

UNIVERSIDADE FEDERAL DO PARANÁ
SIMONE DE FÁTIMA TOMAZZONI GONÇALVES

**ESTUDO DE PARÂMETROS DO MÉTODO *MULTIGRID* GEOMÉTRICO PARA
EQUAÇÕES 2D EM CFD E VOLUMES FINITOS**

CURITIBA

2013

SIMONE DE FÁTIMA TOMAZZONI GONÇALVES

**ESTUDO DE PARÂMETROS DO MÉTODO *MULTIGRID* GEOMÉTRICO PARA
EQUAÇÕES 2D EM CFD E VOLUMES FINITOS**

Tese apresentada como requisito parcial para a obtenção do título de doutor em Engenharia Mecânica no Programa de Pós-Graduação em Engenharia Mecânica, Setor de Tecnologia, Universidade Federal do Paraná, na área de concentração Fenômenos de Transporte e Mecânica dos Sólidos.

Orientador: Prof. Dr. Carlos Henrique Marchi.

Co-orientadores:

Prof. Dr. Marcio Augusto Villela Pinto

Prof. Dr. Luciano Kiyoshi Araki

CURITIBA

2013

Gonçalves, Simone de Fátima Tomazzoni

Estudo de parâmetros do método multigrid geométrico para equações 2D em CFD e volumes finitos / Simone de Fátima Tomazzoni Gonçalves. – Curitiba, 2014

125 f. : il.; tabs., grafs.

Tese (doutorado) – Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Mecânica.

Orientador: Carlos Henrique Marchi

Coorientadores: Marcio Augusto Villela Pinto, Luciano Kiyoshi Araki

Bibliografia: p. 103-109

1. Dinâmica dos fluidos. 2. Modelos matemáticos. 3. Multigrid
I. Marchi, Carlos Henrique. II. Pinto, Marcio Augusto Villela. III.
Araki, Luciano Kiyoshi. IV. Título.

CDD 620.1064

TERMO DE APROVAÇÃO

SIMONE DE FÁTIMA TOMAZZONI GONÇALVES


ESTUDO DE PARÂMETROS DO MÉTODO MULTIGRID GEOMÉTRICO PARA EQUAÇÕES 2D EM CFD E VOLUMES FINITOS

Tese aprovada como requisito parcial à obtenção do grau de Doutora em Engenharia Mecânica do Curso de Doutorado do Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal do Paraná, área de concentração Fenômenos de Transporte e Mecânica dos Sólidos.


Banca Examinadora:



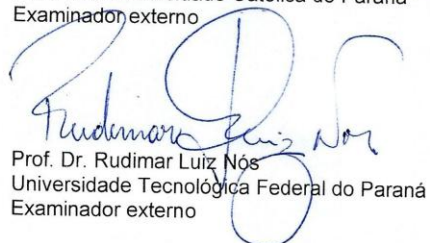
Prof. Dr. Admilson Teixeira Franco
Universidade Tecnológica Federal do Paraná
Examinador externo




Prof. Dr. Roberto Dalledone Machado
Pontifícia Universidade Católica do Paraná
Examinador externo



Prof. Dr. Ricardo Carvalho de Almeida
Universidade Federal do Paraná
Examinador externo



Prof. Dr. Rudimar Luiz Nób
Universidade Tecnológica Federal do Paraná
Examinador externo



Prof. Dr. Carlos Henrique Marchi
Universidade Federal do Paraná
Presidente da Banca Examinadora

Curitiba, 26 de fevereiro de 2013.

Ao meu filho Pedro.

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Carlos Henrique Marchi, por ter aceitado me orientar neste trabalho e pelo conhecimento compartilhado.

Agradeço aos meus co-orientadores, Prof. Dr. Marcio Augusto Villela Pinto e Luciano Kiyoshi Araki, pela dedicação.

Agradeço aos membros da banca examinadora, Prof. Dr. Admilson Teixeira Franco, Prof. Dr. Rudimar Luiz Nos, Prof. Dr. Roberto Dalledone Machado e Prof. Dr. Ricardo Carvalho de Almeida, pelo tempo dispensado à leitura deste trabalho.

Agradeço ao Programa de Pós-Graduação em Engenharia Mecânica (PG-Mec) da Universidade Federal do Paraná (UFPR) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela oportunidade de cursar o doutorado.

Agradeço aos colegas do LENA pela amizade e disposição em ajudar.

Agradeço ao meu marido Nestor Saavedra pelo amor e apoio incondicionais.

RESUMO

A influência de alguns parâmetros do método *multigrid* geométrico sobre o tempo de CPU para três diferentes modelos matemáticos bidimensionais do escopo da CFD (*Computational Fluid Dynamics*) é investigada. Os modelos matemáticos são: a equação de Laplace, a equação de Advecção-Difusão e as Equações de Burgers. Os parâmetros em estudo são: número de iterações internas do *solver* (v); número de malhas (L); número de incógnitas (N); *solvers* e operadores de prolongação. O *multigrid* é empregado com esquema FAS (*Full Approximation Scheme*) e técnica FMG (*Full Multigrid*) com ciclo V e razão de engrossamento $r = 2$. As equações diferenciais são discretizadas pelo Método dos Volumes Finitos (MVF) em geometrias simples e malhas bidimensionais uniformes por direção, com aproximações de 2ª ordem CDS e correção adiada. As condições de contorno, do tipo Dirichlet, são aplicadas mediante a técnica de volumes fictícios. Os sistemas de equações algébricas são resolvidos com o emprego do *solver* Gauss-Seidel Lexicográfico (GS-Lex) e, no caso do problema de Burgers, também com o emprego do Gauss-Seidel *red-black* (GS-RB). Verificou-se principalmente que: o esquema FAS-FMG é cerca de duas vezes mais rápido do que o FAS padrão; que o número de equações ou complexidade do problema não interfere na eficiência do *multigrid*; que o operador de prolongação bilinear é o mais eficiente para interpolar as soluções entre os níveis do FMG.

Palavras-chave: Dinâmica dos fluidos computacional. *Multigrid*. Volumes finitos. Métodos numéricos. Equações de Burgers.

ABSTRACT

This work investigates the influence of some parameters from the Multigrid Geometric method over CPU processing time for three different mathematical bidimensional methods that make up the Computational Fluid Dynamics scope. These mathematical models are: Laplace equation, Advection-Diffusion equation and Burgers' equations. In order to achieve the main target, which consists on optimize the employed algorithms to solve the problems above, the computational time minimization is sought through parameters modifications at the algorithms. The considered parameters are: number of solver's internal iteration (v); number of grids (L); number of incognites (N); solvers and prolongation operators. The multigrid is employed besides FAS (Full Approximation Scheme) and FMG (Full Multigrid) technique, with V cycle and coarsening ratio $r = 2$. The differential equations discretization is made by the Finite Volume Method (MVF) over simple geometries and direction uniform bidimensional grids, with second order CDS and delayed correction. The Dirichlet type boundary conditions are applied through fictitious volume technique. The system of algebraic equations are solved by the Gauss-Seidel Lexicographic (GS-Lex) solver and, at the Burgers problem, the Gauss-Seidel red-black (GS-RB) is also employed. The main results that should be emphasized are: the FAS-FMG scheme is about twice faster than the standard FAS; the multigrid efficiency ate not affected by the number of equations or complexity of the problem; the bilinear prolongation operator is the most efficient to interpolate the solution among the FMG levels.

Keywords: computational fluid dynamics. Multigrid. Finite volume method. Numerical methods. Burgers' Equation.

Lista de Abreviaturas e Siglas

ADI	<i>Alternate Direction Implicit Method</i>
AMG	<i>Algebraic Multigrid</i>
CDS	<i>Central Differencing Scheme</i>
CFD	<i>Computational Fluid Dynamics</i> (Dinâmica de Fluidos Computacional)
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
CS	<i>Correction Scheme</i>
EDP	Equação Diferencial Parcial
FAS	<i>Full Approximation Scheme</i>
FMG	<i>Full Multigrid</i>
GMG	<i>Geometrical Multigrid</i>
GS-Lex	Gauss-Seidel Lexicográfico
GS-RB	Gauss-Seidel <i>Red-Black</i>
LU	<i>Lower and Upper</i>
MDF	Método das Diferenças Finitas
MEF	Método dos Elementos Finitos
MG	<i>Multigrid</i>
MSI	<i>Modified Strongly Implicit Method</i>
MVF	Método dos Volumes Finitos
QML	Quantidade de Movimento Linear
QUICK	<i>Quadratic upstream interpolation for convective kinematics</i>
S	<i>Speed up</i> (fator de aceleração)
SG	<i>Singlegrid</i>
SOR	<i>Successive Over-Relaxation</i>
SUDS	<i>Skew Weighted Upstream Differencing Scheme</i>

TDMA	<i>Tridiagonal Matrix Algorithm</i>
UDS	<i>Upwind Differencing Scheme</i>
VC	Volume de Controle
WUDS	<i>Weighted Upstream Differencing Scheme</i>
WUDS-E	<i>Weighted Upstream Differencing Scheme-Extended</i>

Lista de Símbolos

a_{ij}	coeficientes da matriz A
\mathbf{A}	matriz dos coeficientes
c_p	calor específico
\mathbf{e}	erro algébrico
\mathbf{f}	vetor independente do sistema linear
h	espaçamento entre os pontos da malha
\mathbf{I}_h^{2h}	operador de restrição
\mathbf{I}_{2h}^h	operador de prolongação
l	comprimento do domínio de cálculo
L	número de níveis
N	número de incógnitas
p	pressão estática
Pe	número de Peclet
Ra	número de Rayleigh
Re	número de Reynolds
\mathbf{r}	Resíduo
\mathbf{S}	matriz de iteração do método iterativo
s_p	<i>Speedup</i>
u	velocidade na direção x

v	velocidade na direção y
\mathbf{v}	solução aproximada do sistema
\mathbf{w}	autovetor

Letras Gregas

Λ	sequência de malhas
λ	autovalor
μ_d	viscosidade dinâmica do fluido
ν_c	viscosidade cinemática do fluido
ν	número de iterações internas do MG
ρ	massa específica do fluido
ϕ	solução numérica da variável de interesse
Φ	solução exata da variável de interesse
ω	parâmetro de relaxação
Ω	conjunto de índices

Subscritos e superscritos

i	posição do nó na direção x
j	posição do nó na direção y

h	malha fina
$2h$	malha grossa

Sumário

CAPÍTULO 1

INTRODUÇÃO	17
1.1 Motivação	21
1.2 Objetivos	23
1.3 Organização do Trabalho	24

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA	26
2.1 Resolução de problemas em CFD	26
2.2 O método <i>multigrid</i>	28
2.3 Equação de Burgers	32

CAPÍTULO 3

FUNDAMENTOS TEÓRICOS	36
3.1 Formulação em Volumes Finitos	36
3.2 Condições de contorno	40
3.2.1 Balanço para os volumes da fronteira	40
3.2.2 Com volumes fictícios	41
3.2.3 Com meios volumes	42
3.3 Métodos iterativos básicos	42

3.4 O multigrid	50
3.4.1 Equação residual e esquema de correção	51
3.4.2 Razão de refinamento	53
3.4.3 Operadores de restrição	54
3.4.4 Prolongação	56
3.4.5 Ciclos	57
3.4.6 FMG	58
3.4.7 Algoritmos	59

