,n+1}}{\partial y} = \frac{P_{pp+Nxp}^{*,n+1} - P_{pp}^{*,n+1}}{h} \end{aligned} \quad (4.44)$$

Deve-se notar que $P^{*,n+1}$ vai depender, Eq. (4.32), do método utilizado: $P^{*,n+1} = P^n$ (Métodos 1 e 2) e $P^{*,n+1} = 2P^n - P^{n-1}$ (Método 3).

Comparando-se a Eq. (4.44) com a Fig. 4.1 percebe-se que as derivadas da Eq. (4.44) estão sendo calculadas sobre os pontos pv ou pu , ou seja, sobre as faces do volume onde se encontram as velocidades u_{pu}^{n+1} e v_{pv}^{n+1} , que está de acordo com a proposta de uso de malhas desencontradas, isto é, evitar instabilidades numéricas do tipo *odd-even* (Trottenberg et al., 2001).

É necessário observar que a Eq. (4.44) é uma aproximação da derivada da pressão de segunda ordem de acurácia pois é calculada sobre os pontos pu e pv , por exemplo

$$\frac{\partial P^{*,n+1}}{\partial x} = \frac{P^{*,n+1}(xu + \frac{h}{2}) - P^{*,n+1}(xu - \frac{h}{2})}{2\frac{h}{2}} = \frac{P_{pp+1}^{*,n+1} - P_{pp}^{*,n+1}}{h}, \quad (4.45)$$

onde xu é a posição de u_{pu} (ver Fig. 4.1).

De forma similar ao gradiente da pressão, Eq. (4.44), o gradiente da correção da pressão pode ser reescrito na ordem lexicográfica como

$$\begin{aligned}\nabla\phi^{n+1} &= \frac{\partial\phi^{n+1}}{\partial x} = \frac{\phi_{pp+1}^{n+1} - \phi_{pp}^{n+1}}{h} \\ &\quad \text{ou} \\ \nabla\phi^{n+1} &= \frac{\partial\phi^{n+1}}{\partial y} = \frac{\phi_{pp+Nxp}^{n+1} - \phi_{pp}^{n+1}}{h}\end{aligned}\quad (4.46)$$

As aproximações da Eq. (4.46) são de segunda ordem porque são calculados nos pontos pu e pv .

4.4.4 Coeficientes para as velocidades dos volumes internos

Volumes internos se referem aos volumes que não possuem volumes fictícios como vizinhos, ver Fig. 4.2. Para obter os coeficientes da velocidade u destes volumes, basta substituir a Eq. (4.39) na Eq. (4.38). A seguir, será apresentada apenas a dedução dos coeficientes para a componente horizontal da velocidade (u). A dedução para a velocidade v é análoga.

$$\begin{aligned}u_{pu}^t \left(\alpha_2 + \frac{4h_t}{Reh^2} \right) &= u_{pu+1}^t \left(\frac{h_t}{Reh^2} \right) + u_{pu-1}^t \left(\frac{h_t}{Reh^2} \right) + \\ &\quad u_{pu+Nxu}^t \left(\frac{h_t}{Reh^2} \right) + u_{pu-Nxu}^t \left(\frac{h_t}{Reh^2} \right) + LD\end{aligned}\quad (4.47)$$

Reescrevendo a Eq. (4.47) na forma

$$a_P u_{pu}^t = a_E u_{pu+1}^t + a_W u_{pu-1}^t + a_N u_{pu+Nxu}^t + a_S u_{pu-Nxu}^t + b_P, \quad (4.48)$$

tem-se que os coeficientes para as velocidades dos volumes internos e o termo fonte são

$$\begin{aligned}a_P &= \alpha_2 + \frac{4h_t}{Reh^2} \\ a_W &= a_E = \frac{h_t}{Reh^2} \\ a_S &= a_N = \frac{h_t}{Reh^2} \\ b_P &= LD\end{aligned}\quad (4.49)$$

4.4.5 Coeficientes para as velocidades dos volumes com contornos

Para obter os coeficientes da velocidade u , como antes, substitui-se a Eq. (4.39) na Eq. (4.38). No entanto é necessário levar em consideração os contornos, o que será mostrado a seguir. Será apresentada apenas a dedução dos coeficientes para a componente horizontal da velocidade u no canto sudoeste (*South-West*, SW) pois a dedução para a velocidade v e os outros contornos é análoga.

Na Fig. 4.4 as linhas tracejadas representam os volumes fictícios, ou seja, volumes que não pertencem ao domínio físico do problema. É mostrada uma velocidade, u_{pu} , que possui dois vizinhos internos u_{pu+1} e u_{pu+Nxu} , um vizinho fictício u_{pu-Nxu} e uma velocidade, u_{pu-1} , que contém uma condição de contorno prescrita u_W . Além disso, é mostrada a condição de contorno

u_S que não é considerada pelo disposição das velocidades u . Desta figura podem-se deduzir as seguintes equações

$$\begin{aligned} u_{pu-1} &= u_W \\ u_{pu-Nxu} &= 2u_S - u_{pu}, \end{aligned} \quad (4.50)$$

onde foi utilizado a extrapolação linear $\frac{u_{pu}+u_{pu-Nxu}}{2} = u_S$.

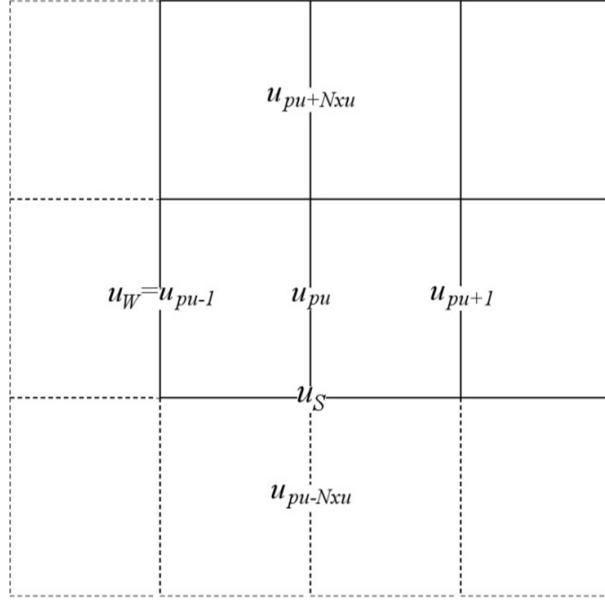


Figura 4.4: Velocidade u de um volume com vizinhos fictícios e contornos.

Substituindo as Eqs. (4.39) e (4.50) na Eq. (4.38) conduz a

$$u_{pu}^t \left(\alpha_2 + \frac{4h_t}{Reh^2} \right) = u_{pu+1}^t \left(\frac{h_t}{Reh^2} \right) + u_{pu+Nxu}^t \left(\frac{h_t}{Reh^2} \right) + LD + \left(\frac{h_t}{Reh^2} \right) (u_W + 2u_S) \quad (4.51)$$

Comparando a Eq. (4.51) com a Eq. (4.48) conclui-se que

$$\begin{aligned} a_P &= \alpha_2 + \frac{4h_t}{Reh^2} \\ a_W &= 0 \\ a_E &= \frac{h_t}{Reh^2} \\ a_S &= 0 \\ a_N &= \frac{h_t}{Reh^2} \\ b_P &= LD + \left(\frac{h_t}{Reh^2} \right) (u_W + 2u_S) \end{aligned} \quad (4.52)$$

4.5 Coeficientes da pressão

4.5.1 Termos difusivos na ordem lexicográfica

Reescrevendo o laplaciano da correção da pressão, $\nabla^2\phi$, na ordem lexicográfica

$$\mathbf{D}_\phi = \frac{1}{h^2} \left(\phi_{pp+1}^{n+1} + \phi_{pp-1}^{n+1} + \phi_{pp+Nxp}^{n+1} + \phi_{pp-Nxp}^{n+1} - 4\phi_{pp}^{n+1} \right). \quad (4.53)$$

onde pp é o índice lexicográfico da correção na pressão, variável ϕ , e Nxp é o número de volumes na direção coordenada x . O índice $pp + Nxp$ representa ϕ ao norte de ϕ_{pp} .

4.5.2 Divergente na ordem lexicográfica

O divergente na segunda equação da Eq. (4.36), discretizado na ordem lexicográfica é reescrito como

$$\nabla \cdot \mathbf{u}^t = \frac{\partial u^t}{\partial x} + \frac{\partial v^t}{\partial y} = \frac{u_{pu}^t - u_{pu-1}^t}{h} + \frac{v_{pv}^t - v_{pv-Nxp}^t}{h}. \quad (4.54)$$

Comparando-se a Eq. (4.54) com a Fig. 4.1 percebe-se que as derivadas da Eq. (4.54) estão sendo calculadas sobre o ponto pp , ou seja, sobre o centro do volume onde se encontra a pressão P_{pp}^{n+1} , que é exatamente o que se pretendia quando se utilizaram malhas desencontradas. Este arranjo de variáveis é o que evita instabilidades numéricas do tipo *odd-even* (Trottenberg et al., 2001).

De forma similar ao gradiente da pressão, Eq. (4.44), as derivadas da Eq. (4.54) são de segunda ordem porque são calculadas sobre o ponto pp .

4.5.3 Coeficientes da pressão para os volumes internos

Para obter os coeficientes dos volumes internos são possíveis duas estratégias: ou utilizar a condição de incompressibilidade ($\nabla \cdot \mathbf{u}^{n+1} = 0$) ou substituir as Eqs. (4.53) e (4.54) na segunda equação da Eq. (4.36)

$$\nabla^2\phi^{n+1} = \frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t, \quad (4.55)$$

o que, satisfaz a condição de incompressibilidade e resulta em

$$\begin{aligned} \phi_{pp}^{n+1} \left(\frac{4}{h^2} \right) &= \phi_{pp+1}^{n+1} \left(\frac{1}{h^2} \right) + \phi_{pp-1}^{n+1} \left(\frac{1}{h^2} \right) + \\ &\phi_{pp+Nxp}^{n+1} \left(\frac{1}{h^2} \right) + \phi_{pp-Nxp}^{n+1} \left(\frac{1}{h^2} \right) - \frac{\alpha_2}{h_t} \left(\frac{u_{pu}^t - u_{pu-1}^t}{h} + \frac{v_{pv}^t - v_{pv-Nxp}^t}{h} \right). \end{aligned} \quad (4.56)$$

Comparando com uma equação na forma (4.48), tem-se que os coeficientes e termo fonte para a pressão são dados por

$$\begin{aligned}
 a_P &= \frac{4}{h^2} \\
 a_W = a_E &= \frac{1}{h^2} \\
 a_S = a_N &= \frac{1}{h^2} \\
 b_P &= -\frac{\alpha_2}{h_t} \left(\frac{u_{pu}^t - u_{pu-1}^t}{h} + \frac{v_{pv}^t - v_{pv-Nxv}^t}{h} \right)
 \end{aligned} \quad . \quad (4.57)$$

4.5.4 Coeficientes da pressão para os volumes com contornos

Um dos grandes problemas das Equações de Navier-Stokes é que a pressão e as velocidades estão acopladas pela condição de incompressibilidade. Para deduzir os coeficientes nos volumes que possuem vizinhos fictícios é necessário utilizar a condição de incompressibilidade:

$$\nabla \cdot \mathbf{u}^{n+1} = \frac{u_{pu}^{n+1} - u_{pu-1}^{n+1}}{h} + \frac{v_{pv}^{n+1} - v_{pv-Nxv}^{n+1}}{h} = 0. \quad (4.58)$$

Das Eqs. (4.36) tem-se $\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{h_t}{\alpha_2} \nabla \phi^{n+1}$ que completamente discretizada vai gerar as equações

$$\begin{aligned}
 u_{pu}^{n+1} &= u_{pu}^t - \frac{h_t}{\alpha_2} \left(\frac{\phi_{pp+1}^{n+1} - \phi_{pp}^{n+1}}{h} \right) \\
 u_{pu-1}^{n+1} &= u_{pu-1}^t - \frac{h_t}{\alpha_2} \left(\frac{\phi_{pp}^{n+1} - \phi_{pp-1}^{n+1}}{h} \right) \\
 v_{pv}^{n+1} &= u_{pv}^t - \frac{h_t}{\alpha_2} \left(\frac{\phi_{pp+Nxp}^{n+1} - \phi_{pp}^{n+1}}{h} \right) \\
 v_{pv-Nxv}^{n+1} &= u_{pv-Nxv}^t - \frac{h_t}{\alpha_2} \left(\frac{\phi_{pp}^{n+1} - \phi_{pp-Nxp}^{n+1}}{h} \right)
 \end{aligned} \quad . \quad (4.59)$$

Substituindo as Eqs. (4.59) na Eq. (4.58) (condição de incompressibilidade) os coeficientes e termo fonte na Eq. (4.57) seriam obtidos para volumes que não possuem volumes fictícios como vizinhos. Para o caso de volumes com contornos será realizada a dedução dos coeficientes para a pressão no canto sudoeste (*South-West*, SW). A dedução para os outros contornos é análoga.

Na Fig. 4.5 as linhas tracejadas representam os volumes fictícios, ou seja, volumes que não pertencem ao domínio físico do problema. É mostrado um volume de pressão, P_{pp} , que possui duas faces internas u_{pu} e v_{pv} , e duas velocidades que possuem condição de contorno prescrita u_W e v_S . Da Fig. 4.5 pode-se deduzir as seguintes equações

$$\begin{aligned}
 u_{pu-1} &= u_W \\
 v_{pv-Nxv} &= v_S
 \end{aligned} \quad . \quad (4.60)$$

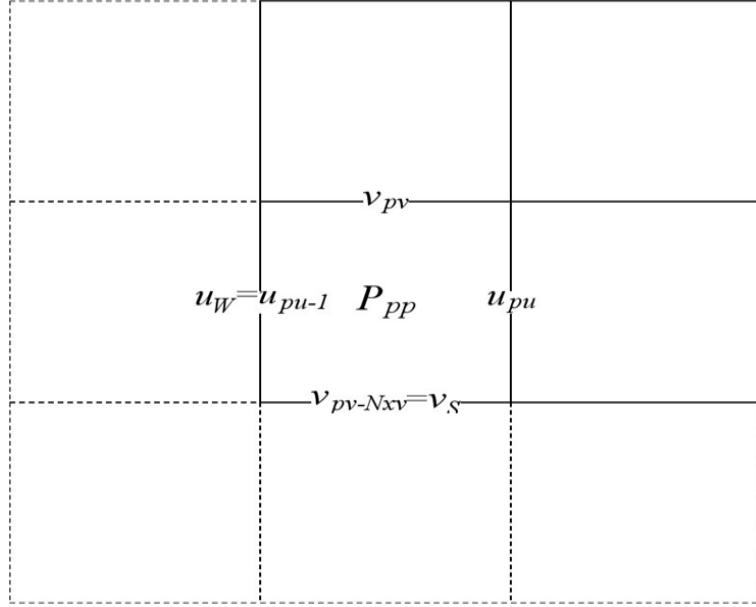


Figura 4.5: Pressão de um volume com contornos.