CAPÍTULO 4

MODELOS MATEMÁTICOS E NUMÉRICOS	62
4.1 Modelos matemáticos	62
4.1.1 Equação de Laplace	62
4.1.2 Equação de Advecção-Difusão	63
4.1.3 Equações de Burgers	64
4.2 Modelos numéricos	66
4.2.1 Discretização das equações algébricas	67
4.2.1.1 Equação de Laplace	67
4.2.1.2 Advecção-Difusão	68
4.2.1.3 Equações de Burgers	69
4.2.2 Resolução dos sistemas de equações algébricas	72
4.2.3 Operadores de transferência	72
4.3 Dados da implementação computacional	79

CAPÍTULO 5

RESULTADOS	80
5.1 ETAPA I: Laplace, Advecção-Difusão e Burgers	82
5.1.1 – Número de iterações internas (v)	82
5.1.2 Número de malhas (L)	84
5.1.3 Número de incógnitas (N)	85
5.1.4 Comparação entre os esquemas FAS e FAS-FMG	86
5.1.5 Esforço computacional	88
5.2 ETAPA II – Equações de Burgers	89
5.2.1 Comparativo GS-Lex e GS-RB	89
5.2.2 Comparativo de operadores de prolongação	92
5.2.2.1 Operadores de prolongação para a solução de cada nível do FMG	92
5.2.2.2 Operadores de prolongação para as correções	95

CAPÍTULO 6

CONCLUSÃO	99
REFERÊNCIAS BIBLIOGRÁFICAS	103

APÊNDICE A

VERIFICAÇÃO DOS CÓDIGOS	110
A.1 Verificação e validação	110
A.2 Verificação dos códigos	112

A.3 Erro de discretização	113
A.4 Ordem de convergência dos métodos SG e MG	118

APÊNDICE B

OPERADOR BICÚBICO	120
B.1 Equações do operador bicúbico	120

CAPÍTULO 1

INTRODUÇÃO

A Dinâmica dos Fluidos envolve a modelagem de fenômenos físico-químicos nas áreas de mecânica dos fluidos, transferência de calor e massa e combustão, entre outras, que são representados por modelos matemáticos. De um modo geral, estes modelos matemáticos consistem em equações demasiadamente complexas para serem resolvidas analiticamente. Para sobrepor esta dificuldade, muitos recursos são investidos no desenvolvimento de tecnologias para simulação experimental dos fenômenos envolvidos em tais aplicações. Porém, abordagens experimentais podem ser de difícil execução ou excessivamente dispendiosas. Diante disto, e aliada à crescente disponibilidade de recursos computacionais, a simulação numérica vem se tornando uma componente essencial nos projetos industriais (MALISKA, 2004).

A Dinâmica dos Fluidos Computacional (do inglês, *Computational Fluid Dynamics* – CFD) trata dos estudos de métodos computacionais para simulação de fenômenos que envolvem fluidos em movimento com ou sem trocas de calor, cujo interesse principal é obter grandezas físicas, como velocidade, temperatura e pressão, na região do escoamento (FORTUNA, 2000). Para Versteeg e Malalasekera (2007), CFD é uma técnica muito poderosa com forte presença em muitas aplicações industriais e não industriais. Alguns exemplos destas aplicações estão relacionados à aerodinâmica de aviões e veículos, hidrodinâmica de navios, máquinas de potência, turbomáquinas, modelagem de polímeros, refrigeração de equipamentos, dispersão de poluentes e previsões meteorológicas.

Em CFD, as simulações computacionais são baseadas em métodos numéricos que aproximam um sistema de equações diferenciais parciais por um sistema de equações algébricas, lineares, em grande parte dos casos, do tipo

$$\mathbf{A}\phi = \mathbf{f}, \quad (1.1)$$

onde \mathbf{A} é a matriz dos coeficientes, ϕ é o vetor das incógnitas e \mathbf{f} é o vetor dos termos independentes.

Os métodos tradicionais empregados neste processo são o Método das Diferenças Finitas (MDF), o Método dos Elementos Finitos (MEF) e o Método dos Volumes Finitos (MVF) (MALISKA, 2004). Dessa forma, a solução passa a ser obtida para um número discreto de pontos definidos por uma malha, assumindo-se um determinado erro de discretização que, de acordo com Marchi (2001), é a diferença entre a solução analítica exata e a solução numérica em uma determinada malha.

A resolução por meio de métodos numéricos requer o emprego de técnicas numéricas eficientes para que se possam obter soluções aceitáveis e com o menor tempo de CPU¹. Independentemente da técnica empregada, o maior custo computacional associado a uma simulação numérica deve-se à solução do sistema linear da Eq. (1.1). Em consequência disto, o foco de muitos estudos tem sido na direção da minimização do tempo de CPU necessário para resolver tal sistema linear, sem afetar a qualidade da solução numérica.

Após a obtenção das equações algébricas o problema é então resolvido por algum método de resolução de sistemas lineares, que neste trabalho será designado *solver*. Duas categorias podem ser consideradas: a dos *solvers* diretos e a dos *solvers* iterativos.

Os *solvers* diretos fornecem, após um número finito de operações, a solução exata do sistema de equações, exceto por erros de arredondamento. Nesta categoria estão os métodos de eliminação de Gauss, fatoração LU (do inglês *Lower and Upper*) e TDMA (*Tridiagonal Matrix Algorithm*), para citar alguns. Devido à necessidade de armazenamento de dados e do alto tempo de CPU, os *solvers* diretos podem ser inviáveis para obter soluções em malhas muito refinadas. Além disto, nestes casos, os erros de arredondamento podem prejudicar a qualidade da solução.

Os *solvers* iterativos baseiam-se em uma estimativa inicial para a solução, a partir da qual é construída uma sequência de aproximações que, mediante certas condições, converge para a solução exata. Os sistemas lineares que aproximam as equações diferenciais em CFD são, dependendo do método de discretização empregado, esparsos e de grande porte, onde o número de incógnitas é da ordem de milhares ou ainda superior. Considerando-se estas características, o emprego de um *solver* iterativo para sua resolução é o mais apropriado, devido principalmente à sua potencialidade quanto à otimização do armazenamento e eficiência computacional (BURDEN e FAIRES, 2003). Alguns exemplos deste tipo de *solver* são: Jacobi, Gauss-Seidel (BURDEN e FAIRES, 2003), ADI (*Alternate Direction Implicit*

¹ Tempo gasto pela Unidade Central de Processamento (do inglês, *Central Processing Unit – CPU*).

Method) (FERZIGER e PERIC, 2002) e MSI (*Modified Strongly Implicit Method*) (SCHNEIDER e ZEDAN, 1981).

No início do processo de resolução do sistema linear a taxa de convergência dos *solvers* iterativos é relativamente alta. Este rápido decréscimo do erro durante as primeiras iterações se deve à propriedade de suavização dos métodos iterativos. Devido a esta propriedade, o *solver* elimina eficientemente as componentes de alta frequência do erro. Quando restam apenas as componentes de baixa frequência do erro o processo torna-se lento, uma vez que as mesmas são difíceis de serem removidas (BRIGGS *et al.*, 2000).

O método *multigrid* (MG) tem se mostrado muito eficiente para acelerar a taxa de convergência dos *solvers* iterativos empregados na resolução de sistemas lineares (BRIGGS *et al.*, 2000; HACKBUSCH, 1985). Sua eficiência pode ser compreendida por meio da análise espectral dos erros da solução. Diversos trabalhos na literatura demonstram que esses erros podem ser representados através da superposição de ondas de diferentes frequências (BRIGGS *et al.*, 2000; TROTTEBERG *et al.*, 2001). Além disso, pode-se mostrar que apenas os erros que possuem comprimento de onda da ordem do espaçamento da malha é que são reduzidos com rapidez nos *solvers* iterativos (BRIGGS *et al.*, 2000). Os outros erros, de menores frequências para esta malha, são reduzidos com muita dificuldade e são os causadores da estagnação do *solver*. Esta observação indica que diferentes malhas podem ser necessárias para reduzir todo o espectro de frequências do erro.

O método *multigrid* consiste no uso de várias malhas com diferentes graus de refinamento, as quais são percorridas ao longo do processo iterativo. Assim, a cada nível de refinamento da malha, as diferentes componentes do erro são eficientemente suavizadas, acelerando a convergência do processo iterativo. A transferência de informações entre as malhas é feita mediante operadores de restrição (malha fina para a malha grossa) e prolongação (malha grossa para a malha fina) e a sequência com que as malhas são visitadas define diferentes ciclos, designados como V, W e F. Na Fig. 1.1 tem-se uma representação do método *multigrid* com três malhas e na Fig. 1.2 uma descrição esquemática do ciclo V.

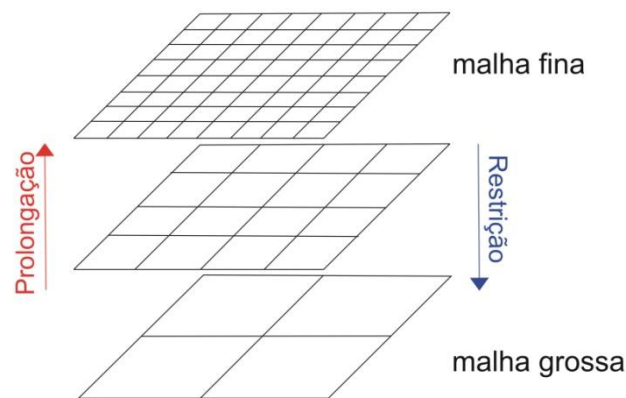


Figura 1.1: Ilustração do método *multigrid* com três malhas. (adaptado de Feng *et al.*, 2012)

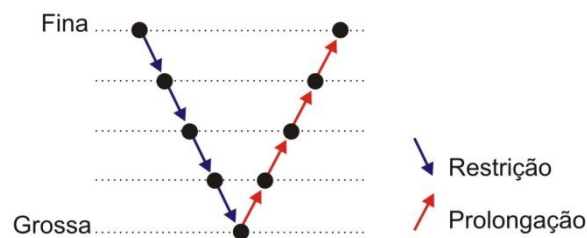


Figura 1.2: Ilustração ciclo V com cinco malhas

Uma boa estimativa inicial para o processo iterativo pode ser obtida interpolando-se uma solução da malha grossa para a malha fina. Esta estratégia dá origem ao chamado *Full Multigrid* (FMG). A ideia do FMG está representada na Fig. (1.3): inicia-se o processo iterativo na malha mais grossa. Então, a solução é interpolada para a segunda malha mais grossa e um ciclo V é percorrido. Estes dois passos são repetidos recursivamente até que a malha mais fina possível seja atingida.

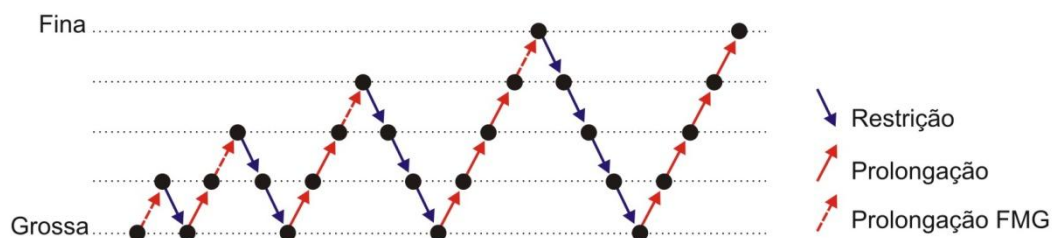


Figura 1.3: Ilustração do FMG com cinco malhas

No âmbito do *multigrid* são distinguidos o *multigrid* geométrico (*Geometrical Multigrid* - GMG) e o algébrico (*Algebraic Multigrid* - AMG), indicados para malhas

estruturadas e não estruturadas, respectivamente. Para ambos os tipos, há dois modos de se construir o algoritmo MG, de acordo com a forma que o sistema de equações algébricas é operado nas malhas mais grossas: o Esquema de Aproximação Completa (do inglês, *Full Approximation Scheme* – FAS), em contraponto ao Esquema de Correção (do inglês, *Correction Scheme* – CS). No esquema CS somente o resíduo é transferido para as malhas grossas, onde é resolvida a equação residual. Já no esquema FAS são transferidos o resíduo e a aproximação da solução para as malhas grossas. (TROTTEMBERG *et al.*, 2001).

1.1. Motivação

Para obter soluções de boa qualidade em CFD, deve-se reduzir o erro de discretização através do emprego de malhas muito refinadas. De acordo com Hortmann e Peric (1990), simulações com mais de 100 volumes em cada direção coordenada são extremamente caras e podem não ser suficientemente acuradas. Mesmo com a oferta atual de computadores com grande capacidade de processamento e memória, ainda não se chegou ao estágio em que restrições de velocidade de processamento e de armazenamento de dados possam ser ignoradas. Hirsch (1988) afirmou que no caso ideal o método *multigrid* tem complexidade $O(N)$, ou seja, o tempo de CPU é linearmente proporcional ao número de pontos da malha. Esta característica o torna um método de grande relevância na solução de problemas em CFD, uma vez que possibilita simulações em malhas muito refinadas.

No entanto, as dificuldades associadas a problemas não elípticos ou com algum termo não elíptico, como exemplo os descritos pelas equações de advecção-difusão e Burgers, ou com perturbações singulares, frequentemente causam deterioração significativa na eficiência do método *multigrid*, resultando em maior tempo computacional (ZHANG, 1997). De acordo com Brandt (1977) e Wesseling e Oosterlee (2001), a eficiência ideal do método *multigrid* ainda não foi obtida para problemas de CFD em geral. Segundo Ferziger e Peric (2001) no contexto geral do *multigrid*, muitos parâmetros podem ser escolhidos mais ou menos arbitrariamente, e a taxa de convergência depende dessas escolhas. Além disso, os maiores ganhos com o método *multigrid* são obtidos em problemas totalmente elípticos, e os menores em problemas dominados pela advecção. Fatores de aceleração² (S) ou *speed up* típicos estão

² O fator de aceleração (S) é definido como a razão entre o tempo de CPU do método de malha única *singlegrid* (SG) e do tempo de CPU do *multigrid*.

na faixa de 10 a 100 quando cinco níveis de malha são usados. Tannehill *et al.* (1997) encontraram $S = 325$ para a equação de Laplace bidimensional, com malha 129×129 pontos. Teoricamente, o S deveria se manter igual em qualquer problema. Entretanto, para $Re = 100$ e 1000 , Ferziger e Peric (2001) obtiveram valores de $S = 42$ e 11 , respectivamente, para o escoamento bidimensional descrito pelas equações de Navier-Stokes, malha 128×128 volumes. Estes resultados demonstram a redução da eficiência do método *multigrid* com relação ao número de Reynolds e com relação à equação de Laplace.

Brandt *et al.* (2002) afirmaram que um ganho de mais de duas ordens de magnitude na redução do tempo de CPU ainda pode ser alcançado com o *multigrid*. Para escoamentos de fluido incompressível já foram desenvolvidos métodos *multigrid* relativamente eficientes, mas essa eficiência ainda pode ser melhorada em aproximadamente uma ordem de magnitude. Jimac (2007) afirma que devido os avanços nos últimos trinta anos, inicialmente para problemas lineares e posteriormente para problemas não lineares, o *multigrid* tornou-se a escolha natural para muitos problemas de escoamentos.

Com o FMG, uma boa estimativa inicial para a malha fina é produzida ao custo de poucas iterações na malha grossa. Adicionalmente, os erros são suavizados eficientemente garantindo uma ótima taxa de convergência e fazendo do FMG a mais eficiente versão do *multigrid*. (TROTTEMBERG *et al.*, 2001; THEKALE *et al.*, 2010; ZHANG *et al.*, 2010).

De acordo com Trottenberg *et al.* (2001), experiências com o método *multigrid* mostram que as escolhas de componentes como *solver*, número de iterações do *solver* em cada ciclo, tipo de ciclo, operadores de restrição e prolongação, podem ter uma grande influência sobre a taxa de convergência do algoritmo. Assim, uma boa combinação de parâmetros pode tornar-se essencial na redução do tempo de CPU, o que justifica a importância do estudo e identificação de parâmetros ótimos do método *multigrid*. Wallis (2008) aponta a escolha do *solver*, a estimativa inicial e o número de malhas, dentre outros, como aspectos importantes a serem considerados na aplicação do método *multigrid*.

As ótimas propriedades teóricas do *multigrid* com relação à aceleração da convergência dos métodos iterativos têm motivado o desenvolvimento de vários trabalhos com o foco na análise de parâmetros do método *multigrid* em problemas de CFD. O grupo de pesquisa em CFD, da Universidade Federal do Paraná, tem dedicado esforços no estudo de parâmetros do *multigrid*, sendo os principais resultados encontrados em Pinto *et al.* (2005), Pinto e Marchi (2006, 2007), Oliveira *et al.* (2006, 2012), Santiago e Marchi (2007, 2008), Santiago *et al.* (2010, 2011) e Suero *et al.* (2012). Outras análises podem ser encontradas em

Ferziger e Peric (2001), Mesquita e De Lemos (2004), Souza *et al.* (2006) e Thekale *et al.*, (2010).

Darbandi (2010) afirmou que um dos principais fatores que melhora o desempenho do *multigrid* é o quão bem as informações da malha grossa são transferidas para a malha imediatamente mais refinada. Por sua vez, Liu (2011) diz que a escolha de um bom operador de prolongação é sempre uma tarefa difícil para o método *multigrid* e considera que o estudo destes operadores ainda é um problema em aberto. De acordo com o que costuma ser recomendado (BRIGGS *et al.*, 2000; TROTTEBERG *et al.*, 2001) o operador bicúbico é o mais comum na prolongação da solução de um nível para outro no FMG (THEKALE *et al.*, 2010; WANG e ZHANG, 2009). Contudo, não foi encontrado nenhum estudo sistemático do efeito desses operadores sobre o tempo de CPU, comparados a outros operadores de menor ordem.

As equações de Burgers 2D são consideradas um modelo de referência no estudo de escoamentos e testes de metodologias numéricas (EFE, 2006). Devido à sua ampla aplicabilidade constitui ainda um campo de pesquisa bastante ativo (DAWHAN *et al.*, 2012, ZHAO *et al.*, 2011). Dentre os trabalhos encontrados que envolvem o método *multigrid* aplicado a estas equações, poucos são os que objetivam melhorar a convergência do método *multigrid*. Nestes trabalhos, os parâmetros investigados são: operador de restrição (FERM e LÖTSTEDT, 1997), razão de refinamento (FAURE *et al.*, 2005) e número de iterações do *solver* (SANTIAGO, 2010). Além desses estudos, Zhang *et al.* (2010) propuseram um *multigrid* espectral onde avaliaram diferentes esquemas e o efeito do número de malhas empregadas na convergência.

Neste contexto, este trabalho concentra-se no estudo de alguns parâmetros do método *multigrid* para modelos matemáticos importantes em CFD.

1.2. Objetivos

Neste trabalho é feita a análise de parâmetros do método *multigrid* geométrico e o desenvolvimento de códigos computacionais eficientes para os seguintes modelos matemáticos bidimensionais: as equações de Laplace, Advecção-Difusão e Burgers.

Os parâmetros a serem analisados para fins de redução do tempo de CPU são: número de iterações internas do *solver*, número de malhas, número de incógnitas e o operador de prolongação.

Os objetivos específicos estão resumidos nos tópicos elencados abaixo.

Para os três problemas citados, após programar o método *multigrid* com o algoritmo FAS-FMG:

- Comparar o desempenho do MG com os algoritmos FAS e FAS-FMG.
- Verificar a influência do número de iterações do *solver* sobre o tempo de CPU e identificar o valor ótimo para este parâmetro.
- Verificar a influência do número malhas sobre o tempo de CPU e identificar o valor ótimo para este parâmetro.
- Verificar a influência do número de incógnitas sobre o tempo de CPU e analisar eficiência do código implementado.
- Verificar se os parâmetros ótimos obtidos com o MDF, disponíveis na literatura, são mantidos quando as equações são discretizadas com o MVF.

Para as equações de Burgers, adicionalmente, são considerados:

- Comparar os *solvers* Gauss-Seidel Lexicográfico (GS-Lex) e Gauss-Seidel *red-black* (GS-RB) para o problema modelado pelas equações de Burgers.
- Testar operadores de interpolação na prolongação das soluções do FMG e identificar aquele que resulta no melhor desempenho do *multigrid*.
- Testar operadores de interpolação para as correções ao longo do ciclo V e identificar aquele que resulta no melhor desempenho do *multigrid*.

1.3. Organização do Trabalho

Além deste primeiro capítulo, esta tese está distribuída em mais cinco capítulos, estruturados de forma que as informações são apresentadas sequencialmente visando facilitar o entendimento do leitor. O Cap. 2 contém uma revisão bibliográfica que aborda algumas generalidades sobre CFD e descreve sucintamente alguns dos trabalhos relatados na literatura e procura contextualizá-los em termos da sua contribuição para o atual estado da arte da simulação numérica em CFD e do *multigrid*. No Cap. 3 são apresentados alguns fundamentos teóricos a respeito do método dos volumes finitos, dos métodos iterativos e do método *multigrid*, incluindo suas componentes e algoritmos. No Cap. 4 são apresentados os modelos matemáticos, bem como um detalhamento do modelo numérico e do processo de obtenção das

equações discretizadas, das aproximações e das condições de contorno empregadas. São também apresentados alguns aspectos relevantes de implementação do método *multigrid* como: tipo de ciclo, esquema, número de malhas, número de iterações do *solver*, critério de parada e tolerância, operadores de restrição e prolongação. Os resultados, comparações e discussões estão reservados para o Cap. 5, enquanto as conclusões finais e sugestões para trabalhos futuros são apresentadas no Cap. 6.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

2.1 Resolução de problemas em CFD

Os métodos de resolução de problemas de interesse em engenharia podem, basicamente, ser classificados em três categorias: métodos experimentais, métodos analíticos e métodos numéricos (MALISKA, 2004). Cada um destes três métodos apresenta vantagens e desvantagens, de acordo com os recursos disponíveis e/ou da formulação matemática do problema a ser resolvido.

Os métodos analíticos e numéricos são classificados como métodos teóricos, uma vez que fazem uso de um modelo matemático. Com relação aos métodos analíticos, uma de suas vantagens é a obtenção da solução de forma fechada ou soluções contínuas. Uma desvantagem é que as hipóteses simplificadoras podem afastar o modelo matemático do modelo físico uma vez que soluções analíticas são possíveis apenas para uma pequena quantidade de problemas e aplicações em engenharia, com geometrias e condições de contorno simples. Os métodos experimentais têm a vantagem de lidar, em escala, com a configuração real do fenômeno físico, mas costumam demandar grandes recursos financeiros. Além disso, nem sempre é possível sua reprodução em laboratório das condições reais do fenômeno e, ainda, questões de segurança podem inviabilizar os experimentos. Por sua vez, os métodos numéricos apresentam menos restrições. Podem ser empregados para resolver problemas definidos em geometrias arbitrárias, com complicadas condições de contorno e apresentam resultados com maior rapidez (MALISKA, 2004). De acordo com Fortuna (2000), em muitas situações os métodos numéricos consistem na forma mais prática, ou até mesmo na única, de se obter informações sobre um determinado escoamento.