Substituindo as Eqs. (4.59) e (4.60) na Eq. (4.58) tem-se

$$\begin{aligned} \phi_{pp}^{n+1} \left(\frac{2}{h^2} \right) &= \phi_{pp+1}^{n+1} \left(\frac{1}{h^2} \right) + \phi_{pp+Nxp}^{n+1} \left(\frac{1}{h^2} \right) \\ &\quad - \frac{\alpha_2}{h_t} \left(\frac{u_{pu}^t - u_W}{h} + \frac{v_{pv}^t - v_S}{h} \right) \end{aligned} \quad (4.61)$$

comparando com uma equação na forma (4.48), tem-se que os coeficientes e termo fonte são dados por

$$\begin{aligned} a_P &= \frac{2}{h^2} \\ a_W &= 0 \\ a_E &= \frac{1}{h^2} \\ a_S &= 0 \\ a_N &= \frac{1}{h^2} \\ b_P &= -\frac{\alpha_2}{h_t} \left(\frac{u_{pu}^t - u_W}{h} + \frac{v_{pv}^t - v_S}{h} \right) \end{aligned} \quad (4.62)$$

Diferentemente dos coeficientes das velocidades, os coeficientes da pressão não dependem nem do número de Reynolds nem da discretização temporal. Devido a esta independência, o método *multigrid* descrito no capítulo 2 apresenta a convergência esperada de sua aplicação na resolução da equação de Poisson.

4.6 Condição de integrabilidade e estabilidade

Uma característica comum dos métodos de projeção é que estes impõem condições de contorno de Neumann homogêneas sobre a pressão nos contornos (Guermond et al., 2006),

$$\begin{aligned} \text{(Método 1)} \quad & \frac{\partial P^{n+1}}{\partial n} \Big|_{\partial\Omega} = 0 \\ \text{(Métodos 2 e 3)} \quad & \frac{\partial P^{n+1}}{\partial n} \Big|_{\partial\Omega} = \frac{1}{Re} (\nabla \times \nabla \times \mathbf{u}^{n+1}) \cdot \mathbf{n} \Big|_{\partial\Omega} \end{aligned} \quad (4.63)$$

Devido a estas condições de contorno, a seguinte condição de integrabilidade precisa ser satisfeita para garantir a existência e unicidade da solução (Briggs et al., 2000):

$$\int_{\Omega} \nabla^2 \phi^{n+1} = \int_{\Omega} \frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t = 0. \quad (4.64)$$

Para satisfazer a Eq. (4.64) é aplicado a cada passo de tempo o procedimento utilizado por Villar (2007):

1. Calcular numericamente a integral de $\frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t$ obtendo uma constante.
2. Subtrair essa constante da correção da pressão ϕ^{n+1} .

Nas seções anteriores foi demonstrado que o termo $\frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t$ é basicamente o termo fonte b_P da pressão (ver Eqs. (4.55) e (4.57)) que é conhecido antes de se resolver a equação de Poisson no segundo passo dos Métodos 1, 2 e 3. No presente trabalho, a Eq. (4.64) é calculada utilizando a regra do retângulo

$$\int_{\Omega} \frac{\alpha_2}{h_t} \nabla \cdot \mathbf{u}^t = \sum_{pp=1}^{NxNy} b_{pp} h^2 \quad (4.65)$$

onde $NxNy$ representa o total de volumes da pressão e b_{pp} o termo fonte do volume pp .

Foi observado nas simulações realizadas neste trabalho que a condição de integrabilidade é necessária para se conhecer o valor “real” da pressão a cada passo de tempo. Se a condição de integrabilidade não for utilizada, a solução numérica das velocidades não será afetada. No entanto, a solução da pressão será do tipo $P + C$, onde C é uma constante, ou seja, a pressão obtida será o valor “real” mais uma constante.

Se o passo de tempo h_t for muito grande em relação ao refinamento espacial (h_x e h_y), as simulações podem se tornar instáveis (Kalland, 2008). O critério utilizado neste trabalho para evitar instabilidades é dado por (Maliska, 2004; Kalland, 2008):

$$h_t < \frac{1}{2Re} \frac{h_x^2 h_y^2}{h_x^2 + h_y^2}. \quad (4.66)$$

Como foi considerado $h_x = h_y = h$ neste trabalho, o critério pode ser reescrito como

$$h_t < \frac{h^2}{4Re}. \quad (4.67)$$

Em Kalland (2008) são apresentados outros critérios que auxiliam a estabilidade numérica.

4.7 Algoritmo do método de projeção sequencial

Embasado nas seções anteriores, o algoritmo do método de projeção pode ser dado por

Algoritmo 3 Método de projeção sequencial

- 1: Inicializar variáveis: $Ti, Tf, Lx, Ly, Nx, Ny, Nt$ (número de passos de tempo), hx, hy, ht e Re .
 - 2: Calcular os coeficientes das velocidades e pressão. Ver seções 4.4.4, 4.4.5, 4.5.3 e 4.5.4
 - 3: Aplicar condições de contorno e iniciais para as velocidades e condições iniciais para a pressão.
 - 4: **para** $n = 0, 1, 2, \dots, Nt - 1$ **faca**
 - 5: Definir as constantes $\alpha_2, \alpha_1, \alpha_0, \beta_1, \beta_0$ da Eq. (4.36).
 - 6: Atualizar condições de contorno das velocidades†.
 - 7: Calcular $LD = -\alpha_1 \mathbf{u}^n - \alpha_0 \mathbf{u}^{n-1} + ht \left(\beta_1 \mathbf{A}_{i,j}^n + \beta_0 \mathbf{A}_{i,j}^{n-1} + \nabla P_{i,j}^{*,n+1} \right)$ da Eq. (4.38) onde os $\mathbf{A}_{i,j}$ são dados pelas Eqs. (4.41) e (4.43) e $\nabla P_{i,j}^{*,n+1}$ dado pela Eq. (4.44)††.
 - 8: Resolver $\alpha_2 \mathbf{u}^t - \frac{ht}{Re} \mathbf{D}_{i,j}^t = LD$, onde $\mathbf{D}_{i,j}^t$ são dados pelas Eqs. (4.39) e (4.40).
 - 9: Resolver a equação de Poisson $\nabla^2 \phi^{n+1} = \frac{\alpha_2}{ht} \nabla \cdot \mathbf{u}^t$.
 - 10: Remover sua constante de integração da correção da pressão $\phi^{n+1} - \int_{\Omega} \frac{\alpha_2}{ht} \nabla \cdot \mathbf{u}^t$. Ver Eq. (4.65).
 - 11: Atualizar velocidades anteriores com $\mathbf{u}^n = \mathbf{u}^{n+1}$.
 - 12: Corrigir as velocidades com $\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{h_t}{\alpha_2} \nabla \phi^{n+1}$.
 - 13: Corrigir a pressão com $P^{n+1} = P^{*,n+1} + \phi^{n+1} - \frac{X}{Re} \nabla \cdot \mathbf{u}^t$.
 - 14: Atualizar pressão anterior com $P^n = P^{n+1}$.
 - 15: **fim para**
-

† Lembrando que estas condições variam a cada passo de tempo e são obtidas da solução analítica para o problema 1. Para o problema 2 estas condições são idênticas para todos os passos de tempo.

†† Para casos $n - 1 < 0$ ($\mathbf{u}^{n-1}, \mathbf{A}_{i,j}^{n-1}, P^{n-1}$) utiliza-se o valor destas variáveis no instante inicial $n = 0$.

5 Métodos de projeção em paralelo

Neste capítulo os métodos de projeção serão modificados para que permitam processamento paralelo ao longo do tempo. Como a correção da pressão não permite processamento ao longo do tempo nela será utilizado processamento paralelo em relação ao espaço. Desta forma o método de projeção proposto neste trabalho apresentará paralelização temporal e espacial.

5.1 Método de projeção sequencial de referência

Como o objetivo deste trabalho é modificar os métodos de projeção para que permitam paralelização em relação ao tempo, optou-se por utilizar um método de projeção mais simples de referência obtido pela combinação de parâmetros $q = 1$, $r = 1$ e $\chi = 1$ na generalização dos métodos de projeção proposta por Guermond et al. (2006) discutida na seção 4.3.3. Para esta combinação tem-se o método de projeção dado por

1. Primeiro passo:

$$\frac{\mathbf{u}^t - \mathbf{u}^n}{h_t} = \beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^t + \nabla P^n, \\ \mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n \quad (5.1)$$

2. Segundo passo:

$$\nabla^2 \phi^{n+1} = \frac{1}{h_t} \nabla \cdot \mathbf{u}^t, \quad (5.2)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - h_t \nabla \phi^{n+1}, \quad (5.3)$$

$$P^{n+1} = P^n + \phi^{n+1} - \frac{1}{\text{Re}} \nabla \cdot \mathbf{u}^t, \quad (5.4)$$

onde βg , com β constante, representa os termos advectivos $\mathbf{u} \cdot \nabla \mathbf{u}$.

O principal motivo de se escolher aqui um método de projeção mais simples e não outros mais complexos, como o Método 3 por exemplo, foi para evitar dificuldades associadas à solução das velocidades (Passo 1) para todos os passos de tempo em paralelo com o método *multigrid*. A aplicação das ideias deste capítulo em métodos de projeção mais acurados (pelo menos segunda ordem nas velocidades e pressão), como o Método 3, será realizada em trabalhos futuros.

Para este método, as velocidades são de segunda ordem em relação ao tempo para as normas l_1 , l_2 e l_∞ e a pressão é de primeira ordem (Guermond et al., 2006).

O método de projeção apresentado é resolvido como ilustrado no Algoritmo 4 onde n representa o passo de tempo e i a iteração do *solver*:

Algoritmo 4 Método de projeção sequencial de referência

- 1: $\mathbf{u}^{-1} = \mathbf{u}^0$ = Condições iniciais de u e v .
 - 2: P^0 = Condições iniciais de P
 - 3: N_t = Último passo de tempo da simulação.
 - 4: **para** $n = 0, 1, 2, \dots, N_t - 1$ **faca**
 - 5: **Passo 1:** Resolver \mathbf{u}^t com
 - 6: $\frac{h_t}{\text{Re}} \nabla^2 \mathbf{u}^t - \mathbf{u}^t = -\mathbf{u}^n - h_t(\beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \nabla P^n)$
 - 7: **Passo 2:** Resolver ϕ^{n+1} com
 - 8: $\nabla^2 \phi^{n+1} = \frac{1}{h_t} \nabla \cdot \mathbf{u}^t$
 - 9: Atualizar $\mathbf{u}^{n+1} = \mathbf{u}^t - h_t \nabla \phi^{n+1}$
 - 10: Atualizar $P^{n+1} = P^n + \phi^{n+1} - \frac{1}{\text{Re}} \nabla \cdot \mathbf{u}^t$
 - 11: **fim para**
-

Nota-se que no Algoritmo 4 as Eqs. (5.1) e (5.2) são resolvidas até o critério de parada ser atingido.

5.2 Métodos de projeção paralelizados no tempo

Nesta seção propõe-se uma modificação no método de projeção sequencial de referência (seção 5.1) para que a Eq. (5.2) possa ser calculada em paralelo no tempo, isto é, a Eq. (5.2) será calculada para cada passo de tempo.

Primeiramente serão unificados os campos de velocidades \mathbf{u}^t e \mathbf{u}^n . Para isto, basta substituir a Eq. (5.3) na Eq. (5.1) e em seguida substituir \mathbf{u}^n por $\tilde{\mathbf{u}}^n$ para caracterizar que se está trabalhando no espaço das projeções.

1. **Primeiro passo:**

$$\frac{h_t}{\text{Re}} \nabla^2 \tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1} = -\tilde{\mathbf{u}}^n - h_t [\beta_1 g_1(\tilde{\mathbf{u}}^n) + \beta_0 g_0(\tilde{\mathbf{u}}^{n-1}) - \nabla(P^n - \phi^n)], \quad (5.5)$$

$$\tilde{\mathbf{u}}^{n+1}|_{\partial\Omega} = \mathbf{b}^{n+1}$$

2. **Segundo passo:**

$$\nabla^2 \phi^{n+1} = \frac{1}{h_t} \nabla \cdot \tilde{\mathbf{u}}^{n+1}, \quad (5.6)$$

$$P^{n+1} = P^n + \phi^{n+1} - \frac{1}{\text{Re}} \nabla \cdot \tilde{\mathbf{u}}^{n+1}. \quad (5.7)$$

Em Borregales et al. (2018) é apresentada uma inovadora abordagem para resolver as equações de Biot de forma paralela no tempo. Borregales et al. (2018) aplicam a paralelização no tempo ao método *fixed-stress* (Gaspar e Rodrigo, 2017). Neste trabalho propõe-se aplicar tal paralelização ao método de projeção sequencial.

Analizando o método de projeção com a velocidade $\tilde{\mathbf{u}}^n$, Eqs. (5.5-5.7), pode-se notar que a Eq. (5.6) é temporalmente independente e pode ser calculada em paralelo para cada passo de tempo desde que $\tilde{\mathbf{u}}^n$ exista.

Neste trabalho propõe-se calcular de forma iterativa aproximações de $\tilde{\mathbf{u}}^n$ para todos os passos de tempo $n = 0, 1, 2, 3, \dots, N_t - 1$. Com tais aproximações, pode-se calcular as aproximações de ϕ^n . Como as aproximações de ϕ^n podem ser utilizadas para calcular a pressão P^n , um processo de sucessivas melhorias em $\tilde{\mathbf{u}}^n$ e ϕ^n pode ser repetido até que algum critério de parada seja atingido. No final do processo iterativo as velocidades nas faces (\tilde{u}^n e \tilde{v}^n) e a pressão no centro dos volumes (P^n) estarão definidas para todos os passos de tempo, $n = 0, 1, 2, 3, \dots, N_t$, considerados.

A Eq. (5.5) pode ser calculada utilizando uma abordagem similar à discutida em Friedhoff et al. (2013), ou seja, considerando cada passo de tempo $n = 0, 1, 2, 3, \dots, N_t - 1$, pode-se escrever o sistema linear

$$A\tilde{\mathbf{u}} = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & & \\ & & -\Phi & I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}^0 \\ \tilde{\mathbf{u}}^1 \\ \vdots \\ \tilde{\mathbf{u}}^{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{g}^0 \\ \mathbf{g}^1 \\ \vdots \\ \mathbf{g}^{N_t-1} \end{bmatrix} = \mathbf{g}. \quad (5.8)$$

onde os termos Φ e \mathbf{g} serão obtidos a seguir.