Diante do exposto, a crescente demanda por soluções de problemas em engenharia aliada à disponibilidade de recursos computacionais tem impulsionado o desenvolvimento de várias técnicas numéricas, como alternativa ou complemento aos métodos analíticos e experimentais, para a obtenção de soluções aproximadas (MALISKA, 2004).

Assim como as soluções analíticas podem ser afetadas por erros de modelagem, causados pelas simplificações feitas sobre o fenômeno real na concepção do modelo matemático, as soluções numéricas podem, adicionalmente, ser afetadas por erros numéricos.

Uma vez que métodos numéricos originam geralmente soluções aproximadas, há uma diferença entre a solução numérica (ϕ) e a analítica exata (Φ), para qualquer variável de interesse, denominada erro numérico (FERZIGER e PERIC, 2001):

$$E(\phi) = \Phi - \phi. \quad (2.1)$$

Diversas podem ser as fontes de origem dos erros numéricos. Marchi (2001) apresentou a seguinte classificação: erros de truncamento, erros de iteração, erros de arredondamento e erros de programação.

Erros de truncamento têm origem no processo de discretização de um modelo matemático e estão relacionados ao conceito de consistência de um método numérico. De acordo com Maliska (1995), um método é consistente se o erro de truncamento tende a zero com o refinamento da malha, ou seja, a equação discretizada deve tender à equação original, quando o espaçamento entre os pontos da malha tende a zero. A consistência, contudo, não garante que a solução do problema discretizado seja a solução exata para malhas extremamente refinadas; para tanto, é necessário que haja estabilidade do método. A estabilidade em um método numérico é obtida quando o erro numérico tende a um valor finito quando o número de iterações tende ao infinito. Consistência e estabilidade são condições necessárias e suficientes para a convergência. A solução numérica é convergente quando é estável e tende para a solução das equações diferenciais quando a malha é refinada (MALISKA, 1995).

Erros de iteração estão associados ao emprego de métodos iterativos na solução dos sistemas de equações algébricas obtidas no processo de discretização. Considerando-se que a solução do sistema de equações seja única, define-se o erro de iteração como sendo a diferença entre a solução exata e a solução do sistema de equações em uma dada iteração (FERZIGER e PERIC, 2001). De acordo com Martins (2002), pode ser originado por diferentes fatores como: o emprego de métodos iterativos para solução das equações discretizadas, o uso de métodos segregados na obtenção da solução de modelos constituídos por várias equações diferenciais, ou ainda pela existência de não linearidades no modelo matemático (MARTINS, 2002).

Erros de arredondamento estão relacionados à capacidade finita de representação de certo número por um computador. Desta forma, os números são armazenados levando-se em consideração um número limitado de dígitos, que varia de acordo com a linguagem de programação, o tipo de variável utilizado e o processador empregado (MARTINS, 2002). Erros de programação, por sua vez, são inerentes ao programador e à utilização do código implementado, incluindo basicamente (ROACHE, 1998): os erros resultantes do uso incorreto de um modelo numérico na aproximação de um modelo matemático, os erros gerados na implementação do modelo numérico em um código computacional, os erros cometidos no uso do código durante a solução numérica e quaisquer outras fontes de erro.

A magnitude aceitável para o erro numérico depende, entre outros fatores, da finalidade da solução numérica, dos recursos financeiros envolvidos, do tempo permitido para realizar as simulações e dos recursos computacionais disponíveis. Sabendo-se que as soluções numéricas contêm erros, é importante estimá-los pelos seguintes motivos (MARCHI, 2001):

- Quando o erro é maior que o aceitável, compromete-se a confiabilidade do uso da solução numérica;
- Quando o erro é menor que o necessário, há desperdício de recursos computacionais (tempo de processamento e quantidade de memória);
- Para validar e desenvolver modelos matemáticos que visem explicar modelos físico-químicos ainda não modelados adequadamente e cujas soluções analíticas são desconhecidas;
- Para otimizar o uso da malha, isto é, adaptá-la visando homogeneizar o nível de erro no domínio de cálculo; para evitar interpretações equivocadas.

2.2 O método *multigrid*

O *multigrid* pode ser interpretado como uma técnica numérica empregada para acelerar a convergência dos métodos iterativos. De acordo com a literatura, o *multigrid* é considerado um dos métodos mais eficientes e gerais desenvolvidos nos últimos anos, sem restrições ao tipo de malha, à técnica de discretização ou à simetria do sistema de equações (BRIGGS *et al.*, 2000; HIRSCH, 1988; TANNEHILL *et al.*, 1997; TROTTERBERG *et al.*, 2001).

Os métodos *multigrid* começaram a ser investigados a partir da década de 1960 com os trabalhos de Fedorenko (1964) e Bakhvalov (1966). Porém, foi apenas nas décadas de 1970 e 1980 que ganharam popularidade a partir dos trabalhos de Brandt (1973 *apud* TROTTEMBERG *et al.*, 2001), quando os métodos *multigrid* geométricos foram introduzidos. Neste tipo de *multigrid*, o processo de formação das malhas mais grossas se dá estritamente em função de aspectos geométricos da malha original, sendo necessária a discretização das equações nas malhas mais grossas também. Posteriormente, as dificuldades associadas à discretização das equações nas malhas grosseiras motivaram o desenvolvimento dos métodos *multigrid* algébricos, que foram introduzidos por Brandt *et al.* (1982 *apud* CLEARY *et al.*, 2000) nos quais não é necessário repassar nenhuma informação a respeito da malha e do esquema de discretização.

Brandt (1977) apresentou uma análise teórica e numérica que incluiu uma introdução do método e do esquema FAS, específico para problemas não lineares. O trabalho também apresenta estudos sobre razões de refinamento, análise local de Fourier, taxa de convergência e a sistematização da ideia do FMG, entre outras análises. A partir deste trabalho de referência surgiram muitos trabalhos com o método *multigrid*, abordando o aperfeiçoamento do método e a resolução de novos problemas em CFD. Aplicações do método *multigrid* podem ser encontradas em Tannehill *et al.* (1997) e Trottenberg *et al.* (2001) e Ferziger e Peric (2002).

As principais dificuldades computacionais observadas com o *multigrid* (escoamentos recirculantes, características da malha, anisotropias, pontos de estagnação, discretização e relaxação próximos a contornos, etc.) foram resumidas por Brandt (1998), que apresenta uma extensa tabela onde são apontadas algumas possibilidades para contornar tais dificuldades. Jimac (2007) demonstrou que o método *multigrid* pode ser combinado com sucesso com outras técnicas numéricas modernas, como malhas adaptativas, computação paralela e discretizações de alta ordem, para uma variedade de problemas não lineares em CFD. No entanto, afirma que ainda existem obstáculos serem superados com o emprego dessas técnicas em problemas em regime transiente.

Wesseling e Oosterlee (2001) fizeram uma revisão do desenvolvimento do *multigrid* geométrico na década de 90, enfatizando aplicações em CFD e apresentando o estado da arte para escoamentos compressíveis e incompressíveis. Apontaram o método *multigrid* como um dos mais importantes desenvolvimentos em análise numérica na segunda metade do século XX. Análise semelhante, relativa ao mesmo período, foi feita por Stüben (2001) enfocando o método *multigrid* algébrico. Neste trabalho, Stüben afirmou que apesar do grande número de

métodos desenvolvidos, ainda nenhum deles seria capaz de tratar com eficiência todos os problemas práticos em CFD. Brandt *et al.*, (2002) apresentaram os recentes avanços do método *multigrid* e abordaram algumas estratégias para atingir a sua eficiência teórica.

Vários trabalhos têm identificado as principais vantagens do *multigrid*, por meio de análises teóricas ou numéricas. Wesseling (1984), Hortmann *et al.* (1990) e Ferziger e Peric (2001) destacaram como propriedade mais importante do método o fato de que o número de iterações necessárias para alcançar a convergência na malha mais refinada é independente do número de pontos da malha.

Uma boa introdução ao método *multigrid*, onde são abordados tópicos como análise local de Fourier, para os algoritmos CS e FAS, *solvers*, operadores de transferência entre malhas, tipos de ciclo, complexidade e aplicações, pode ser encontrada em Wesseling (1992), Briggs *et al.* (2000) e Trottenberg *et al.* (2001). Um dos autores que mais contribuiu para o desenvolvimento do *multigrid* nas últimas três décadas foi Achi Brandt, do Weizmann Institute of Science. Suas principais publicações podem ser encontradas em <http://www.wisdom.weizmann.ac.il/~achi/>. Um repositório de informações gerais sobre o método *multigrid*, contendo teses, artigos, programas, além de um arquivo com uma grande quantidade de referências pode ser encontrado no sítio <http://www.mgnet.org/>.

Visando reduzir o tempo de CPU, vários trabalhos têm sido publicados. Nestes trabalhos, o objetivo é melhorar a eficiência do método *multigrid* por meio de estudos de parâmetros para problemas específicos de CFD. Tannehill *et al.* (1997) afirmaram que o melhor desempenho do método *multigrid* é obtido com diversos níveis de malha e sugerem o uso de 4 ou 5 para o problema de Laplace 2D com malha mais fina de 129 x 129 pontos. Rabi e De Lemos (2001) apresentaram alguns parâmetros relativos ao melhor desempenho do método *multigrid* para o problema de advecção-difusão 2D, com esquema CS e MVF. Indicaram o número de iterações internas do *solver* igual a 1 para o ciclo V e 2 para o ciclo W. Mesquita e De Lemos (2004) afirmaram que parâmetros ideais para o número de iterações internas e número de malhas empregadas não podem ser facilmente determinados e que podem depender de características específicas do escoamento em questão. Pinto e Marchi (2006) fizeram uma análise dos esquemas CS e FAS com três *solvers* para o problema Laplace 2D e MDF. Eles constataram que o ideal é empregar o número máximo de malhas possível e que, além disso, o esquema FAS é mais rápido que o esquema CS, mesmo para problemas lineares. Santiago e Marchi (2008) analisaram a influência dos esquemas CS e FAS e do número de equações diferenciais sobre o desempenho do método *multigrid* para os

problemas bidimensionais de Laplace, Navier e Burgers, com MDF. Afirmaram que o esquema CS é mais rápido do que o FAS, para os problemas lineares estudados, e que o número de equações diferenciais envolvidas no problema não afeta o desempenho do *multigrid*. Kumar *et al.* (2009) resolveram o clássico problema da cavidade com tampa móvel na formulação em variáveis primitivas e MVF e apresentaram resultados para malhas com $N = 129 \times 129$, 257×257 e 513×513 volumes de controle, para diferentes Re. Afirmaram que o emprego de 2 a 4 malhas são suficientes para problemas de tamanho $N = 513 \times 513$ e que o emprego de um maior número de malhas não resultou em um ganho significativo. No entanto, não há esclarecimento se o relativo ganho é na taxa de convergência. Suero *et al.* (2012) analisaram alguns parâmetros do AMG para as equações de Laplace e Poisson bidimensionais em malhas quadrangulares e triangulares. Para os três problemas, em ambas as malhas, encontraram que o emprego do número máximo de malhas resulta no menor tempo de CPU. Para o número ótimo de iterações do *solver* obtiveram os valores $v = 1$ e $v = 2$ para malhas triangulares e quadrangulares, respectivamente. Oliveira *et al.* (2012) investigaram o efeito de alguns parâmetros para as equações de Laplace e Poisson, discretizadas pelo MDF, sobre o desempenho do *multigrid*. Verificaram que para problemas isotrópicos, o operador de restrição não afetou o desempenho do *multigrid*, mas para problemas anisotrópicos o operador de ponderação parcial, proposto pelos autores, foi o que apresentou o melhor desempenho. Também observaram que o número ótimo de iterações do *solver* varia com a anisotropia da malha. Para o problema de Laplace isotrópico obtiveram $v = 2$ como valor ótimo.

Considerando que a escolha de uma boa interpolação é uma dificuldade para os métodos *multigrid* e que o seu desenvolvimento ainda é um problema em aberto, Liu (2010) propôs o operador de prolongação *full local* baseado no resíduo, para o MDF. Apresentou testes em malhas de tamanho $N = 64 \times 64$, 128×128 e 256×256 , com a equação bidimensional de Poisson e uma equação elíptica com coeficientes descontínuos. Na malha $N = 256 \times 256$ o novo operador foi 22% mais rápido do que o operador bilinear. Ainda nesta malha, verificou que o número de ciclos V necessários para obter a solução cai de 12 para 9 com o novo operador. Conclui que nos dois casos testados, o operador proposto é mais eficiente do que o operador bilinear. Darbandi *et al.* (2008) consideraram um novo operador de restrição-prolongação na resolução das equações de Navier-Stokes, discretizadas com o MVF baseado em elementos. Realizaram testes em um algoritmo de 2 e 3 malhas e obtiveram, respectivamente, $S = 4,7$ e $S = 9,0$ donde concluíram que o desempenho do

multigrid melhora com o aumento do número de malhas. Neste trabalho não há informações sobre o tamanho das malhas empregadas nos testes.

Drikakis *et al.*(1998) propuseram o operador *mixed* para prolongar as soluções no FMG no problema modelado pelas equações de Navier-Stokes 3D. Realizaram testes comparando este operador com os operadores trilinear, constante por partes e *upwind*. Testaram diferentes combinações destes operadores aplicando-os na prolongação das soluções no FMG ou nas correções entre as malhas no ciclo V, respectivamente: *mixed* com trilinear, trilinear com trilinear, constante por partes com constante por partes e *upwind* com trilinear. Deste estudo concluíram que a combinação do operador *mixed* com o operador trilinear resulta no melhor desempenho, considerando que requer menos ciclos V. Contudo, não são apresentados resultados sobre o tempo de CPU ou fatores de convergência, tampouco há indicação do tamanho da malha empregada na obtenção destes resultados.

Thekale *et al.* (2010) introduziram uma abordagem para otimizar o custo computacional do FMG. Neste trabalho, propuseram um algoritmo que emprega a técnica de otimização *branch and bound* para determinar o número ótimo de ciclos V a cada nível. Apresentaram um teste com a equação de Laplace, para um problema com 7 níveis de malha. Com este algoritmo que se baseia no controle da redução do erro, mostraram que é possível obter um ganho de 35% no tempo de CPU com relação a uma escolha arbitrária deste parâmetro.

2.3 Equação de Burgers

Durante as últimas décadas, consideráveis esforços têm sido feitos no desenvolvimento de procedimentos computacionais robustos para resolver equações diferenciais parciais não lineares encontradas em vários campos das ciências da natureza e da engenharia. As equações mais comuns que envolvem efeitos não lineares e efeitos difusivos são as equações de Burgers. Consistem de equações diferenciais parciais importantes da mecânica dos fluidos e têm sido amplamente usadas em várias aplicações, tais como modelagem de dinâmica de gases, descrição de propagação de ondas em acústica e hidrodinâmica, etc. Foi introduzida por Bateman, em 1915 e posteriormente por J. M. Burgers (1939-1965). Devido ao extensivo trabalho deste último, que desenvolveu um modelo matemático para descrever a turbulência e investigou aspectos espectrais e estatísticos da equação e sistemas de equações relacionados, é conhecida popularmente como equação de

Burgers (DAWHAN *et al.*, 2012). São equações diferenciais parciais não lineares, formuladas a partir de princípios de conservação, que combinam termos difusivos e advectivos. São consideradas formas simplificadas das equações de Navier-Stokes, reduzidas às equações de conservação da Quantidade de Movimento Linear (QML). De acordo com certas condições, podem ser classificadas como: hiperbólicas, parabólicas ou elípticas. Sua ampla aplicabilidade tem levado muitos pesquisadores ao desenvolvimento de diversos esquemas numéricos para encontrar soluções para as equações de Burgers, sendo ainda um campo de pesquisa bastante ativo (DAWHAN *et al.*, 2012; ZHAO *et al.*, 2011).

No presente estudo, observou-se a existência de uma vasta literatura disponível abordando os diferentes aspectos das equações de Burgers, bem como uma variedade de esquemas para encontrar a sua solução. Foram encontrados vários trabalhos baseados na transformação de Hopf-Cole, na qual as equações de Burgers são reduzidas à equação do calor (KANNAN e WANG, 2012; LIAO, 2010 e ZHAO *et al.*, 2011). Os trabalhos de Hon *et al.* (1998) e Siraj-Ul-Islam *et al.* (2012) apresentaram uma abordagem de um método sem malha com o emprego de funções de base radial. Para dar uma ideia da variedade de métodos propostos, cabe mencionar os métodos Lattice-Boltzmann (LIU e SHI, 2011), variacional (AKSAN e ÖZDES, 2004) e esquema de diferenças finitas tipo Crank-Nicolson (XU e SAN, 2009). Uma revisão bibliográfica de trabalhos em que esse problema é usado como *benchmark* pode ser encontrada em Efe (2006). Em um artigo recente, Dawhan *et al.* (2012) apresentaram uma extensa e detalhada revisão das diferentes técnicas empregadas na resolução das equações de Burgers. Nesse trabalho estão as principais referências acerca do estudo destas equações até os dias atuais, acompanhadas de um resumo das análises e dos importantes resultados e desenvolvimentos obtidos.

No presente estudo, foram encontrados alguns trabalhos onde o método *multigrid* é empregado na resolução de problemas que envolvem as equações de Burgers. Entretanto, em sua maioria, o foco de tais estudos não é a redução do tempo de CPU ou o estudo sistemático de parâmetros do *multigrid*, mas apresentar estudos teóricos e/ou testar novas metodologias numéricas.

Considerando as estimativas teóricas sobre as taxas de convergência do método *multigrid* para equações hiperbólicas “escassas e incompletas”, Mulder (1990) investigou a convergência do método *multigrid* para a equação unidimensional de Burgers para escoamento invíscido. Desenvolveu uma análise teórica e apresentou resultados numéricos com o emprego de um algoritmo FAS-FMG de duas malhas e ciclo F e comparou com o

problema linear (equação da convecção com coeficientes constantes). Outro trabalho acerca da convergência do método *multigrid* sob o ponto de vista teórico foi desenvolvido por Ibraheem e Demuren (1995). Estes propuseram o estudo da convergência teórica baseado na análise local dos modos de frequência, para algoritmos de duas malhas. Apresentaram alguns resultados numéricos onde comprovaram a superioridade das previsões realizadas pelo método proposto para os modelos unidimensionais das equações de advecção-difusão e Burgers linearizada, com um algoritmo FAS com duas malhas, ciclo V e os *solvers* LU e ADI. Persson *et al.* (1999) apresentaram dois teoremas acerca da convergência do método *multigrid*. O primeiro prova a convergência para um problema bidimensional modelado pela equação da advecção-difusão, dominado pela advecção. O segundo teorema para problemas de ondas de choque, descritos pelo modelo bidimensional das equações de Burgers.

Foram encontrados alguns poucos trabalhos envolvendo as equações de Burgers onde são apresentados estudos que objetivam melhorar a convergência do método *multigrid*. Ferm e Lötstedt (1997) propuseram um operador de restrição dos resíduos baseado no resíduo para acelerar a convergência do método *multigrid* para problemas com ondas de choque. Um algoritmo FAS para duas e três malhas com o operador proposto foi testado para a equação unidimensional de Burgers e para os modelos unidimensional e bidimensional de Euler. Os resultados mostraram que a taxa de convergência triplica para o problema modelado pela equação de Burgers e dobra para os problemas modelados por Euler, confirmando a superioridade do operador de restrição proposto sobre o operador de restrição linear para os resíduos. Zhang *et al.* (2010), visando reduzir o custo computacional, propuseram o algoritmo *multigrid* espectral. Implementaram quatro métodos iterativos com esquema FAS e ciclo V com 2 e 3 malhas, incluindo o método de malha única, *singlegrid* (SG). Avaliaram a precisão e a eficiência do método através da equação unidimensional de Burgers, Taylor-vórtice bidimensional e o problema bidimensional da cavidade quadrada. Concluíram que o algoritmo FMG com 3 malhas é o que apresenta melhor desempenho, com *speed up* de duas ordens de magnitude. Além disso, mostraram que com o aumento do número de Reynolds, entre 400 e 1000, a taxa de convergência é significativamente afetada pela estimativa inicial. Santiago (2010) investigou alguns parâmetros do *multigrid* para os modelos bidimensionais de Laplace, Navier e Burgers, discretizados pelo método das diferenças finitas. Para as equações de Burgers, o conjunto de parâmetros que resultou no menor tempo de CPU foi: número fixo de iterações internas do *solver* MSI, $\nu = 5$ e número de malhas igual ao máximo para o tamanho de problema considerado. Com esses parâmetros obteve para este modelo fator de aceleração

$S = 15$ na malha $N = 1025 \times 1025$. Observou também que o aumento no tempo de CPU do *multigrid* cresce à mesma taxa para os problemas com uma ou duas equações e concluiu que o número de equações envolvidas no modelo matemático não interfere no desempenho do *multigrid*. Além disso, concluiu que não é vantajoso aplicar método *multigrid* em apenas uma equação quando o modelo envolve duas, como no caso das equações de Burgers. Neveu *et al.* (2011) aplicaram o *multigrid* em um problema de assimilação de dados variacional, que resulta na equação unidimensional de Burgers, transiente com condições de contorno periódicas. Realizaram testes com um algoritmo FAS para duas malhas com razão de refinamento 3, onde avaliaram a taxa de convergência do *multigrid* com o *singlegrid*.

Esta revisão bibliográfica embasa e justifica os objetivos já expostos desta tese, que almeja principalmente desenvolver parametrizações e códigos computacionais eficientes, ao unir o método de discretização de volumes finitos com o método *multigrid*.

CAPÍTULO 3

FUNDAMENTOS TEÓRICOS

3.1 Formulação em Volumes Finitos

Os fenômenos resolvidos com CFD podem ser modelados matematicamente por meio de equações ou sistemas de equações diferenciais. A solução numérica destas equações requer a discretização do domínio de cálculo, de modo a gerar uma malha onde os termos da equação diferencial são aproximados, resultando em um sistema de equações algébricas envolvendo as funções incógnitas. Em CFD, os métodos mais empregados na discretização das equações diferenciais são o MDF, MEF e o MVF (MALISKA, 2004; VERSTEEG e MALALASEKERA, 2007).