Para $n = 0, 1, 2, 3, \dots, N_t - 1$ a Eq. (5.8) é equivalente à

$$\begin{aligned} \tilde{\mathbf{u}}^{n+1} &= \Phi\tilde{\mathbf{u}}^n + \mathbf{g}^n \\ \tilde{\mathbf{u}}^0 &= \text{condição inicial de } \tilde{\mathbf{u}} \end{aligned} \quad (5.9)$$

A discretização espacial por volumes finitos, seção 4.2, da Eq. (5.5) por ser escrita como

$$\left(\frac{h_t}{\text{Re}} D - I \right) \tilde{\mathbf{u}}^{n+1} = -\tilde{\mathbf{u}}^n - h_t \left[\beta_1 \mathbf{A}_{i,j}^n + \beta_0 \mathbf{A}_{i,j}^{n-1} - \nabla P_{i,j}^n + \nabla \phi_{i,j}^n \right] \quad (5.10)$$

onde I é a matriz identidade, os termos $\mathbf{A}_{i,j}^n$ e $\nabla P_{i,j}^n$ são dados pelas Eqs. (4.14). A discretização espacial do termo $\nabla \phi^n$ é similar à discretização de ∇P^n . O termo D é a matriz da discretização espacial por volumes finitos de ∇^2 tal que para um volume (i, j) no interior do domínio têm-se

$$(D\tilde{\mathbf{u}}^n)_{i,j} = \frac{1}{h^2} \left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n \right) \quad (5.11)$$

Para volumes com contornos $(D\tilde{\mathbf{u}}^n)_{i,j}$ pode ser obtida de acordo com a discussão apresentada na seção 4.4.5.

Isolando $\tilde{\mathbf{u}}^{n+1}$ na Eq. (5.10)

$$\tilde{\mathbf{u}}^{n+1} = - \left(\frac{h_t}{\text{Re}} D - I \right)^{-1} \tilde{\mathbf{u}}^n - \left(\frac{h_t}{\text{Re}} D - I \right)^{-1} h_t \left[\beta_1 \mathbf{A}_{i,j}^n + \beta_0 \mathbf{A}_{i,j}^{n-1} - \nabla P_{i,j}^n + \nabla \phi_{i,j}^n \right] \quad (5.12)$$

Os termos Φ e \mathbf{g} da Eq. (5.8) são obtidos comparando-se as Eqs. (5.12) e (5.9), ou seja

$$\begin{aligned} \Phi &= - \left(\frac{h_t}{\text{Re}} D - I \right)^{-1} \\ \mathbf{g} &= - \left(\frac{h_t}{\text{Re}} D - I \right)^{-1} h_t \left[\beta_1 \mathbf{A}_{i,j}^n + \beta_0 \mathbf{A}_{i,j}^{n-1} - \nabla P_{i,j}^n + \nabla \phi_{i,j}^n \right] \end{aligned} \quad (5.13)$$

O sistema da Eq. (5.8) é temporalmente sequencial, pois a velocidade no passo de tempo n , $\tilde{\mathbf{u}}^n$, depende da velocidade no passo de tempo anterior $n - 1$, $\tilde{\mathbf{u}}^{n-1}$, desta forma dois passos

de tempo consecutivos não podem ser processados simultaneamente. No entanto, é possível calcular o sistema da Eq. (5.8) em paralelo no tempo dividindo-se o domínio em nós *red* e *black*. Os detalhes de como este processo é realizado serão discutidos na próxima seção.

O Algoritmo 5 apresenta a proposta deste trabalho. Como são calculadas aproximações para todos os passos de tempo foi necessária a introdução de um ciclo externo (variável k do algoritmo 5) para garantir a convergência do processo iterativo. No algoritmo 5, n representa o passo de tempo considerado e a variável k a iteração do ciclo externo. As variáveis $\tilde{\mathbf{u}}$ e P são inicializadas com $\tilde{\mathbf{u}}^0$ e P^0 em todos os passos de tempo.

Algoritmo 5 Método de projeção paralelizado no tempo

- 1: \mathbf{u}^0 = Condições iniciais de u e v
- 2: P^0 = Condições iniciais de P
- 3: **para** $k = 0, 1, 2, 3 \dots$ **faça**
- 4: **Passo 1:** dado $\tilde{\mathbf{u}}^{n,k-1}, \tilde{\mathbf{u}}^{n-1,k-1}, \phi^{n,k-1}$ e $P^{n,k-1}$
- 5: Suavizar
- 6: **para** $n = 0, 1, 2, 3, \dots, N_t - 1$ **faça**
- 7: $\frac{h_t}{\text{Re}} \nabla^2 \tilde{\mathbf{u}}^{n+1,k} - \tilde{\mathbf{u}}^{n+1,k} = -\tilde{\mathbf{u}}^{n,k} - h_t(\beta_1 g_1(\tilde{\mathbf{u}}^{n,k-1}) + \beta_0 g_0(\tilde{\mathbf{u}}^{n-1,k-1}) - \nabla(P^{n,k-1} - \phi^{n,k-1}))$
- 8: **fim para**
- 9: **Passo 2:** dado $\tilde{\mathbf{u}}^{n+1,k}$ calcule $\phi^{n+1,k}$ com
- 10: Suavizar
- 11: **para** $n = 1, 2, 3, \dots, N_t - 1$ **faça**
- 12: $\nabla^2 \phi^{n+1,k} = \frac{1}{h_t} \nabla \cdot \tilde{\mathbf{u}}^{n+1,k}$
- 13: **fim para**
- 14: Atualizar $P^{n+1,k}$ com
- 15: **para** $n = 1, 2, 3, \dots, N_t - 1$ **faça**
- 16: $P^{n+1,k} = P^{n,k} + \phi^{n+1,k} - \frac{1}{\text{Re}} \nabla \cdot \tilde{\mathbf{u}}^{n+1}$
- 17: **fim para**
- 18: Verifica algum critério de parada
- 19: **fim para**

No Algoritmo 4 (Projeção de referência) o campo de velocidades \mathbf{u} e a correção na pressão ϕ são resolvidas até que algum critério de parada seja atingido. Resolvidos os campos \mathbf{u} e ϕ , o método de projeção avança um passo tempo. Este processo se repete até que a solução no passo de tempo N_t seja atingido. Naturalmente, o avanço no tempo nesta abordagem é sequencial e não pode ser resolvido em paralelo no tempo, no entanto é possível empregar *solvers* paralelizados ou paralelização espacial via decomposição do domínio (Neundorf et al., 2016).

Diferentemente da versão sequencial, o Algoritmo 5 não resolve os campos \mathbf{u} e ϕ , mas calcula aproximações destas variáveis. Como são calculadas aproximações para todos os passos de tempo, foi necessária a introdução de um ciclo externo (variável k do Algoritmo 5) para garantir a convergência do processo iterativo. No entanto, como pode-se observar no passo 2 do Algoritmo 5, aproximações para a correção da pressão ϕ podem ser calculadas de forma totalmente independente para todos os passos de tempo.

Apesar do passo 1 do Algoritmo 5 ser sequencial no tempo, devido às variáveis $\tilde{\mathbf{u}}^{n+1,k}$ e $\tilde{\mathbf{u}}^{n,k}$, a solução do sistema representado pela Eq. (5.8) pode ser realizada em paralelo no tempo desde que o domínio seja dividido em volumes *red* e *black*. Basicamente todos os volumes *red* de todos os passos de tempo são resolvidos em paralelo repetindo-se o processo para os volumes *black*. Isto é possível porque os volumes *red* e *black* alternam tanto no espaço quanto no tempo, isto é, volumes *red* em um passo de tempo serão volumes *black* em outro passo de tempo.

5.3 Paralelização temporal no método *multigrid*

Para se resolver em paralelo o sistema linear no tempo, dado pela Eq. (5.8), pode-se dividir o domínio em volumes *red* e *black*. A Fig. 5.1 ilustra dois passos de tempo consecutivos divididos em volumes *red* (círculos) e *black* (quadrados).

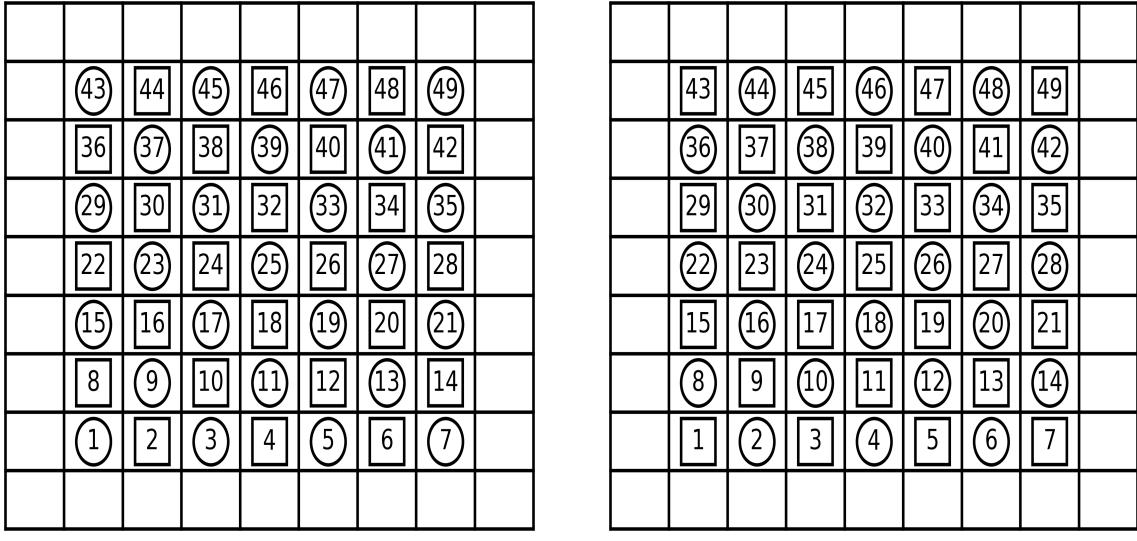


Figura 5.1: Domínio *red-black* para dois passos de tempo consecutivos.

O detalhe mais importante na Fig. 5.1 é o fato de que todos os vizinhos de um volume *black* são volumes *red* tanto no passo de tempo considerado (n) quanto nos passos de tempo anterior ($n - 1$) e posterior ($n + 1$). Desta forma pode-se resolver em paralelo no tempo o domínio considerado na Fig. 5.1.

Por exemplo: considerando-se duas *threads* resolvendo o domínio da Fig. 5.1. Como pretende-se que seja resolvido em paralelo no tempo, uma *thread* irá se encarregar do domínio dado pela Fig. 5.1(a) e a outra do domínio dado pela Fig. 5.1(b). Considerando que as duas *threads* irão resolver apenas os volumes *black* e depois apenas os volumes *red*, não existirão casos de leitura e escrita de informações de um volume simultaneamente por diferentes *threads*. Portanto, o método *multigrid* pode ser resolvido em paralelo no tempo também.

No Algoritmo 6 é apresentada uma versão simplificada do método *multigrid* paralelizado no tempo. Nele considera-se que cada *thread* está resolvendo um passo de tempo n . Além disso considera-se que existem $Nt - 1$ *threads* resolvendo $Nt - 1$ passos de tempo, ou seja, um passo de tempo apenas por *thread*.

5.4 Paralelização espacial da atualização da pressão

A Eq. (5.7) não pode ser resolvida em paralelo no tempo, além disso a divisão *red-black* discutida na seção 5.3 não pode ser empregada. No entanto pode-se observar que cada volume da Eq. (5.7) depende apenas do volume no passo de tempo anterior, desta forma pode-se partitionar o domínio em subdomínios que serão processados em paralelo.

Neste trabalho o domínio é partitionado de acordo com os conceitos discutidos na seção 2.2.1, isto é, o domínio é partitionado com

Algoritmo 6 Método *multigrid* paralelizado no tempo

1: *Início do paralelismo*: Todos os passos do algoritmo são realizados simultaneamente por cada *thread*.

2: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

3: Atualizar o valor inicial de $\tilde{\mathbf{u}}$.

4: **fim para**

5: Barreira: Esperar que todas as *threads* atinjam este ponto.

6: **para** $\text{ciclo} = 1, 2, 3, \dots, NcV$ (NcV = Número máximo de ciclos V permitidos) **faça**

7: **para** $\text{level} = 1, 2, 3, \dots, L - 1$ (L = Número máximo de níveis de malha) **faça**

8: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

9: Atualizar o termo fonte de $\tilde{\mathbf{u}}$

10: **fim para**

11: Barreira: Esperar que todas as *threads* atinjam este ponto.

12: **para** $\nu_1 = 1, 2, 3, \dots, 6$ (iterações do *solver*) **faça**

13: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

14: Aplicar Gauss-Seidel apenas nos volumes *Red*

15: **fim para**

16: Barreira: Esperar que todas as *threads* atinjam este ponto.

17: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

18: Aplicar Gauss-Seidel apenas nos volumes *Black*

19: **fim para**

20: Barreira: Esperar que todas as *threads* atinjam este ponto.

21: **fim para**

22: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

23: Calcular o resíduo

24: **fim para**

25: Barreira: Esperar que todas as *threads* atinjam este ponto.

26: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

27: Restringir o resíduo

28: **fim para**

29: Barreira: Esperar que todas as *threads* atinjam este ponto.

30: **fim para**

31: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

32: Resolver na malha mais grossa

33: **fim para**

34: Barreira: Esperar que todas as *threads* atinjam este ponto.

35: **para** $\text{level} = L, \dots, 2$ **faça**

36: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

37: Prolongar a solução

38: **fim para**

39: Barreira: Esperar que todas as *threads* atinjam este ponto.

40: **para** $\nu_2 = 1, 2, 3, \dots, 6$ (iterações do *solver*) **faça**

41: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

42: Aplicar Gauss-Seidel apenas nos volumes *Red*

43: **fim para**

44: Barreira: Esperar que todas as *threads* atinjam este ponto.

45: **para** $n = 1, 2, 3, \dots, N_t - 1$ (Realizado em paralelo por cada *thread*) **faça**

46: Aplicar Gauss-Seidel apenas nos volumes *Black*

47: **fim para**

48: Barreira: Esperar que todas as *threads* atinjam este ponto.

49: **fim para**

50: **fim para**

51: **fim para**

$$\begin{aligned} i_t &= (t - 1) N_{sub} + 1 + C(t - t_c) \\ f_t &= t N_{sub} + C(t - t_c + 1) \end{aligned}$$

em que i_t representa o início e f_t representa o fim de cada subdomínio. O termo C é calculado com

$$C = \begin{cases} 1 & \text{se } t \geq t_c \\ 0 & \text{se } t < t_c \end{cases},$$

e t_c é dado por

$$t_c = \bar{T} + 1 - \left[N_x N_y - \bar{T} N_{sub} \right].$$

N_{sub} pode ser entendido como o número de volumes por thread (\bar{T}), ou

$$N_{sub} = \left\lfloor \frac{N_x N_y}{\bar{T}} \right\rfloor.$$

Considerando que cada subdomínio ilustrado na Fig. 2.14 pertence a uma *thread*, a paralelização da atualização da pressão pode ser realizada de forma com que cada *thread* calcule a Eq. (5.7) apenas para os volumes que pertencem ao seu subdomínio do segundo passo de tempo até N_t . Apesar do processo ser serial no tempo, o processamento será realizado em paralelo no espaço devido à divisão em subdomínios.

O Algoritmo 7 ilustra como este processo é realizado em paralelo.