No MVF, o domínio de cálculo é dividido em um número finito de subdomínios, denominados volumes de controle, os quais envolvem um único ponto nodal da malha, onde os valores da variável de interesse são calculados. As equações são integradas sobre cada volume de controle (VC) e esquemas de interpolação são utilizados para obter os valores das variáveis nas faces do VC em termos dos valores nodais. Um volume de controle pode ser construído de duas formas (PANTAKAR, 1980): com os nós centrados entre as faces do volume ou com as faces centradas entre os nós. No caso de malhas uniformes as duas formas coincidem. A Fig. 3.1 exemplifica a disposição dos volumes de controle em uma malha bidimensional uniforme (todos os volumes de controle possuem as mesmas dimensões).

Para ilustrar a etapa de discretização da Equação diferencial parcial (EDP), considera-se a equação que expressa a advecção-difusão de uma determinada propriedade ϕ , em regime permanente, dada por

$$\nabla \cdot (\rho \mathbf{v} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi, \quad (3.1)$$

onde o termo do lado esquerdo se refere à advecção da propriedade ϕ , o primeiro termo do lado direito se refere à difusão da propriedade ϕ e segundo termo do lado direito é o termo fonte. Os escalares ρ e Γ referem-se à massa específica e a um coeficiente de transporte,

respectivamente. O vetor velocidade é referenciado por \mathbf{v} e o operador $\vec{\nabla}$ indica o gradiente da propriedade ϕ quando assume a operação $\vec{\nabla}\phi$ e o divergente do vetor velocidade quando assume o produto escalar $\vec{\nabla} \cdot \mathbf{v}$.

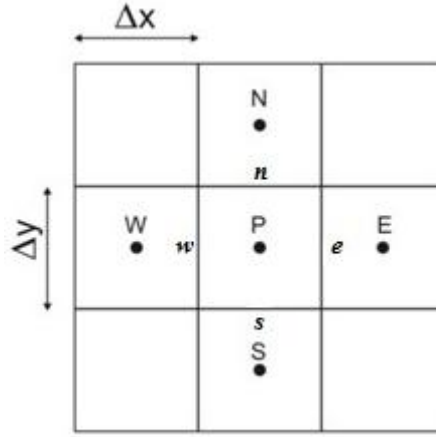


Figura 3.1: Disposição de um VC P e seus vizinhos N (norte), S (sul), (E) leste e W (oeste), em uma malha bidimensional uniforme

Integrando-se a Eq.(3.1) sobre cada volume de controle tem-se

$$\iiint_{VC} \nabla \cdot (\rho \mathbf{v} \phi) dV = \iiint_{VC} \nabla \cdot (\Gamma \nabla \phi) dV + \iiint_{VC} S_\phi dV . \quad (3.2)$$

Aplicando-se o Teorema da Divergência à Eq.(3.2), obtém-se

$$\iint_A (\rho \mathbf{v} \phi) \cdot \mathbf{n} dA = \iint_A (\Gamma \nabla \phi) \cdot \mathbf{n} dA + \iiint_{VC} S_\phi dV , \quad (3.3)$$

onde dA representa o elemento de área da superfície do volume dV e \mathbf{n} representa um vetor unitário normal à superfície do elemento dA .

A solução das integrais na Eq. (3.3) conduz a uma expressão contendo as taxas de transferência advectiva e difusiva da propriedade ϕ através das faces do volume de controle. A avaliação da propriedade ϕ e de suas taxas nas interfaces do volume de controle, em função do valor das propriedades armazenadas nos centros dos volumes, é feita mediante o uso de funções de interpolação. Em Maliska (2004), são apresentados diversos esquemas de interpolação unidimensionais, como os esquemas CDS (*Central Differencing Scheme*), UDS (*Upwind Differencing Scheme*), Esquema Exponencial, WUDS (*Weighted Upstream*

Differencing Scheme), QUICK (*Quadratic upwind differencing*) e bidimensionais, como o SUDS (*Skew Weighted Upstream Differencing Scheme*) e o WUDS-E (*Weighted Upstream Differencing Scheme-Extended*).

Para aproximar as derivadas da propriedade ϕ nas faces e , w , n e s do volume de controle P, pode-se empregar o CDS de 2ª ordem. Considerando-se a disposição dos volumes de controle na Fig. 3.1, as aproximações são dadas por uma interpolação linear:

$$\left. \frac{\partial \phi}{\partial x} \right|_e = \frac{\phi_E - \phi_P}{\Delta x}; \quad (3.4)$$

$$\left. \frac{\partial \phi}{\partial x} \right|_w = \frac{\phi_P - \phi_W}{\Delta x}; \quad (3.5)$$

$$\left. \frac{\partial \phi}{\partial y} \right|_n = \frac{\phi_N - \phi_P}{\Delta y}; \quad (3.6)$$

$$\left. \frac{\partial \phi}{\partial y} \right|_s = \frac{\phi_P - \phi_S}{\Delta y}; \quad (3.7)$$

Para avaliar a propriedade ϕ nas faces e , w , n e s do volume de controle P pode-se empregar o esquema UDS de 1ª ordem, que considera a direção do escoamento através das seguintes aproximações:

$$\phi_w = \phi_W \quad ; \quad \phi_e = \phi_P \quad \text{se} \quad u > 0; \quad (3.8)$$

$$\phi_w = \phi_P \quad ; \quad \phi_e = \phi_E \quad \text{se} \quad u < 0; \quad (3.9)$$

$$\phi_n = \phi_N \quad ; \quad \phi_s = \phi_P \quad \text{se} \quad v < 0; \quad (3.10)$$

$$\phi_n = \phi_P \quad ; \quad \phi_s = \phi_S \quad \text{se} \quad v > 0; \quad (3.11)$$

onde u e v correspondem às velocidades nas faces w e e e s e n do volume de controle, respectivamente.

O emprego da correção adiada (LILEC *et al.*, 1997) permite trabalhar de uma forma implícita com uma formulação *upwind* promovendo a correção para um esquema de 2ª ordem explicitamente. É dada pela seguinte expressão, para uma dada face denotada por f ,

$$\phi_f = \phi_f^{UDS} + \left(\phi_f^{CDS} - \phi_f^{UDS} \right)^*, \quad (3.12)$$

onde os sobrescritos CDS e UDS denotam os esquemas *upwind* e diferenças centrais e $*$ indica condições conhecidas do nível iterativo anterior. Na convergência, as avaliações *upwind* cancelam-se e o esquema CDS é o efetivamente utilizado, resultando em uma aproximação de 2ª ordem. Outros esquemas de primeira ordem e alta ordem podem substituir, respectivamente, os esquemas UDS e CDS na Eq. (3.12).

Depois que a propriedade ϕ e suas derivadas são avaliadas nas faces do volume de controle, por meio das funções de interpolação, tem-se, para cada propriedade ϕ e para cada volume de controle, uma equação algébrica do tipo

$$a_P \phi_P = a_W \phi_W + a_E \phi_E + a_S \phi_S + a_N \phi_N + b_P, \quad (3.13)$$

onde W , E , S e N denotam cada um dos volumes vizinhos do volume P , oeste, leste, sul e norte, respectivamente.

Ferziger e Peric (2002) destacam algumas vantagens do MVF: pode ser aplicado a qualquer tipo de malha, em geometrias complexas, e diferentes sistemas de coordenadas. De acordo com Patankar (1980), a característica mais atraente do MVF é que o esquema resultante da discretização deve implicar na conservação integral das propriedades da grandeza em questão através de qualquer grupo de volumes de controle e, por consequência, de todo o domínio de cálculo. Esta característica é verdadeira para qualquer número de pontos. Assim, mesmo para uma malha grosseira, a solução exhibe balanços integrais exatos.

3.2 Condições de contorno

A Eq. (3.13) representa a equação discretizada para um volume de controle interno. Para a obtenção do sistema de equações algébricas completo é necessário obter, ainda, as equações para os volumes que estão na fronteira.

Os três tipos de condições de contorno utilizados na literatura, inclusive em transferência de calor e de massa, para resolver equações diferenciais parciais, são as condições de contorno de Dirichlet, quando a propriedade ϕ é conhecida no contorno; de Neumann, quando o fluxo da propriedade ϕ é conhecido no contorno; e de Robin, quando a condição de contorno é mista (PATANKAR, 1980).

Existem várias formas de aplicar as condições de contorno. Maliska (2004) apresenta as três maneiras descritas a seguir: balanço para os volumes da fronteira, com volumes fictícios e com meio-volume.

3.2.1 Balanço para os volumes da fronteira

A aplicação das condições de contorno é realizada por meio da integração das equações de conservação também para os volumes dos contornos, da mesma forma realizada para os volumes internos ao domínio, respeitando a condição de contorno (PATANKAR, 1980; MALISKA, 2004). Neste caso, a discretização do domínio é realizada em volumes inteiros. A Fig. 3.2 ilustra a discretização do domínio feita com volumes elementares para aplicação desta forma de condição de contorno, onde P e E são os centros dos volumes, Δx é a distância entre os volumes de controle e Δx_c a distância do contorno ao centro do volume imediatamente vizinho.



Figura 3.2: Condições de contorno com volumes inteiros na fronteira

Maliska (2004) considera esta a forma mais adequada para aplicação das condições de contorno, devido ao seu embasamento físico e à possibilidade de generalização para sistemas coordenados mais complexos.

3.2.2 Com volumes fictícios

A técnica dos volumes fictícios consiste em adicionar volumes de controle ao redor do domínio físico, de modo que o balanço entre as propriedades nos volumes fictícios e seus vizinhos reais satisfaça às condições de contorno originais do problema. Para condição de contorno de Dirichlet, a propriedade ϕ no contorno é conhecida, ϕ_c . Como ilustração, os coeficientes dos volumes fictícios para o volume P ilustrado na Fig.3.3 são determinados por

$$\phi_c = \frac{\phi_P + \phi_E}{2}. \quad (3.14)$$

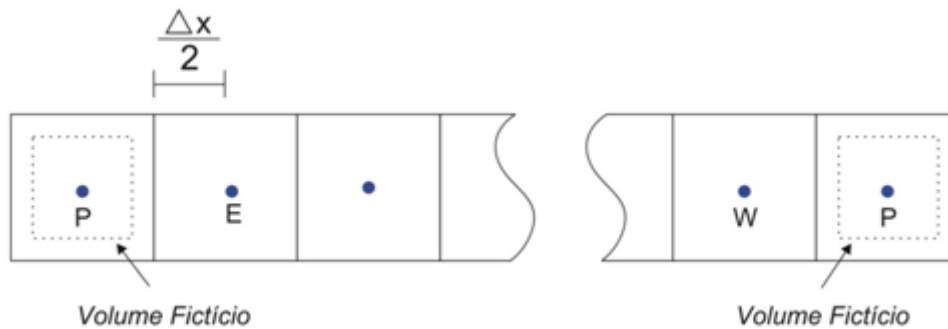


Figura 3.3: Condições de contorno com volumes fictícios na fronteira

Em seguida, isola-se a variável no volume de controle P

$$\phi_P = -\phi_E + 2\phi_c \quad (3.15)$$

e compara-se com a Eq. (3.13) para se obter as expressões para os coeficientes e termo fonte do volume fictício

$$a_P = 1 ; \quad a_E = -1 ; \quad a_W = a_S = a_N = 0 ; \quad b_P = 2\phi_c. \quad (3.16)$$

Como vantagem, todos os volumes do domínio, incluindo os de fronteira, são interpretados como volumes internos. Uma desvantagem é o aumento do número de incógnitas no sistema de equações, que é agravada com o aumento da dimensão do problema.

3.2.3 Com meios volumes

Para aplicar as condições de contorno com meio-volume é necessário criar uma malha em que o ponto central de um volume de controle fique sobre o contorno (PATANKAR, 1980), dando origem a meio-volume de controle entre o contorno e os volumes internos, conforme pode ser visto na Fig. 3.4.

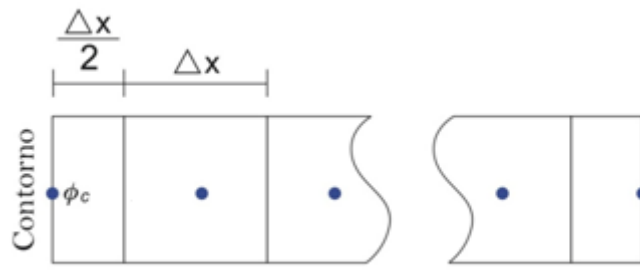


Figura 3.4: Condições de contorno com meio-volume na fronteira

A não uniformidade dos volumes é uma das desvantagens deste procedimento. Outra desvantagem ocorre no caso de condição de contorno prescrita. Embora não seja necessário criar uma equação para o volume da fronteira, uma vez que $\phi_p = \phi_c$, não são observados os balanços de conservação.

3.3 Métodos iterativos básicos

De acordo com Burden e Faires (2003) os *solvers* iterativos são computacionalmente mais eficientes e robustos do que os *solvers* diretos para a resolução de sistemas de equações algébricas. A ideia básica dos *solvers* iterativos consiste em determinar a solução do sistema de equações algébricas, Eq.(1.1) a partir de uma estimativa inicial. Com esta, novas soluções aproximadas são geradas sucessivamente até encontrar uma solução que atenda a certo critério de parada (que pode ser baseado no erro, resíduo ou número de iterações) previamente estabelecido. Esses métodos convergem mediante certas condições impostas sobre a matriz

dos coeficientes do sistema linear e são vantajosos com relação aos métodos diretos quando o sistema de equações é esparso e de grande porte. Para comparar, o esforço computacional para os métodos diretos é da ordem de $O(n^3)$ e para os métodos iterativos $O(n^2/2)$, onde n é o número de incógnitas (FERZIGER e PERIC, 2001).

Considera-se o sistema de equações algébricas da Eq. (1.1), em que a matriz dos coeficientes pode ser decomposta como

$$\mathbf{A} = \mathbf{M} - \mathbf{N}. \quad (3.17)$$

Supondo que \mathbf{M} é inversível, obtém-se, substituindo-se a Eq. (3.17) na Eq. (1.1)

$$\mathbf{M}\phi^{(m+1)} = \mathbf{N}\phi^{(m)} + \mathbf{f}, \quad (3.18)$$

onde o sobrescrito de ϕ é a representação dos passos iterativos.

A solução do sistema linear é obtida fazendo-se

$$\phi^{(m+1)} = \mathbf{S}\phi^{(m)} + \mathbf{M}^{-1}\mathbf{f}, \quad (3.19)$$

onde $\mathbf{S} = \mathbf{M}^{-1}\mathbf{N}$ é chamada de matriz de iteração do método iterativo e $m = 1, 2, 3, \dots$ é a iteração.

Os sistemas de equações não lineares também são representados por $\mathbf{A}\phi = \mathbf{f}$ e a matriz \mathbf{A} pode ser decomposta da mesma forma que na Eq. (3.17).

No procedimento descrito anteriormente, a matriz de iteração é constante ao longo do processo iterativo. Os *solvers* iterativos que possuem esta propriedade são chamados métodos iterativos básicos ou estacionários. Alguns exemplos deste tipo de *solvers* são: Jacobi, SOR (*Successive Over-Relaxation*) e Gauss-Seidel. Particularmente, Gauss-Seidel e suas variantes constituem uma importante classe de *solvers* iterativos: Gauss-Seidel lexicográfico (GS-Lex), Gauss-Seidel *red-black* (GS-RB), entre outros. No contexto do *multigrid* as propriedades de suavização do Gauss-Seidel são mais importantes do que suas propriedades de convergência (TROTTERBERG *et al.*, 2001).

No GS-Lex, as incógnitas são atualizadas na ordem lexicográfica, ou seja, de oeste para leste e do sul para o norte, de acordo com a numeração apresentada na Fig. 3.5.

• ₃₁	• ₃₂	• ₃₃	• ₃₄	• ₃₅	• ₃₆
• ₂₅	• ₂₆	• ₂₇	• ₂₈	• ₂₉	• ₃₀
• ₁₉	• ₂₀	• ₂₁	• ₂₂	• ₂₃	• ₂₄
• ₁₃	• ₁₄	• ₁₅	• ₁₆	• ₁₇	• ₁₈
• ₇	• ₈	• ₉	• ₁₀	• ₁₁	• ₁₂
• ₁	• ₂	• ₃	• ₄	• ₅	• ₆

Figura 3.5: Ordenação lexicográfica em uma malha bidimensional

No GS-RB, a malha é organizada conforme a Fig.3.6, onde as incógnitas em vermelho e preto são atualizadas alternadamente. Com esta ordenação, os elementos de uma cor são cercados por quatro elementos (norte, sul, leste e oeste) da outra cor. Assim, os elementos de uma cor podem ser atualizados independentemente dos elementos de outra cor em cada ciclo de varredura da malha.

• ₃₄	• ₁₆	• ₃₅	• ₁₇	• ₃₆	• ₁₈
• ₁₃	• ₃₁	• ₁₄	• ₃₂	• ₁₅	• ₃₃
• ₂₈	• ₁₀	• ₂₉	• ₁₁	• ₃₀	• ₁₂
• ₇	• ₂₅	• ₈	• ₂₆	• ₉	• ₂₇
• ₂₂	• ₄	• ₂₃	• ₅	• ₂₄	• ₆
• ₁	• ₁₉	• ₂	• ₂₀	• ₃	• ₂₁

Figura 3.6: Ordenação *red-black* em uma malha bidimensional

A ordenação do GS-RB permite que a atualização seja realizada simultaneamente nos pontos definidos como vermelhos e pretos. Esta possibilidade favorece o processamento paralelo e, de acordo com Zhang (1996), pode apresentar resultados superiores ao do GS-Lex

no processamento serial. De acordo com Briggs (2000), a vantagem do GS-RB sobre o GS-Lex depende muitas vezes do problema em que são aplicados estes *solvers*.

A seguir, são apresentados os algoritmos que descrevem o procedimento destes dois *solvers*.

Algoritmo 3.1: Procedimento de GS-Lex para $A\phi = f$.

GS-Lex ($m = 0, m_{\text{máx}}, \phi^0$)

- 1) Calcular $\phi_P^{m+1} = (a_N \phi_N^m + a_S \phi_S^m + a_E \phi_E^m + a_W \phi_W^m + f) / a_P$ para todos pontos da Fig. 3.5.
- 2) Faça m receber $m + 1$.
- 3) Volte ao passo 1 até convergir³ ou até atingir $m_{\text{máx}}$.

Algoritmo 3.2: Procedimento de GS-RB para $A\phi = f$.

GS-RB ($m = 0, m_{\text{máx}}, \phi^0$)

- 1) Calcular $\phi_P^{m+1} = (a_N \phi_N^m + a_S \phi_S^m + a_E \phi_E^m + a_W \phi_W^m + f) / a_P$ para os pontos vermelhos da Fig.3.6.
- 1) Calcular $\phi_P^{m+1} = (a_N \phi_N^m + a_S \phi_S^m + a_E \phi_E^m + a_W \phi_W^m + f) / a_P$ para os pontos pretos da Fig. 3.6.
- 3) Faça m receber $m + 1$.
- 4) Volte ao passo 1 até convergir ou até atingir $m_{\text{máx}}$.

A taxa de convergência dos métodos iterativos depende fortemente do raio espectral⁴ da matriz de iteração. Para o método de Jacobi com um parâmetro de relaxação ω , por exemplo, a matriz de iteração S é dada por

$$S_\omega = (1 - \omega)I + \omega D^{-1}(L + U), \quad (3.20)$$

onde D é uma matriz diagonal, que contém a diagonal principal de A , L e U são matrizes triangulares inferior e superior, respectivamente, da matriz A e I é a matriz identidade. De acordo com Briggs *et al.* (2000), os autovalores da matriz S_ω são dados pela expressão

³ Segundo uma tolerância pré estabelecida para o resíduo.

⁴ $\rho(A) = \max |\lambda|$, onde λ são os autovalores da matriz dos coeficientes.

$$\lambda_k(\mathbf{S}_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2N}\right), \quad 1 \leq k \leq N-1, \quad (3.21)$$

onde N é o número de pontos em cada direção da malha computacional e k denota o número de ondas ou modos de Fourier. Na Eq.(3.21) pode-se observar que, para $0 < \omega \leq 1$, tem-se $|\lambda_k(\mathbf{S}_\omega)| < 1$, condição suficiente para garantir a convergência. Além disso, se $k = 1$, para todos os valores de ω no intervalo $0 < \omega \leq 1$ tem-se,

$$\begin{aligned} \lambda_1(\mathbf{S}_\omega) &= 1 - 2\omega \sin^2\left(\frac{\pi}{2N}\right) \\ &= 1 - 2\omega \sin^2\left(\frac{\pi h}{2}\right) \\ &\approx 1 - \frac{\omega \pi^2 h^2}{2}, \end{aligned} \quad (3.22)$$

sendo $h = \frac{1}{N}$ o espaçamento da malha uniforme.

A Eq. (3.22) implica que $\lambda_1(\mathbf{S}_\omega) \approx 1 \quad \forall \omega \in (0,1)$, para N suficientemente grande.

Como λ_1 é associado a um autovetor da forma $\mathbf{w}_j = \sin\left(\frac{j\pi}{N}\right)$, $0 \leq j \leq N$, as

componentes do erro associadas às baixas frequências não são removidas com eficiência pelo método de Jacobi.

Grande parte dos métodos iterativos padrão apresentam propriedades de suavização de erros locais de alta frequência (componentes oscilatórias), enquanto as baixas frequências (componentes suaves) são mantidas praticamente inalteradas. Deste modo, as primeiras iterações do processo geralmente apresentam rápida convergência, caracterizando a presença de modos oscilatórios do erro. Porém, após algumas iterações, a convergência torna-se lenta, sinalizando a presença de modos suaves (BRANDT, 1977; WESSELING, 1992).

Para ilustrar este procedimento, considera-se um problema homogêneo unidimensional $\mathbf{A}\phi = \mathbf{0}$, que tem solução exata conhecida (nula). A estimativa inicial é dada pelo vetor \mathbf{v} , de forma que o erro na aproximação é dado por $\mathbf{e} = -\mathbf{v}$ e a j -ésima componente é dada por

$$v_j = \sin\left(\frac{jk\pi}{N}\right), \quad 0 \leq j \leq N \quad \text{e} \quad 0 \leq k \leq N-1, \quad (3.23)$$

onde N é o número de pontos e k é o número de ondas ou modos de Fourier.