Algoritmo 7 Atualização da pressão em paralelo via particionamento do domínio em subdomínios

- 1: *Início do paralelismo*
 - 2: **para** $n = 1, 2, 3, \dots, N_t - 1$ **faça**
 - 3: **para** volume $= i_t, \dots, f_t$ (Realizado em paralelo por cada *thread*) **faça**
 - 4: $P^{n+1} = P^n + \phi^{n+1} - \frac{1}{Re} \nabla \cdot \tilde{\mathbf{u}}^{n+1}$
 - 5: **fim para**
 - 6: **fim para**
 - 7: Barreira: Esperar que todas as *threads* atinjam este ponto.
-

5.5 Métodos de projeção paralelizados no tempo e no espaço

Na seção 5.2 foi discutido como os métodos de projeção podem ser modificados para que a Eq. (5.2) possa ser calculada em paralelo no tempo. Na seção 5.3 foi apresentado como o método *multigrid* pode ser utilizado para resolver em paralelo no tempo o sistema da Eq. (5.8). Apesar da Eq. (5.7) não permitir paralelização temporal foi exposto na seção 5.4 como a paralelização pode ser realizada em relação ao espaço utilizando o particionamento do domínio em subdomínios.

Combinando os Algoritmos 5, 6 e 7, apresentados nas seções anteriores, obtém-se o Algoritmo 8, isto é, a versão paralelizada no tempo e no espaço do Algoritmo 4.

Algoritmo 8 Método de projeção paralelizado no tempo e no espaço

- 1: \mathbf{u}^0 = Condições iniciais de u e v
 - 2: P^0 = Condições iniciais de P
 - 3: *Início do paralelismo*: Todos os passos do algoritmo são realizados simultaneamente por cada *thread*.
 - 4: **para** $k = 0, 1, 2, 3 \dots$ **faça**
 - 5: **Passo 1:** Suavizar a Eq. (5.8) utilizando o método *multigrid* paralelizado no tempo apresentado no Algoritmo 6.
 - 6: Barreira: Esperar que todas as *threads* atinjam este ponto.
 - 7: **Passo 2:** Em paralelo, suavizar a Eq. (5.6) para cada passo de tempo $n = 1, 2, 3, \dots, N_t - 1$, utilizando o método *multigrid* apresentado no Algoritmo 2.
 - 8: Barreira: Esperar que todas as *threads* atinjam este ponto.
 - 9: Atualizar a Eq. (5.7) utilizando o Algoritmo 7 (paralelização espacial).
 - 10: Barreira: Esperar que todas as *threads* atinjam este ponto.
 - 11: Verifica algum critério de parada
 - 12: **fim para**
-

6 Resultados

Neste capítulo serão discutidos os resultados obtidos no presente trabalho. Serão apresentados testes para validar a implementação do código, bem como a acurácia das soluções numéricas. Serão apresentados resultados para cinco algoritmos implementados: Método 1 (dado na seção 4.3.1), resultado da combinação $q = 2, r = 1$ e $\chi = 0$, conhecido como método de correção incremental na pressão na forma padrão. O Método 2 (dado na seção 4.3.2), resultado da combinação $q = 2, r = 1$ e $\chi = 1$, é conhecido como método de correção incremental na pressão na forma rotacional. O Método 3 (dado na seção 4.3.3), resultado da combinação $q = 2, r = 2$ e $\chi = 1$ também conhecido como correção incremental na pressão na forma rotacional. O método de projeção de referência, Algoritmo 4, é resultado da combinação $q = 1, r = 1$ e $\chi = 1$. E finalmente, o método de projeção paralelizado no tempo, dado pelo Algoritmo 5. Todas as soluções numéricas aqui apresentadas foram obtidas com o método *multigrid*, descrito no capítulo 2 e seu formato em paralelo detalhado no capítulo 5, aplicado tanto nas velocidades como na resolução da pressão.

6.1 Dados de implementação

Nesta seção são apresentados os parâmetros empregados no método *multigrid* padrão utilizado neste trabalho.

O tipo de ciclo utilizado é o ciclo V.

O processo de restrição é feito empregando-se as Eqs. (2.2) e (2.3). O processo de prolongação é realizado empregando-se as Eqs. (2.4), (2.5) e (2.6).

A razão de engrossamento das malhas (r) é a razão de engrossamento padrão, ou seja, $r = 2$ (Briggs et al., 2000).

Tanto no método *singlegrid* (solução obtida pelo *solver* utilizando-se apenas uma malha) quanto no método *multigrid*, o *solver* utilizado foi o método de Gauss-Seidel *red-black*.

Neste trabalho, o número de iterações internas empregadas na pré-suavização, ν_1 , é igual ao número de iterações internas utilizadas na pós-suavização, ν_2 .

Com o Algoritmo 3 são resolvidos os Métodos 1, 2, 3 e com o Algoritmo 4 é resolvido o método de projeção de referência. Para estes algoritmos foi utilizado $\nu_1 = \nu_2 = 3$ no método *multigrid* aplicado nas velocidades e pressão. Estes parâmetros foram obtidos através de um estudo de otimização de parâmetros.

No caso do Algoritmo 5, o método *multigrid* que resolve as velocidades é aplicado ao longo de todos os passos de tempo considerados, ver Eq. (5.8). Para esta aplicação do *multigrid* utilizou-se $\nu_1 = \nu_2 = 6$. Para o *multigrid* aplicado na pressão utilizou-se $\nu_1 = \nu_2 = 3$. Estes parâmetros foram obtidos através de um estudo de otimização de parâmetros.

Os critérios de parada para os Algoritmos 3 e 4 são diferentes dos critérios para o Algoritmo 5.

No caso dos Algoritmos 3 e 4 a cada passo de tempo são resolvidas numericamente, com *multigrid* ou *singlegrid*, as velocidades e a pressão. O critério de parada usado para interromper o processo iterativo de cada uma das variáveis (velocidades e pressão) é a norma euclidiana do resíduo adimensionalizada pela norma euclidiana do resíduo na estimativa inicial (Briggs et al., 2000; Trottenberg et al., 2001), ou seja, $\|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(0)}\|_2$, onde o termo $\mathbf{r}^{(i)}$ é o resíduo na iteração i e $\mathbf{r}^{(0)}$, o resíduo na estimativa inicial. Entre os trabalhos que usam essa norma pode-se citar: Briggs et al. (2000); Oliveira et al. (2012); Trottenberg et al. (2001). O processo iterativo é interrompido quando a norma é menor ou igual à tolerância $\varepsilon_1 > 0$, isto é $\|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(0)}\|_2 \leq \varepsilon_1$. Neste trabalho foi utilizado $\varepsilon_1 = 10^{-9}$.

Como existe um ciclo externo no Algoritmo 5 (representado pela variável k) o critério de parada usado para interromper o processo iterativo deve ser realizado em relação a este ciclo. Neste trabalho o critério utilizado foi $\max|\mathbf{u}(N_t)^k - \mathbf{u}(N_t)^{k-1}| \leq \varepsilon_2$, isto é, para alguma iteração externa o valor máximo da diferença em módulo das velocidades u e v na iteração k e $k-1$ (iteração externa anterior) no último passo de tempo considerado, N_t , deve ser menor que alguma tolerância (ε_2). Neste trabalho utilizou-se $\varepsilon_2 = 10^{-11}$. Para todos os experimentos realizados com o Algoritmo 5 o critério $\max|\mathbf{u}(N_t)^k - \mathbf{u}(N_t)^{k-1}| \leq \varepsilon_2 = 10^{-11}$ mostrou-se suficiente para obter resultados similares aos do Algoritmo 4. Outros critérios de parada para o Algoritmo 5 serão investigados em trabalhos futuros.

Em todas as simulações, o método *multigrid* partiu da malha mais fina, nível 1, e foi até a malha mais grossa possível, ou seja, utilizou-se L_{\max} .

Para avaliar o fator *speedup* (aceleração em relação ao processamento serial do algoritmo) devido à paralelização é necessário medir o tempo de CPU (TCPU). Este tempo foi medido em segundos (s) com o uso da função `OMP_GET_WTIME()`. Entende-se por TCPU o tempo gasto para realizar a geração de malhas, atribuir a estimativa inicial, calcular os coeficientes e resolver os sistemas lineares até que atinjam a tolerância estabelecida. Na avaliação do paralelismo foram utilizados de 1 a 16 *threads* até atingir a tolerância estabelecida. Em todos os experimentos foi considerado um passo de tempo por *thread*, ou seja, foram resolvidos 16 passos de tempo.

Os algoritmos foram implementados na linguagem Fortran 2003 utilizando o compilador GNU Fortran (GFortran), versão 4.6.3, com a opção OpenMP e precisão dupla.

Os testes com 16 passos de tempo foram realizados em um servidor disponibilizado pelo Sistema Meteorológico do Paraná-SIMEPAR. O servidor utilizado está equipado com processadores Intel (R) Xeon (R) CPU E5-2660 v3 de 2.60GHz e totalizando 40 *threads*, das quais 16 foram utilizadas para testes; 32 GB de memória RAM e sistema operacional Linux 64 bits.

6.2 Avaliação dos métodos de projeção sequenciais

Para avaliação do erro numérico foi utilizado o Problema 1 (Vórtices de Taylor-Green) que possui solução analítica conhecida.

É importante observar que neste capítulo será utilizado o termo “ordem efetiva do erro” para referir-se à ordem do erro associada à discretização espacial e “taxa de convergência” para referir-se à ordem do erro associada à discretização temporal dos métodos testados.

Para a discretização espacial dos métodos será avaliada se a ordem efetiva do erro (P_E) tende para a ordem assintótica (P_L). No caso da discretização temporal será verificado se as taxas de convergência, calculadas a partir do melhor ajuste linear dos valores das normas calculados com o método dos mínimos quadrados, estão de acordo com as ordens teóricas apresentadas em Guermond et al. (2006).

A quantidade $E_h = \Phi - \phi$, onde Φ representa a solução analítica de uma variável e ϕ a solução numérica, é conhecida como erro de discretização (E_h) ou erro numérico (Ferziger e Peric, 2001).

Nas seções seguintes será verificado se a ordem efetiva do erro (P_E) tende para a ordem assintótica (P_L) à medida que a malha é refinada. As discretizações utilizadas em todos os algoritmos e em todas as variáveis, são de segunda ordem, portanto, espera-se que $P_E \rightarrow P_L = 2$. A ordem efetiva é calculada com (Marchi, 2001)

$$P_E = \frac{\log \left(\frac{\Phi - \phi_1}{\Phi - \phi_2} \right)}{\log (q)}, \quad (6.1)$$

onde ϕ_1 e ϕ_2 são soluções numéricas de duas malhas e $q = h_1/h_2$, onde h_1 e h_2 representam os tamanhos dos volumes de controle das malhas grossa e fina, respectivamente. Neste trabalho foi adotado $q = 2$. Como variável de interesse, para ϕ_2 e ϕ_1 foi utilizado a solução numérica no ponto central do domínio para o tempo final da simulação (último passo de tempo) 1s (segundo).

No caso das taxas de convergência, as ordens dependem do método de projeção e da norma considerada. Os resultados obtidos neste trabalho são comparados com as estimativas teóricas apresentadas em Guermond et al. (2006). É importante observar que o método de projeção paralelizado no tempo e espaço proposto neste trabalho deve apresentar as mesmas ordens do método de projeção de referência, visto que é uma adaptação deste.

6.2.1 Ordem efetiva

Nesta seção será verificado se a ordem efetiva do erro (P_E) tende para a ordem assintótica para o caso dos Métodos 1, 2, 3 e de referência. Para tal deve-se observar que a solução numérica foi considerada no último passo de tempo, visto que o Problema 1 depende do tempo. Portanto, para atingir o tempo em que se deseja a solução numérica foram necessários centenas de passos de tempo: em todos os testes o tempo final (no último passo de tempo) é 1 segundo e a simulação parte de 0 segundos (t inicial).

A Fig. 6.1 apresenta a ordem efetiva do (P_E) do erro para as velocidades e pressão. Nela pode-se observar que a ordem (P_E) do erro, calculado com a Eq. (6.1), para o ponto central do domínio no último passo de tempo, de todas as variáveis primitivas tende para a ordem assintótica esperada ($P_L = 2$) para todos os métodos sequenciais testados. Isto significa que para todos os métodos, o erro de discretização espacial *a posteriori* está de acordo com o previsto *a priori*, já que todas as discretizações espaciais utilizadas são segunda ordem.

6.2.2 Taxa de convergência

Nesta seção serão verificadas se as taxas de convergência (discretização temporal) estão de acordo com as estimativas teóricas apresentadas em Guermond et al. (2006).

A Fig. 6.2 apresenta as taxas de convergência para as normas l_1 , l_2 e l_∞ dos métodos 1, 2, 3 e de referência. Taxas de convergência referem-se à inclinação das curvas apresentadas na Fig. 6.2. Pode-se observar que para todos os métodos testados as velocidades possuem taxas de convergência muito próximas e que apenas os Métodos 3 e de referência possuem a mesma taxa de convergência para a pressão para as normas avaliadas.

Os resultados apresentados na Fig. 6.2 são confirmados pela Tab. 6.1. Pode-se concluir desta tabela que as velocidades são de segunda ordem para todas as normas e métodos estudados. No entanto as taxas de convergência da pressão variam de acordo com método e com a norma considerada: observa-se que o Método 1 é de primeira ordem na norma l_∞ , o Método 2 é de

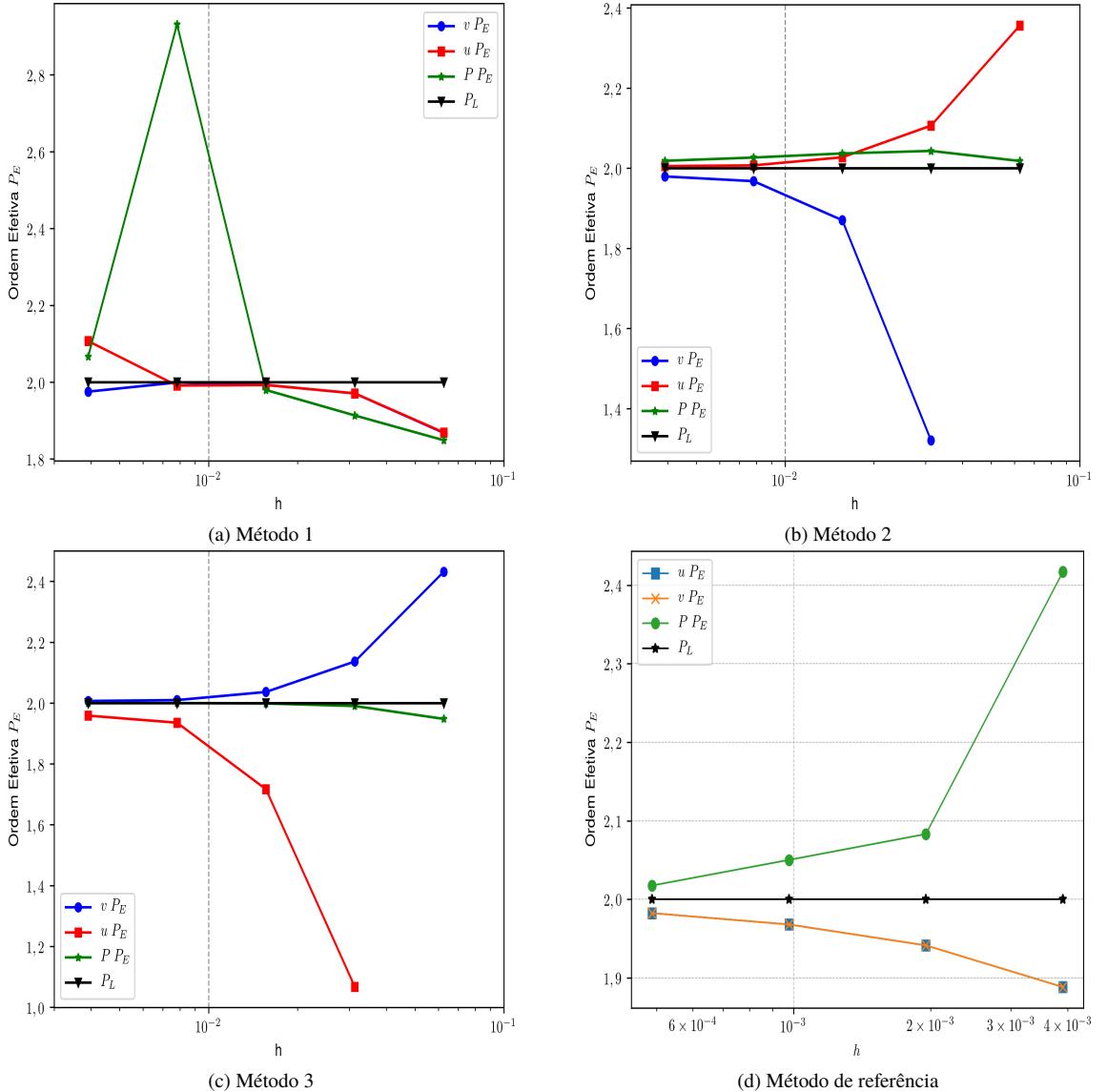


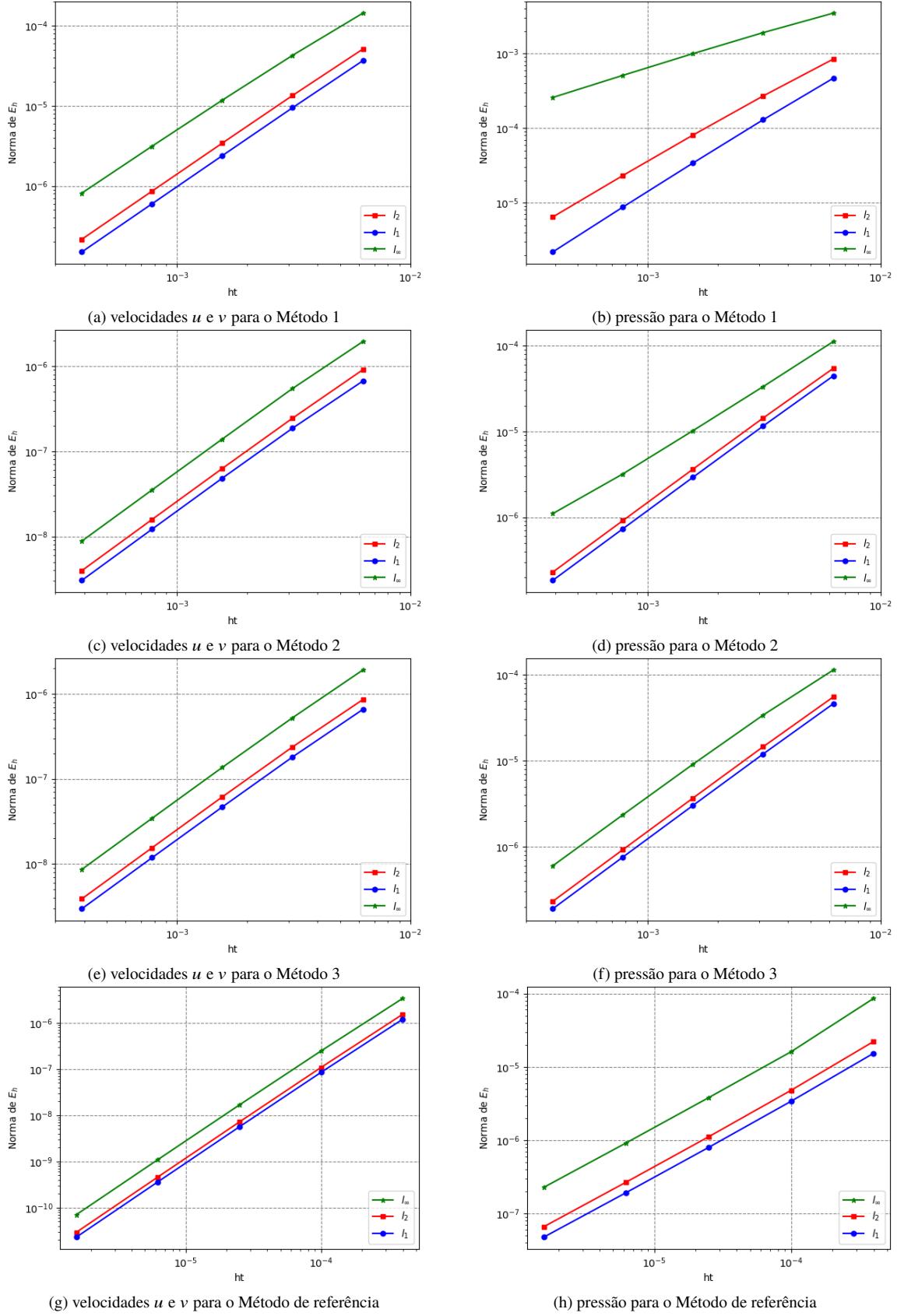
Figura 6.1: Ordem efetiva do erro (P_E) para as variáveis u , v e P

ordem 1,6 na norma l_∞ , o Método 3 é de segunda ordem em todas as normas e o método de referência é de primeira ordem para todas as normas testadas.

Todos os resultados obtidos para os métodos de projeção sequenciais no tempo estão de acordo com as estimativas teóricas apresentadas em Guermond et al. (2006). Segundo estes autores, por exemplo, a taxa de convergência da pressão para o Método 2 na norma l_∞ é de 1,5, consideravelmente próxima do valor 1,6 obtido neste trabalho.

Tabela 6.1: Taxas de convergência para os métodos sequenciais

	Método 1			Método 2			Método 3			Método referência		
P	l_1	l_2	l_∞	l_1	l_2	l_∞	l_1	l_2	l_∞	l_1	l_2	l_∞
u	1,963	1,797	0,963	1,994	1,991	1,644	1,995	1,993	1,944	1,041	1,048	1,063
v	1,995	1,989	1,906	1,975	1,978	1,979	1,978	1,98	1,977	1,962	1,963	1,947

Figura 6.2: Taxas de convergência para as variáveis u , v e P

6.2.3 Multigrid versus singlegrid

Nesta seção serão apresentados os resultados do *multigrid* e do *singlegrid*. Basicamente foi utilizado o método iterativo Gauss-Seidel *red-black* para o *singlegrid* e como suavizador para o método *multigrid* (veja capítulo 2).

Para gerar o tempo de CPU foi resolvido o Problema 1 usando-se o Algoritmo 3 (Método de projeção sequencial) considerando apenas o primeiro passo de tempo (lembrando que h_t depende do refinamento espacial utilizado de acordo com as restrições ao passo de tempo). A solução numérica depende do tempo e diferentes refinamentos de tempo podem gerar diferentes números de passos de tempo, logo, para que a comparação tenha sentido foi realizado apenas um passo de tempo para vários tamanhos de malha. Foram realizadas pelo menos 5 simulações (repetições) de cada experimento numérico com as respectivas coletas de tempo de CPU para cada tamanho de malha. O tempo apresentado na Fig. 6.3 é uma média destes tempos (esta figura está na escala bi-logarítmica). Além disso, foi utilizado o mesmo critério de parada para ambos os métodos: norma l_2 do resíduo da iteração atual dividida pela norma l_2 do resíduo na estimativa inicial. Esta razão deve ser menor ou igual à uma tolerância, isto é $\|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(0)}\|_2 \leq \varepsilon$. Neste trabalho foi utilizado $\varepsilon = 10^{-9}$.

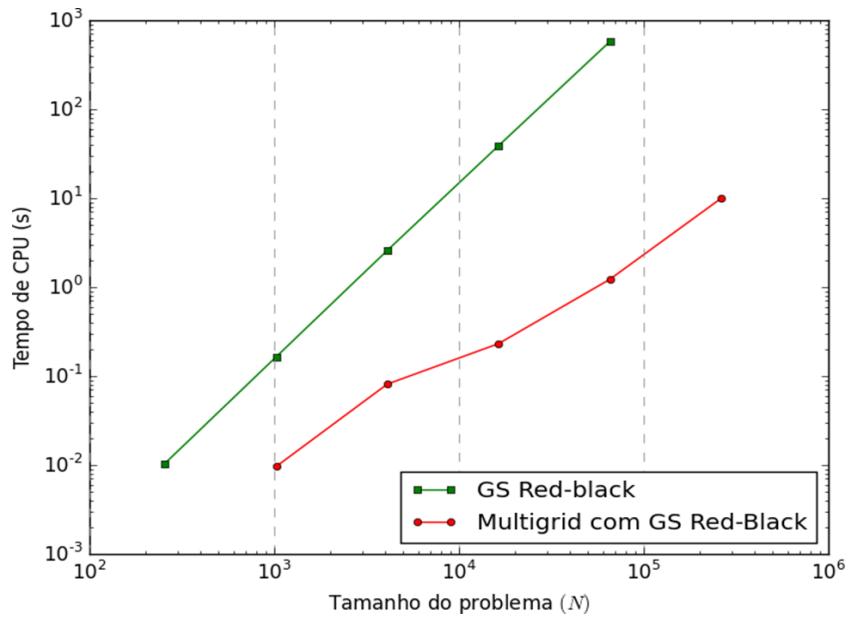


Figura 6.3: Tempo de CPU para *multigrid* versus *singlegrid*.

Na Tab. 6.2 são apresentados os tempos de CPU do *multigrid* (serial), do *singlegrid* e o fator *speedup* (aceleração do *multigrid* em relação ao *singlegrid*). Pode-se observar um fator *speedup* significativo do *multigrid* em relação ao *singlegrid*. Pode-se observar ainda que o fator *speedup* aumenta com o refinamento da malha, uma propriedade desejável.

Tabela 6.2: Tempo de CPU *multigrid* (MG) versus *singlegrid* (SG)

Malha	Multigrid	Singlegrid	speedup
32×32	0,01	0,16	16
64×64	0,08	2,58	32
128×128	0,23	38,8	169
256×256	1,22	572,94	470
512×512	9,91	—	—

A partir de malhas com 512×512 volumes torna-se inviável a coleta dos tempos de processamento do *singlegrid*. Para a malha com 512×512 volumes, por exemplo, o tempo de processamento de apenas uma simulação seria de aproximadamente 3,6 horas. Desta forma optou-se por não realizar simulações com *singlegrid* a partir de malhas com 256×256 volumes.

Para calcular o esforço computacional (ordem de complexidade) foi realizado o ajuste geométrico dos dados da Tab. 6.2 considerando a função

$$TCPU(N) = cN^p, \quad (6.2)$$

onde p representa a ordem do *solver* utilizado (*multigrid* ou *singlegrid*) ou a inclinação das curvas apresentadas na Fig. 6.3. No caso do *multigrid*, quanto mais próximo p estiver de 1 melhor o desempenho do algoritmo testado (Briggs et al., 2000; Trottenberg et al., 2001). A Tab. 6.3 apresenta o valor de p calculado utilizando a Eq. (6.2) e os dados da Tab. 6.2.

Tabela 6.3: Esforço computacional dos algoritmos *multigrid* e *singlegrid*

	<i>Multigrid</i>	<i>Singlegrid</i>
p	1,196	1,978

6.2.4 Resultados para o Problema 2

Nesta seção serão apresentados resultados para as simulações do Problema 2.

Apesar do Problema 2 (cavidade com tampa deslizante) não ter solução analítica para comparações diretas, em Ghia et al. (1982) podem ser encontrados resultados numéricos para comparação, resultados estes amplamente utilizados na literatura. Como não é conhecido o momento a partir do qual o Problema 2 atinge o regime permanente, não é possível definir um tempo final (T_f) para este problema. Logo, para interromper o processo iterativo foi considerado $\max|\varphi^n - \varphi^{n-1}| \leq 10^{-6}$, onde φ representa a variável de interesse (pressão ou velocidades) e n é o passo de tempo atual. O processo só foi interrompido quando todas as variáveis atingiram a tolerância estabelecida.

Na Fig. 6.4 são apresentadas as simulações para o Problema 2. Nela são mostrados os resultados de Ghia et al. (1982) para malhas com 128×128 volumes e resultados deste trabalho utilizando malha de 1024×1024 volumes e Reynolds variando de 100 até 5000. Pode-se observar boa concordância entre os resultados até Reynolds igual a 1000 e pequenas discrepâncias para Reynolds igual 5000. Pode-se observar também que há uma forte dependência dos perfis das velocidades em relação ao número de Reynolds.

Na Fig. 6.5 são apresentadas as simulações para o Problema 2 utilizando Reynolds igual a 5000. Nela são mostrados os resultados dos trabalhos de Ghia et al. (1982) para malhas com 128×128 volumes, Kumar et al. (2009) utilizando malhas com 513×513 volumes e os resultados obtidos neste trabalho empregando malhas com 1024×1024 volumes.

Da Fig. 6.5 pode-se observar que os resultados obtidos neste trabalho, em geral, apresentam melhor concordância com Kumar et al. (2009) do que com os resultados obtidos por Ghia et al. (1982).

Nas Tabs. 6.5 e 6.4 são apresentadas as diferenças percentuais de u em $x = 0,5$ e v em $y = 0,5$ para $Re = 5000$. As diferenças percentuais são calculadas de acordo com

$$DP = \frac{\text{Presente} - \text{Referência}}{\text{Referência}} \quad (6.3)$$

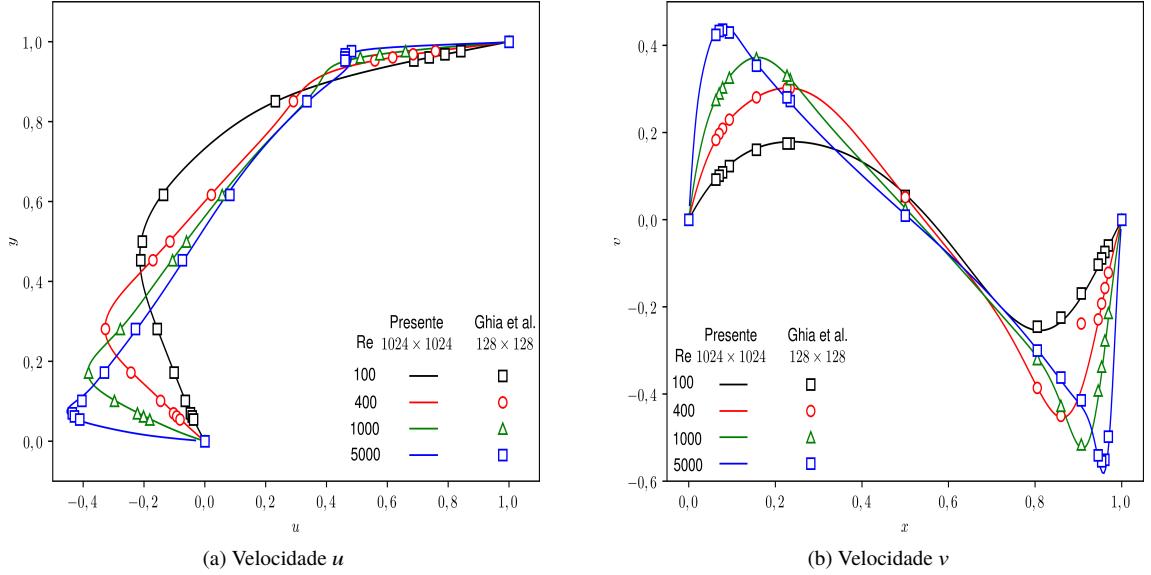


Figura 6.4: Perfis das velocidades u e v em $x = 0,5$ e $y = 0,5$ para malhas de 1024×1024 .

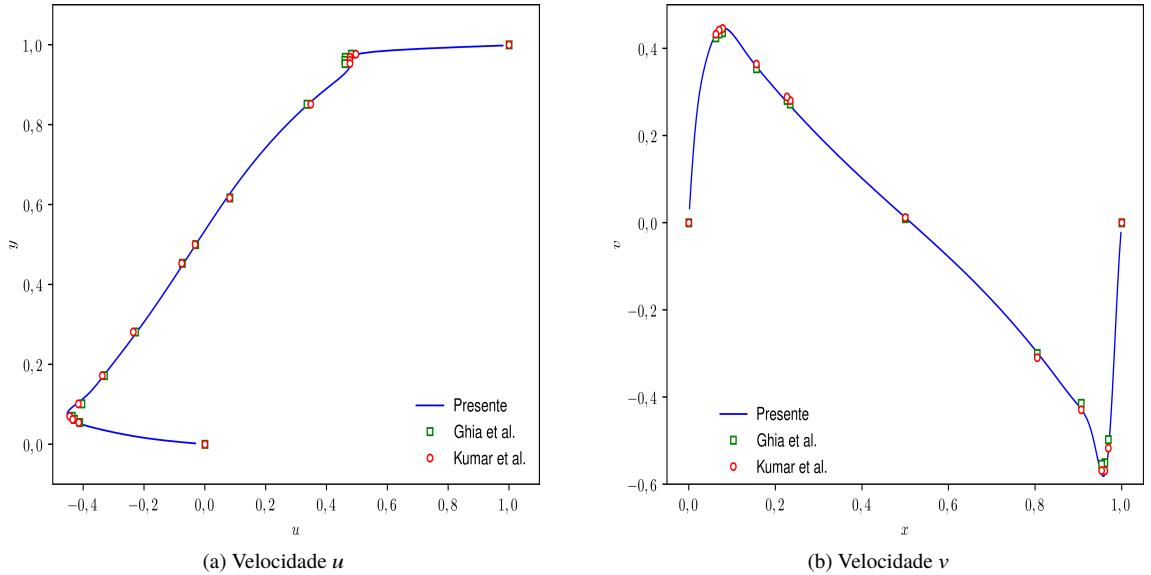


Figura 6.5: Perfis das velocidades u e v em $x = 0,5$ e $y = 0,5$ para $Re = 5000$.

onde DP é o valor da diferença percentual, Presente os resultados obtidos neste trabalho e Referência os valores apresentados nos trabalhos de Ghia et al. (1982) e Kumar et al. (2009).

Da Tab. 6.4 pode-se observar que os resultados obtidos apresentam melhor concordância com Kumar et al. (2009), calculando-se a média do valor absoluto das diferenças percentuais de v obtém-se para Ghia et al. (1982) média de 4,89% e para Kumar et al. (2009) 2,75%.

De forma similar, da Tab. 6.5 para Ghia et al. (1982) a diferença percentual média de u obtida foi de 3,41% enquanto que para Kumar et al. (2009) este valor foi 2,46%.

Das Tabs. 6.4 e 6.5 nota-se que, exceto para v em $x = 0,5$, as diferenças percentuais não superam 9 %.

Tabela 6.4: Diferenças percentuais de v em $y = 0,5$ para $Re = 5000$.

	Presente	Ghia et al. (1982)		Kumar et al. (2009)	
x	v	v	DP	v	DP
0,00000	0,03302	0,00000	0,00000	0,00000	0,00000
0,96880	-0,50665	-0,49774	1,79090	-0,51740	-2,07692
0,96090	-0,58166	-0,55069	5,62300	-0,56888	2,24569
0,95310	-0,57063	-0,55408	2,98666	-0,56835	0,40090
0,90630	-0,43049	-0,41442	3,87776	-0,42919	0,30295
0,80470	-0,30028	-0,30018	0,03470	-0,30968	-3,03405
0,50000	0,01360	0,00945	43,93710	0,01163	16,95663
0,23440	0,26617	0,27280	-2,42948	0,28036	-5,06050
0,22660	0,27478	0,28066	-2,09509	0,28859	-4,78537
0,15630	0,35605	0,35368	0,66980	0,36438	-2,28636
0,07810	0,44585	0,43648	2,14778	0,44599	-0,03035
0,07030	0,44032	0,43329	1,62231	0,44231	-0,45007
0,06250	0,42882	0,42447	1,02473	0,43287	-0,93569
1,00000	-0,02294	0,00000	0,00000	0,00000	0,00000

Tabela 6.5: Diferenças percentuais de u em $x = 0,5$ para $Re = 5000$.

	Presente	Ghia et al. (1982)		Kumar et al. (2009)	
y	u	u	DP	u	DP
1,00000	1,00000	1,00000	0,00000	1,00000	0,00000
0,97660	0,51557	0,48223	6,91467	0,49601	3,94439
0,96880	0,48378	0,46120	4,89599	0,47696	1,42995
0,96090	0,48049	0,45992	4,47149	0,47649	0,83849
0,95310	0,48001	0,46036	4,26786	0,47699	0,63262
0,85160	0,34345	0,33556	2,35110	0,34748	-1,15997
0,61720	0,07469	0,08183	-8,72162	0,08098	-7,76352
0,50000	-0,03117	-0,03039	2,55171	-0,03208	-2,85079
0,45310	-0,07096	-0,07404	-4,15633	-0,07604	-6,67720
0,28130	-0,22131	-0,22855	-3,16837	-0,23558	-6,05795
0,17190	-0,33060	-0,33050	0,02912	-0,33743	-2,02524
0,10160	-0,41781	-0,40435	3,32861	-0,41598	0,43974
0,07030	-0,45062	-0,43643	3,25176	-0,44444	1,39089
0,06250	-0,43864	-0,42901	2,24528	-0,43485	0,87214
0,05470	-0,42869	-0,41165	4,13968	-0,41478	3,35382
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

6.3 Avaliação do método de projeção paralelizado

Nesta seção serão avaliados a convergência, relação entre iterações externas e internas e desempenho paralelo do método de projeção paralelizado proposto, Algoritmo 5, para o Problema 1 (Vórtices de Taylor-Green).

6.3.1 Número de iterações externas *versus* o número de iterações do *solver*

Considerando-se que no Algoritmo 5 foi introduzido um ciclo externo, nesta seção será discutida a relação destas iterações com as iterações do *solver*.

A Fig. 6.6(a) mostra o número de iterações externas *versus* o número de iterações do *solver* para o caso do *singlegrid*. Em 6.6(b) é mostrado o tempo de processamento (TCPU para apenas uma *thread*) para o número de iterações externas apresentadas em 6.6(a). O tempo de processamento mostrado em 6.6(b) foi normalizado dividindo-se pelo tempo de processamento utilizando apenas uma iteração do *solver* para que os resultados das malhas possam ser apresentados no mesmo gráfico.

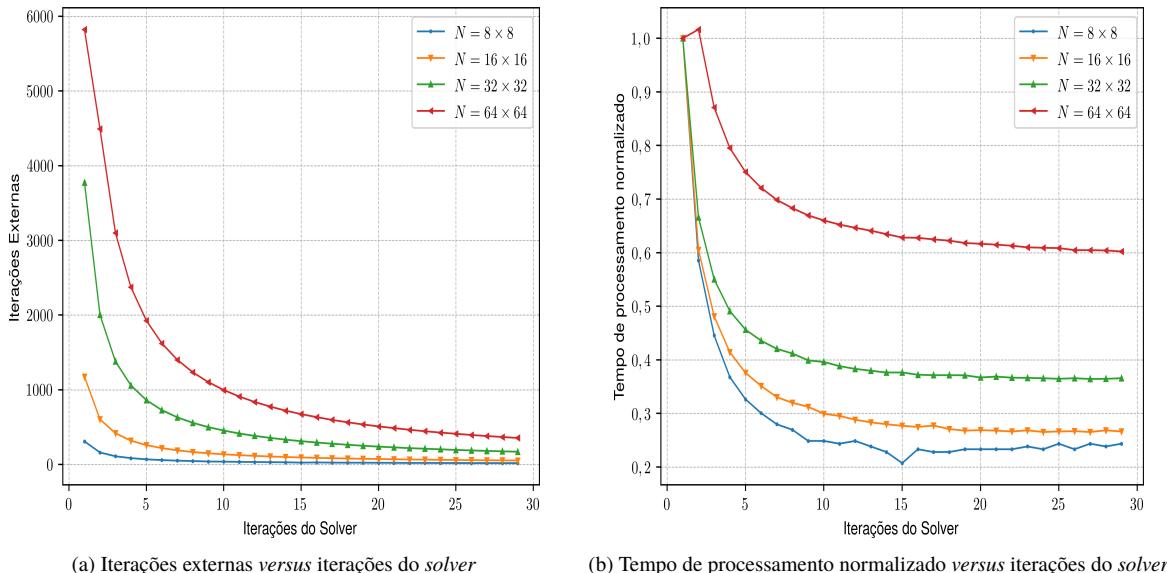


Figura 6.6: Comparativo entre iterações externas contra iterações do *solver*.

Da Fig. 6.6(a) pode-se concluir que o aumento de iterações do *solver* reduz o número de iterações externas. Naturalmente este resultado é esperado considerando-se que a filosofia do Algoritmo 5 é a cada iteração externa obter uma aproximação para as variáveis de interesse, logo, mais iterações do *solver* resulta em melhores aproximações e portanto menos iterações externas. O conhecimento da relação entre iterações externas e do *solver* é importante no contexto de otimizações. Por exemplo: no caso da malha com 64×64 volumes é vantajoso realizar até 25 iterações do *solver* (40% de redução no tempo de processamento) e então focar na otimização das iterações externas, principalmente pelo fato de que é em relação aos passos de tempo que o Algoritmo 5 está paralelizado.

A Fig. 6.6(b) apresenta um resultado muito mais interessante: com o refinamento das malhas o tempo de processamento empregado no *solver* torna-se mais custoso. Apesar de não ter sido mostrado neste trabalho, pode-se esperar que para malhas mais refinadas, o custo computacional de uma iteração do *solver* seja equivalente ao custo de uma iteração externa. Neste caso duas possibilidades: 1) substituir o *solver* por uma alternativa menos custosa ou, 2) manter

o *solver* e focar na otimização do ciclo externo. No contexto do Algoritmo 5 recomenda-se a otimização do ciclo externo, pois este pode ser realizado em paralelo.

6.3.2 Ordem efetiva e taxa de convergência

Nesta seção será verificado se a ordem efetiva do erro (P_E) tende para a ordem assintótica para o caso do método de projeção paralelizado. Para tal deve-se observar que a solução numérica foi considerada no último passo de tempo e para o ponto central do domínio.

Na Fig. 6.7 pode-se observar que a ordem efetiva (P_E) do erro das variáveis tende para a ordem assintótica esperada ($P_L = 2$). Além disso, é interessante notar que a Fig. 6.7 é idêntica a Fig. 6.1(d).

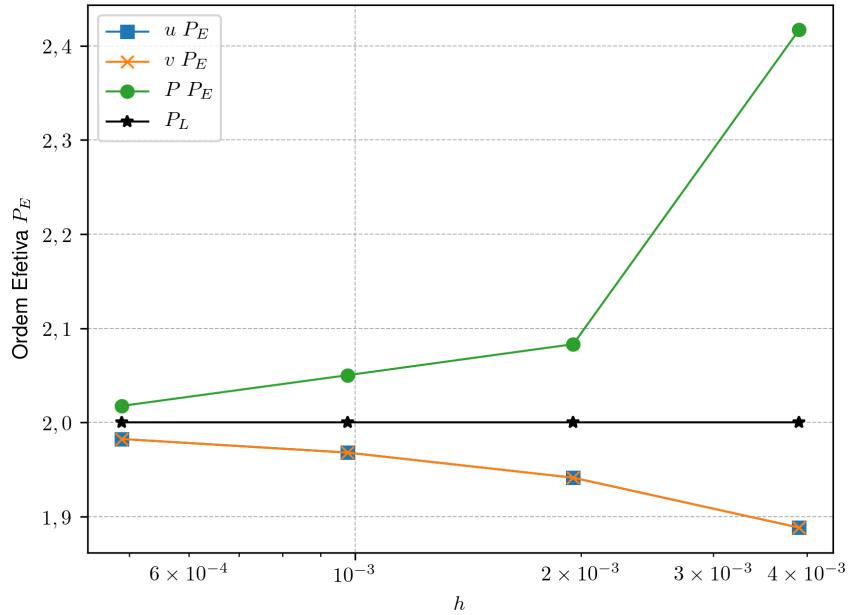


Figura 6.7: Ordem efetiva do erro (P_E) para todas as variáveis primitivas

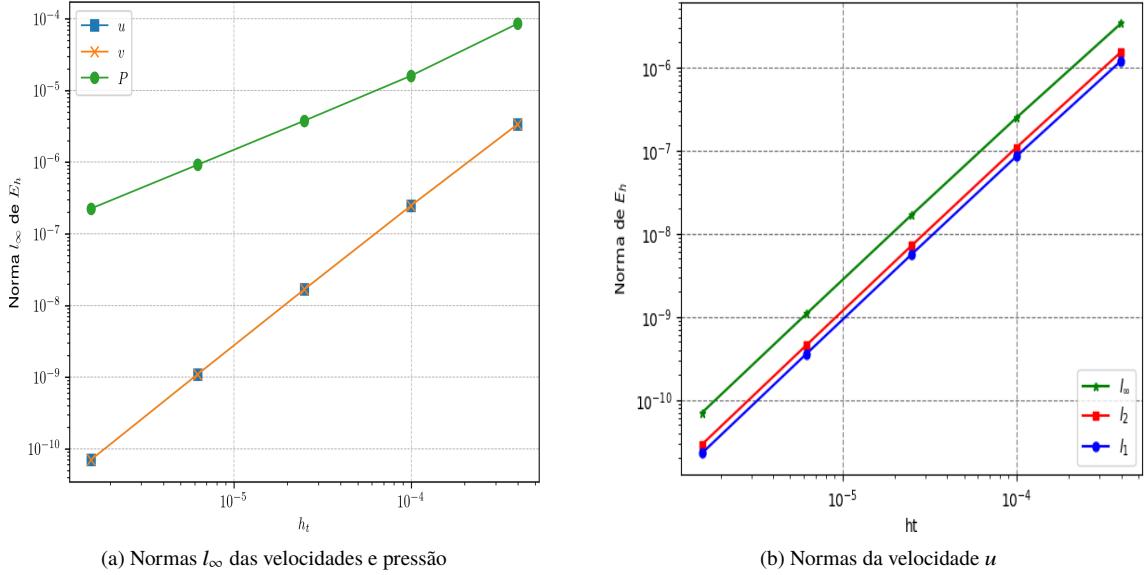
Na Fig. 6.8(a) são apresentadas as normas l_∞ para as velocidades e pressão. Pode-se observar que as velocidades são de segunda ordem e a pressão é, como já esperado (Guermond et al., 2006), de primeira ordem.

Na Fig. 6.8(b) são apresentadas as taxas de convergência em relação às normas l_1 , l_2 e l_∞ para a velocidade u . Para este caso em particular os resultados para v ficaram idênticos aos resultados de u , por isso foi utilizada a mesma figura. Della pode-se observar que as velocidades são de segunda ordem para todas as normas testadas.

Com base nos resultados obtidos pode-se constatar que as ordens efetivas e as taxas de convergência do algoritmo proposto (método de projeção em paralelo) não foram alteradas, isto é, do ponto de vista dos erros de discretização espacial e temporal, o algoritmo proposto é idêntico ao sequencial. Este é um resultado importante porque demonstra que as melhorias de desempenho do novo algoritmo, que iremos mostrar na próxima seção, estão associadas exclusivamente ao seu paralelismo, que é a proposta deste trabalho.

6.3.3 Desempenho paralelo

Nesta seção são apresentados os resultados da paralelização do método de projeção utilizando o método *multigrid*. Vale lembrar que a proposta deste trabalho foi a de modificar os

Figura 6.8: Taxas de convergência das variáveis primitivas u , v e P .

métodos de projeção sequenciais para que estes permitam processamento paralelo no tempo e no espaço, logo, nesta seção são apresentados os resultados mais importantes deste trabalho. Dos resultados apresentados aqui espera-se principalmente:

1. O tempo de CPU do método proposto, ou seja, do método de projeção paralelizado no tempo e no espaço, seja menor em relação ao método de projeção de referência sequencial no tempo.
2. O método proposto apresente escalabilidade, isto é, que o tempo de CPU diminua em relação à versão serial à medida que novos processadores sejam acrescentados no sistema.

Para a geração dos tempos computacionais foram simulados 16 passos de tempo de forma que os tempos totais de CPU medidos ficassem acima de 1s e, portanto, acima da precisão da função OMP_GET_WTIME() do módulo OMP da linguagem FORTRAN. Além disso, foram feitas pelo menos 5 simulações de cada experimento numérico com as respectivas coletas de tempo de CPU para cada tamanho de problema e *thread* utilizada. O tempo de CPU considerado foi a média destes tempos.

A Fig. 6.9(a) mostra o fator *speedup* (aceleração da versão paralela em relação à versão paralela utilizando apenas 1 *thread*) versus o número de *threads*, com 16 passos de tempo utilizando-se várias malhas e o método Gauss-Seidel *red-black* (*singlegrid*). Desta figura pode-se observar redução significativa no tempo de CPU (14 vezes utilizando 16 *threads*). Além disso pode-se observar que à medida que *threads* são adicionadas no sistema, o tempo de CPU diminui, portanto o Algoritmo 5 apresenta escalabilidade. Pode-se notar que a aceleração da versão paralela do algoritmo de projeção aumenta com o refinamento da malha, portanto conclui-se que o maior *speedup* acontecerá na malha mais refinada possível o que é uma propriedade desejável.

A Fig. 6.9(b) mostra o fator *speedup* versus o número de *threads*, com 16 passos de tempo utilizando-se várias malhas e o método *multigrid*.

Uma análise da Fig. 6.9(b) mostra que o algoritmo 5 utilizando o método *multigrid* apresentou características vantajosas em relação ao paralelismo como a escalabilidade em relação ao refinamento das malhas. Desta forma pode-se concluir que o maior *speedup* será obtido

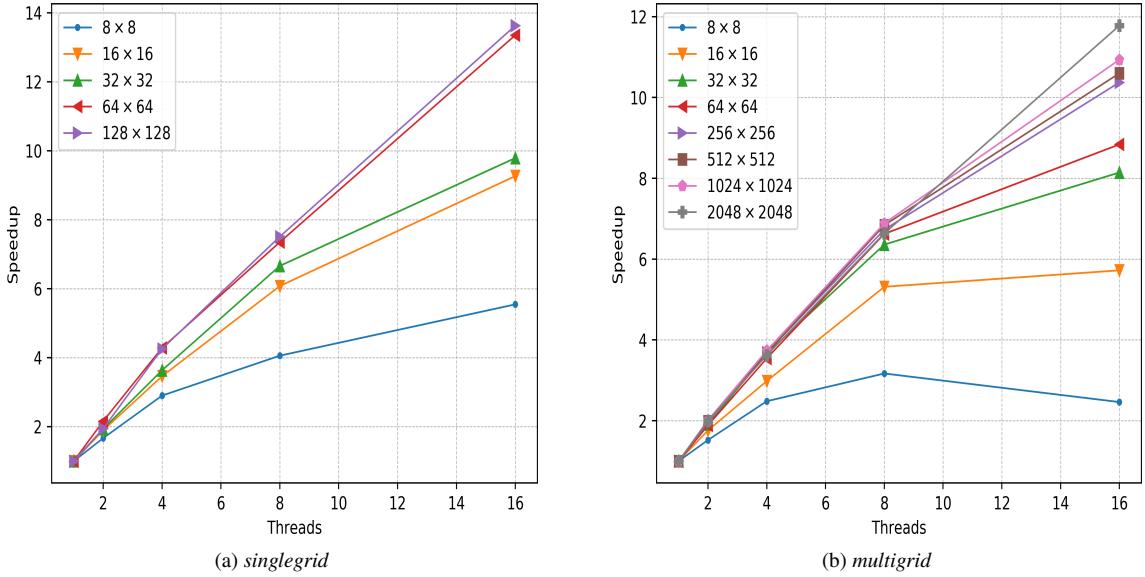


Figura 6.9: Speedups para os métodos *singlegrid* e *multigrid* para várias malhas e 16 threads

utilizando o número máximo de *threads* disponíveis na malha mais refinada possível, o que é um resultado desejável.

Apesar do método *singlegrid* apresentar escalabilidade superior ao método *multigrid*, e portanto se beneficiar mais do paralelismo, este método é inviável para malhas muito refinadas. Neste trabalho a malha mais refinada viável (tempo de processamento aceitável) para a utilização do método *singlegrid* foi uma malha de 128×128 volumes. Com o método *multigrid* foi possível testar malhas mais refinadas.

A Fig. 6.10 apresenta o fator *speedup* para as malhas mais refinadas testadas para os métodos *singlegrid* (128×128) e *multigrid* (2048×2048). Além disso, são apresentados as curvas de *speedup* teórico máximo previsto, pela Eq. (2.17), para programas que têm, hipoteticamente, frações paralelas de 95%, 97% e 99%. Como pode-se observar para ambos os métodos o Algoritmo 5 apresentou boa escalabilidade, realizando, pelo menos, 97% do seu processamento em paralelo.

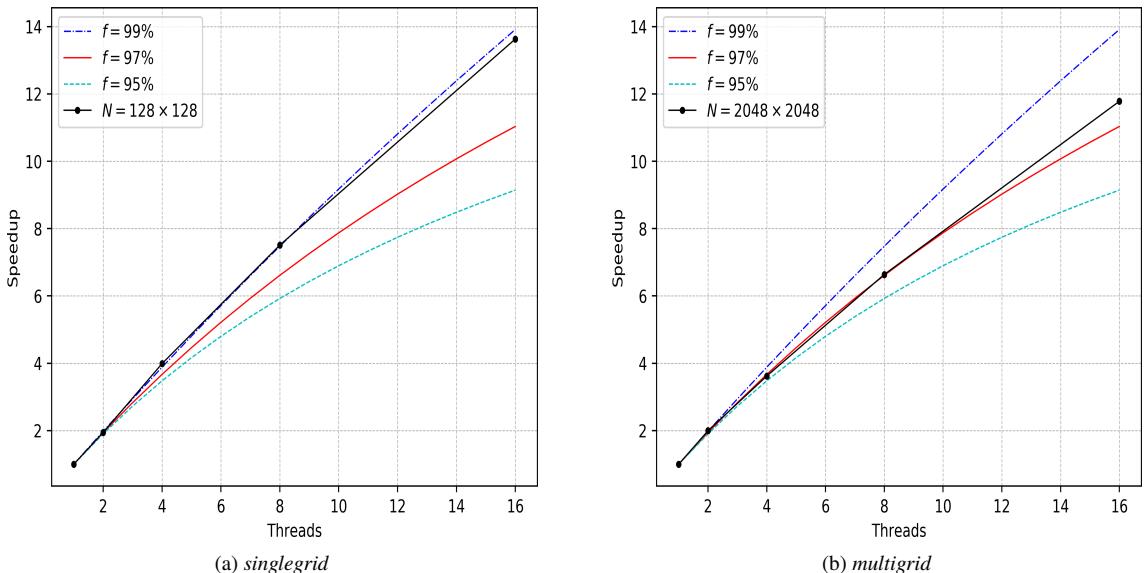


Figura 6.10: Escalabilidade para os métodos *singlegrid* e *multigrid*

Considerando que, tanto para o método *singlegrid* quanto para o método *multigrid*, o tempo de CPU foi consideravelmente reduzido e estes métodos apresentaram escalabilidade pode-se considerar que o Algoritmo 5 representa um avanço sobre o método de referência sequencial no tempo.

Na Fig. (6.11) pode-se observar o fator *speedup* obtido utilizando paralelização espacial, via decomposição de domínios, apresentado em Neundorf et al. (2016) empregando uma malha com 1024×1024 volumes e os resultados da paralelização temporal proposta neste trabalho utilizando uma malha similar. É possível notar que as abordagens (espacial e temporal) apresentam escalabilidades equivalentes até 8 *threads*, a partir deste valor a paralelização temporal proposta apresenta melhor desempenho em termos de uso de CPUs e portanto torna-se preferível em relação à paralelização espacial.

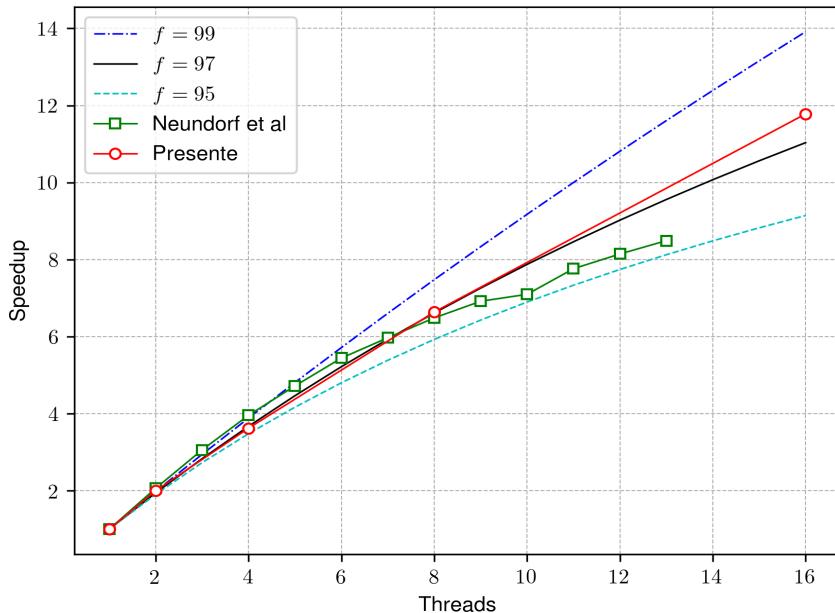


Figura 6.11: Paralelização espacial apresentada em Neundorf et al. (2016) *versus* paralelização temporal

6.3.4 Consolidação dos resultados

Nesta seção é apresentado um resumo dos resultados obtidos neste trabalho.

Na seção 6.1 são discutidos os detalhes de implementação e os critérios de parada utilizados para encerrar os processos iterativos. Além disso são expostos os parâmetros dos algoritmos, o *software* e o *hardware* empregados.

Nas seções 6.2.1 e 6.2.2 a ordem efetiva e taxas de convergência foram avaliadas e seus resultados comparados com previsões teóricas disponíveis na literatura (Guermond et al., 2006).

Na seção 6.2.3 foi demonstrado como os métodos de projeção beneficiam-se significativamente com emprego do método *multigrid*. Nesta seção são apresentados *speedups* consideráveis do método *multigrid* em relação ao uso de uma única malha (*singlegrid*).

Na seção 6.2.4 é discutido o problema da cavidade com tampa deslizante amplamente utilizado na literatura. É debatido o critério de parada e são apresentados resultados para fluidos com Reynolds variando de 100 até 5000, próximo do limite para o qual o escoamento pode ser considerado laminar (Erturk, 2009). Os resultados são comparados com os trabalhos de Ghia et al. (1982) e Kumar et al. (2009).

Na seção 6.3.1 são discutidas características do método de projeção proposto com foco na relação entre as iterações externas (ciclo introduzido no novo método) e as iterações do *solver*.

Nesta seção recomenda-se a otimização das iterações externas para o cenário em que iterações do *solver* apresentem custo comparável às iterações externas do método paralelizado.

Na seção 6.3.2 verifica-se que a ordem efetiva e a taxa de convergência do método paralelizado proposto são muito próximas da sua versão de referência.

Na seção 6.3.3 são apresentados o produto do estudo, isto é, o desempenho do algoritmo paralelizado no tempo e no espaço. São apresentados resultados que mostram escalabilidade satisfatória utilizando o método *multigrid* e *singlegrid*. Além disso é realizado um comparativo entre as escalabilidades espacial, obtida via decomposição de domínios, apresentada em Neundorf et al. (2016) e temporal obtidas neste trabalho.

7 Conclusão

Neste trabalho foi apresentado um método de projeção com correção incremental na pressão paralelizado no tempo e no espaço para a resolução de problema de escoamento laminar bidimensional em regime transiente de um fluido incompressível modelado pelas equações de Navier-Stokes nas variáveis primárias. Foi utilizado o método dos volumes finitos em malhas desencontradas para a discretização das variáveis. Os sistemas lineares gerados foram resolvidos com o método *multigrid* geométrico com esquema CS e ciclo V. O método *multigrid* foi usado em conjunto com técnicas de paralelização no tempo e no espaço.

7.1 Constatações gerais

Com base nos resultados obtidos neste trabalho verificou-se que:

1. Foi possível modificar os métodos de projeção sequenciais para que permitam processamento paralelo para cada passo de tempo.
2. As ordens de discretização espacial e temporal do algoritmo proposto são similares às do algoritmo sequencial.
3. Foram realizados testes numéricos com resultados satisfatórios para um problema com solução analítica conhecida (vórtices de Taylor-Green) e solução analítica não conhecida (problema da cavidade com tampa móvel de Ghia et al. (1982)).
4. Utilizando-se 16 *threads* foi possível reduzir em até 12 vezes, de sua versão serial, o tempo de CPU necessário para se obter as soluções das equações de Navier-Stokes.
5. Verificou-se que, empregando o método de projeção paralelizado, o tempo de CPU diminuiu à medida que aumentou-se o número processadores utilizados, ou seja, o algoritmo proposto apresenta escalabilidade.
6. O uso de memória é a principal desvantagem do método proposto, pois é proporcional ao número de passos de tempo processados em paralelo.

7.2 Contribuições

Este trabalho contribui com a literatura no sentido que:

1. Desenvolveu uma nova formulação dos métodos de projeção paralelizados no tempo e no espaço.
2. Apresenta a paralelização do método *multigrid* em relação ao tempo no contexto do novo método de projeção.

3. Apresenta a paralelização espacial da atualização da pressão via decomposição de domínios.
4. Demonstra que o novo algoritmo apresenta escalabilidade, isto é, o desempenho melhora à medida que mais processadores são inseridos no sistema.

7.3 Trabalhos futuros

Apresentam-se a seguir algumas questões que servem de sugestões para novas pesquisas:

1. O método de projeção modificado é conhecido como correção na pressão. Nestes métodos são resolvidos dois sistemas lineares na primeira etapa e um na segunda. O método de projeção proposto neste trabalho consiste justamente em permitir que a segunda etapa seja processada em paralelo. Considerando que o custo computacional da primeira etapa é maior, propõe-se modificar os métodos de projeção conhecidos como correção nas velocidades, pois estes possuem etapas invertidas em relação aos métodos com correção na pressão.
2. Neste trabalho foi escolhido modificar um método de projeção mais simples e de acurácia reduzida (segunda ordem nas velocidades e primeira na pressão). Propõe-se modificar métodos mais complexos e acurados.
3. Na resolução dos sistemas lineares foi empregado apenas o suavizador Gauss-Seidel *Red-Black*. Propõe-se utilizar outros suavizadores como ILU, Gradiente Conjugado, etc.

Referências

- Al-Nasra, M. e Nguyen, D. (1991). An algorithm for domain decomposition in finite element analysis. *Computers & Structures*, 39(3):277 – 289.
- Almgren, A., Aspden, A., Bell, J. e Minion, M. (2013). On the use of higher-order projection methods for incompressible turbulent flow. *SIAM Journal on Scientific Computing*, 35(1):B25–B42.
- Almgren, A. S., Bell, J. B., Colella, P. e Howell, L. (1993). An adaptive projection method for the incompressible Euler equations. Em *Proceedings of the 11th Computational Fluid Dynamics Conference*, página 530.
- Almgren, A. S., Bell, J. B. e Szymczak, W. G. (1996). A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM Journal on Scientific Computing*, 17.
- Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. Em *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), páginas 483–485, New York, NY, USA. ACM.
- Astorino, M. e Grandmont, C. (2010). Convergence analysis of a projection semi-implicit coupling scheme for fluid–structure interaction problems. *Numerische Mathematik*, 116(4):721–767.
- Baker, A. H., Schulz, M. e Yang, U. M. (2011). On the performance of an algebraic multigrid solver on multicore clusters. Em Palma, J. M. L. M., Daydé, M., Marques, O. e Lopes, J. C., editores, *High Performance Computing for Computational Science – VECPAR 2010*, páginas 102–115, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Bell, J., Colella, P. e Glaz, H. M. (1989). A second order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283.
- Borregales, M., Kumar, K., Radu, F. A., Rodrigo, C. e Gaspar, F. J. (2018). A parallel-in-time fixed-stress splitting method for Biot’s consolidation model. *ArXiv e-prints*.
- Bramble, J. H., Pasciak, J. E. e Xu, J. (1990). Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22.
- Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390.
- Briggs, W. L., Henson, V. E. e McCormick, S. F. (2000). *A Multigrid Tutorial (2Nd Ed.)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Brown, D. L., Cortez, R. e Minion, M. L. (2001). Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 168(2):464 – 499.
- Burden, R. e Faires, J. (2008). *Análise numérica*. Cengage Learning.

- Chan e Saad (1986). Multigrid algorithms on the hypercube multiprocessor. *IEEE Transactions on Computers*, C-35(11):969–977.
- Chan, T. e Schreiber, R. (1985). Parallel networks for multigrid algorithms: Architecture and complexity. *Siam Journal on Scientific and Statistical Computing*, 6.
- Chapman, B., Jost, G. e Pas, R. v. d. (2007). *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. The MIT Press.
- Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22(104):745.
- Christou, I. T. e Meyer, R. R. (1996). Optimal equi-partition of rectangular domains for parallel computation. *Journal of Global Optimization*, 8(1):15–34.
- Courtecuisse, H. e Allard, J. (2009). Parallel dense gauss-seidel algorithm on many-core processors. Em *2009 11th IEEE International Conference on High Performance Computing and Communications*, páginas 139–147.
- Donaldson, W. (2000). *Grid-Graph Partitioning*. Tese de doutorado, University of Wisconsin, Madison, EUA.
- E, W. e Liu, J.-G. (1995). Projection method i: Convergence and numerical boundary layers. *SIAM Journal on Numerical Analysis*, 32(4):1017–1057.
- Erturk, E. (2009). Discussions on driven cavity flow. *International Journal for Numerical Methods in Fluids*, 60(3):275–294.
- Ferziger, J. e Peric, M. (2001). *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg.
- Fortuna, A. . (2000). *Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações*. Edusp.
- Frederickson, P. O. e McByran, O. A. (1987a). Parallel superconvergent multigrid. Relatório técnico, Ithaca, NY, USA.
- Frederickson, P. O. e McByran, O. A. (1987b). Superconvergent multigrid methods. *Cornell Theory Center Preprint*.
- Frederickson, P. O. e McByran, O. A. (1990). Proceedings of the third european conference on multigrid methods, october, ed u. trottenberg and w. hackbusch. *Cornell Theory Center Preprint*.
- Friedhoff, S., Falgout, R. D., Kolev, T. V., MacLachlan, S. e Schroder, J. B. (2013). A multigrid-in-time algorithm for solving evolution equations in parallel. Em *Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States, Mar 17 - Mar 22, 2013*.
- Gannon, D. e Rosendale, J. V. (1986). On the structure of parallelism in a highly concurrent pde solver. *Journal of Parallel and Distributed Computing*, 3(1):106 – 135.

- Gaspar, F. J. e Rodrigo, C. (2017). On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics. *Computer Methods in Applied Mechanics and Engineering*, 326:526 – 540.
- Ghia, U., Ghia, K. N. e Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411.
- Goda, K. (1979). A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of Computational Physics*, 30:76–95.
- Golub, G. e Van Loan, C. (2013). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press.
- Graham, R. L., Knuth, D. E. e Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.
- Gramma, A., Kumar, V., Gupta, A. e Karypis, G. (2003). *Introduction to Parallel Computing*. Pearson Education. Addison-Wesley.
- Greenbaum, A. (1986). A multigrid method for multiprocessors. *Applied Mathematics and Computation*, 19(1):75 – 88.
- Griffith, B. E. (2009). An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner. *Journal of Computational Physics*, 228(20):7565 – 7595.
- Guermond, J., Minev, P. e Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6011 – 6045.
- Guermond, J.-L. e Salgado, A. (2009). A splitting method for incompressible flows with variable density based on a pressure poisson equation. *Journal of Computational Physics*, 228(8):2834 – 2846.
- Guermond, J.-L. e Shen, J. (2001). Quelques résultats nouveaux sur les méthodes de projection. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 333(12):1111 – 1116.
- Guermond, J. L. e Shen, J. (2003). Velocity-correction projection methods for incompressible flows. *SIAM J. Numer. Anal.*, 41(1):112–134.
- Guermond, J. L. e Shen, J. (2004). On the error estimates for the rotational pressure-correction projection methods. *Mathematics of Computation*, 73(248):1719–1737.
- Gustafson, J. L. (1988). Reevaluating Amdahl's law. *Commun. ACM*, 31(5):532–533.
- Harlow, F. H. e Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189.
- Hempel, R. e Schüller, A. (1988). Experiments with parallel multigrid using the suprenum communications library. *GMD-Studie*, (141).
- Herbin, R., Gerbi, S. e Sonnad, V. (1988). Parallel implementation of a multigrid method on the experimental lcap supercomputer ii. *Applied Mathematics and Computation*, 27:281–312.

- Hirsch, C. (1991). *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*. Wiley Series in Numerical Methods in Engineering. Wiley.
- Hughes, T. (2000). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering. Dover Publications.
- Kadioglu, S. Y., Klein, R. e Minion, M. L. (2008). A fourth-order auxiliary variable projection method for zero-mach number gas dynamics. *Journal of Computational Physics*, 227(3):2012 – 2043.
- Kalland, K. (2008). A Navier-Stokes solver for single and two-phase flow. *Master's thesis, Faculty of Mathematics and Natural Sciences, Master Thesis , University of Oslo*.
- Karniadakis, G. E., Orszag, S. A. e Israeli, M. (1991). High-order splitting methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 97:414–443.
- Kim, J. e Moin, P. (1985). Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59(2):308 – 323.
- Kowarschik, M., Rüde, U., Weiß, C. e Karl, W. (2000). Cache-aware multigrid methods for solving poisson's equation in two dimensions. *Computing*, 64(4):381–399.
- Kumar, D. S., Kumar, K. S. e Das, M. K. (2009). A fine grid solution for a lid-driven cavity flow using multigrid method. *Engineering Applications of Computational Fluid Mechanics*, 3(3):336–354.
- Maliska, C. (2004). *Transferência de calor e mecânica dos fluidos computacional*. Livros Técnicos e Científicos.
- Marchi, C. H. (2001). *Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos*. Tese de doutorado, Universidade Federal de Santa Catarina, Florianópolis- SC.
- Martin, W. (1998). Fast equi-partitioning of rectangular domains using stripe decomposition. *Discrete Applied Mathematics*, 82(1):193 – 207.
- McAdams, A., Sifakis, E. e Teran, J. (2010). A parallel multigrid poisson solver for fluids simulation on large grids. Em *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, páginas 65–74, Goslar Germany, Germany. Eurographics Association.
- McBryan, O. A. (1990). Sequencial and parallel efficiency of multigrid fast solvers. *University of Colorado CS Dept. Tech Report, September*.
- Mesquita, M. S. (2000). Solução numérica de escoamentos bidimensionais não-isotérmicos usando o método multigrid. Dissertação de Mestrado, Engenharia Aeronáutica e Mecânica - Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos - SP.
- Neundorf, R. L. A., Pinto, M. A. V., Araki, L. K. e Calvetti, L. (2016). Desempenho de um algoritmo multigrid paralelo aplicado as equações de Navier-Stokes. Em *XXXVII Iberian Latin American Congress on Computational Methods in Engineering - CILAMCE, 2016, Brasília - DF. Proceedings of CILAMCE 2016, 2016. v. XXXVII*.

- Oliveira, F., Pinto, M., Marchi, C. e Araki, L. (2012). Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*, 36(10):4665 – 4676.
- Orszag, S. A., Israeli, M. e Deville, M. O. (1986). Boundary conditions for incompressible flows. *Journal of Scientific Computing*, 1(1):75–111.
- Oseledets, V. I. (1989). On a new way of writing the Navier-Stokes equation. the hamiltonian formalism. *Russian Mathematical Surveys*, 44(3):210.
- Pearson, C. E. (1965). A computational method for viscous flow problems. *Journal of Fluid Mechanics*, 21(4):611–622.
- Pinto, M. A. V. e Marchi, C. H. (2007). Optimum parameters of a geometric multigrid for the two-dimensional Laplace's equation. *Proceedings of COBEM*.
- Pletcher, R., Tannehill, J. e Anderson, D. (1997). *Computational Fluid Mechanics and Heat Transfer, Second Edition*. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis.
- Pritchard, P. e Mitchell, J. (2015). *Fox and McDonald's Introduction to Fluid Mechanics, 9th Edition*. Wiley.
- Quarteroni, A. e Valli, A. (1999). *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press.
- Reddy, J. e Gartling, D. (2010). *The Finite Element Method in Heat Transfer and Fluid Dynamics*. Applied and Computational Mechanics. CRC Press.
- Shen, J. (1996). On error estimates of the projection methods for the Navier-Stokes equations: Second-order schemes. *Mathematics of Computation*, 65(215):1039–1065.
- Smith, B., Bjørstad, P. e Gropp, W. (1996). *Domain decomposition : parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press.
- Sun, X. e Ni, L. (1993). Scalable problems and memory-bounded speedup. *Journal of Parallel and Distributed Computing*, 19(1):27 – 37.
- Sun, X.-H. e Chen, Y. (2010). Reevaluating Amdahl's law in the multicore era. *J. Parallel Distrib. Comput.*, 70(2):183–188.
- Temam, R. (1969). Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des fractionnaires. *Archive for Rational Mechanics and Analysis*, 33(5):377–385.
- Thole, C. A. (1985). Experiments with multigrid methods on the caltech-hypercube. *GMD-Studie*, (103).
- Thomas, J. L., Diskin, B. e Brandt, A. (2003). Textbook multigrid efficiency for fluid simulations. *Annual Review of Fluid Mechanics*, 35(1):317–340.
- Timmermans, L. J. P., Minev, P. D. e Van de Vosse, F. N. (1996). An approximate projection scheme for incompressible flow using spectral elements. *International Journal for Numerical Methods in Fluids*, 22(7):673–688.

- Trottenberg, U., Oosterlee, C. e Schüller, A. (2001). *Multigrid*. Academic Press.
- van Kan, J. (1986). A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7(3):870–891.
- Versteeg, H. e Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited.
- Villar, M. M. (2007). *Análise Numérica Detalhada de Escoamentos Multifásicos Bidimensionais*. Tese de doutorado, Universidade Federal de Uberlândia, Uberlândia - MG.
- Wallin, D., Löf, H., Hagersten, E. e Holmgren, S. (2006). Multigrid and gauss-seidel smoothers revisited: Parallelization on chip multiprocessors. Em *Proceedings of the 20th Annual International Conference on Supercomputing*, ICS '06, páginas 145–155, New York, NY, USA. ACM.
- Wesseling, P. (2004). *An Introduction to Multigrid Methods*. An Introduction to Multigrid Methods. R.T. Edwards.
- Xu, J. (1992). Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613.
- Yackel, J. e Meyer, R. R. (1992). Optimal tilings for parallel database design. Em *PARDALOS, P. M. (org.), Advances in Optimization and Parallel Computing*, Amsterdam: Elsevier / North Holland, páginas 293–309.
- Zhang, Q. (2014). A fourth-order approximate projection method for the incompressible Navier–Stokes equations on locally-refined periodic domains. *Applied Numerical Mathematics*, 77:16 – 30.