Na Fig. 3.7 podem ser visualizados os modos de Fourier $k = 1$, $k = 3$ e $k = 6$. Os valores de k denotam quantos “meios-senos” constituem a curva v no domínio do problema. Observa-se que pequenos valores de k correspondem a ondas longas e suaves, enquanto que valores maiores de k correspondem a ondas mais curtas e oscilatórias. Os modos de Fourier localizados na metade inferior do espectro, com $1 < k < N/2$, são chamados de modos de Fourier de baixa frequência ou modos suaves. Os modos de Fourier localizados na metade superior do espectro, $N/2 \leq k \leq N-1$, são chamados de modos de Fourier de alta frequência ou modos oscilatórios. Nota-se que esta definição dos modos de Fourier como sendo modos suaves ou oscilatórios depende do número de incógnitas com que o problema está sendo resolvido.

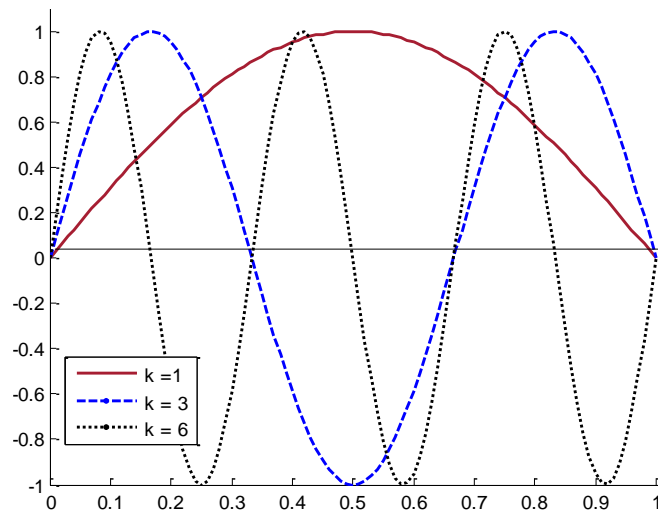


Figura 3.7: Modos de Fourier com $k=1, 3, 6$ (Adaptado de BRIGGS *et al.*, 2000)

Para um caso unidimensional, a Fig. 3.8 ilustra a representação de um modo do erro na malha Ω^h com $N = 12$ pontos e sua projeção na malha Ω^{2h} com $N = 6$ pontos.

O modo de Fourier na malha $N = 12$ tem $k = 4$ e, pela definição, corresponde a um modo suave nesta malha. Na malha $N = 6$, de acordo com a definição, $k = 4$ corresponde a um modo oscilatório. Assim, o erro projetado é mais oscilatório na malha mais grossa.

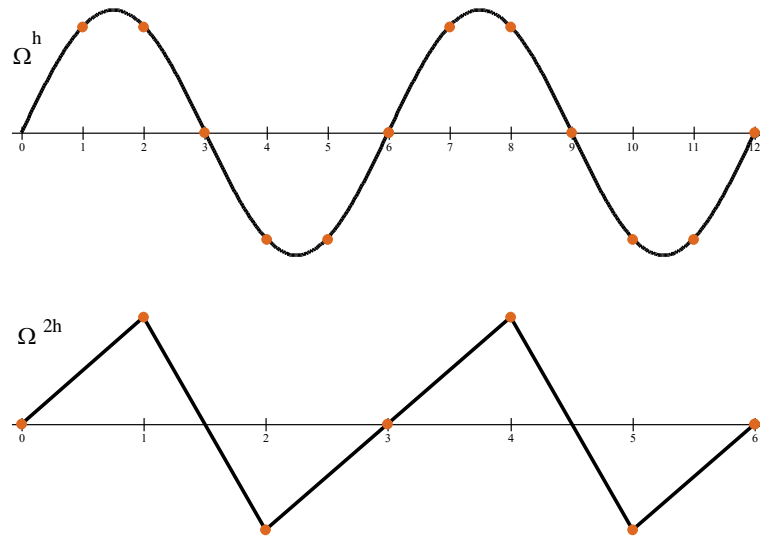


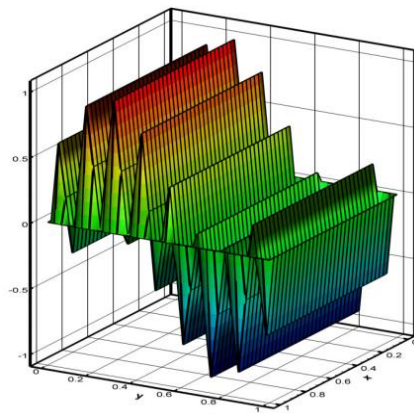
Figura 3.8: Modo do erro na malha Ω^h projetado na malha grossa Ω^{2h}
(Adaptado de BRIGGS *et al.*, (2000))

Para ilustrar esse efeito no caso bidimensional, Santiago (2010) resolveu a equação de Laplace bidimensional, que tem solução exata conhecida. Realizou algumas iterações com o método de Gauss-Seidel, com estimativa inicial \mathbf{v} dada por uma combinação de modos de Fourier, com altas e baixas frequências, na direção de y , dados por

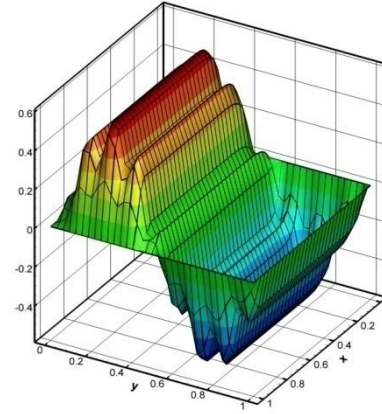
$$v_{i,j} = \frac{1}{2} \left[\sin \left(\frac{k_1 i \pi}{N_y} \right) + \sin \left(\frac{k_2 j \pi}{N_y} \right) \right], \quad 0 \leq i, j \leq N_y - 1, \quad 0 \leq k_1, k_2 \leq N_y - 1. \quad (3.24)$$

As iterações foram realizadas com dois tamanhos de malha: 33×33 com $k_1 = 2$ e $k_2 = 16$ e na malha grossa 17×17 com $k_1 = 2$ e $k_2 = 8$, onde k_1 e k_2 representam os modos de baixa e alta frequência, respectivamente. As figuras 3.9a-d e 3.10a-d ilustram as superfícies que demonstram o erro na estimativa inicial e após 2, 4 e 10 iterações nas malhas 17×17 e 33×33 , respectivamente.

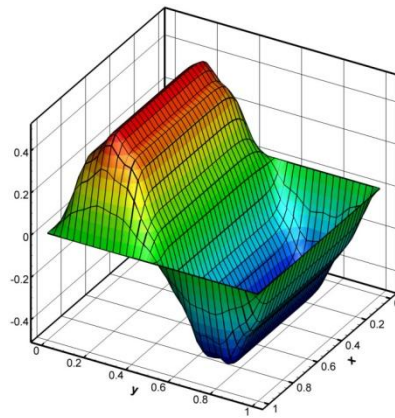
Na Fig. 3.9 observa-se que as componentes de alta frequência são suavizadas ao longo do processo iterativo. Já as componentes de baixa frequência são pouco influenciadas ao longo das iterações. Quando o mesmo problema é iterado na malha mais grossa (17×17), observa-se que tanto as componentes de alta frequência quanto as de baixa frequência são suavizadas mais rapidamente.



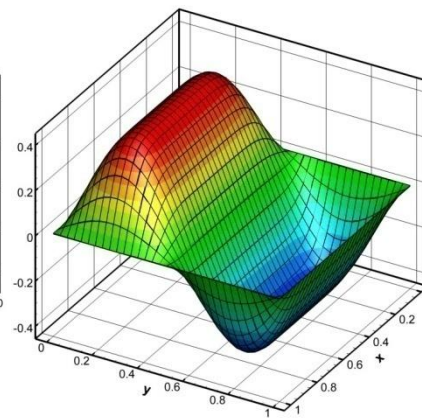
(a) Erro na estimativa inicial.



(b) Erro após 2 iterações.



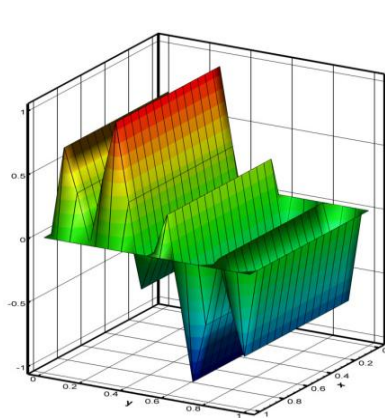
(c) Erro após 4 iterações.



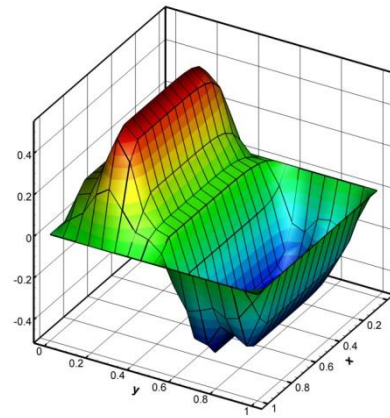
(d) Erro após 10 iterações.

Figura 3.9: Suavização do erro com o método Gauss-Seidel na malha 33×33 (Adaptado de Santiago. (2010))

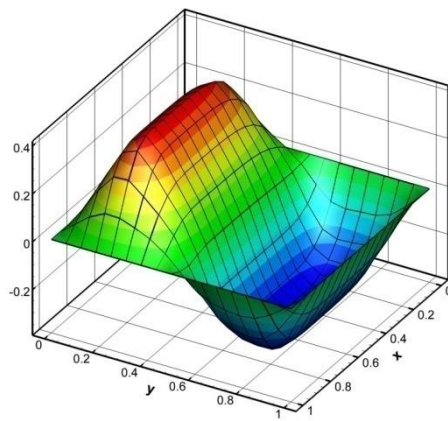
Em geral, as componentes suaves do erro aparecem oscilatórias em malhas mais grossas, o que sugere que quando um processo iterativo começa a tornar-se lento, sinalizando a predominância de modos suaves do erro, é recomendável transferir o problema para uma malha mais grossa, onde se apresentarão mais oscilatórios, e a suavização será mais eficiente (WESSELING, 1992 e TROTTEBERG *et al.* 2001).



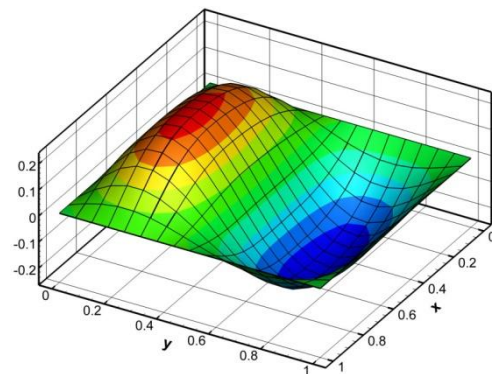
(a) Erro na estimativa inicial.



(b) Erro após 2 iterações.



(c) Erro após 4 iterações.



(d) Erro após 10 iterações.

Figura 3.10: Suavização do erro com o método Gauss-Seidel na malha 17x17
(Adaptado de Satiago (2010))

3.4 O *multigrid*

O método *multigrid* tem como princípios básicos a propriedade de suavização e a correção de malha grossa (BRIGGS *et al.*, 2000). O princípio de suavização assegura que métodos iterativos básicos tem forte efeito de suavização sobre o erro de qualquer aproximação. O princípio da malha grossa afirma que um modo de erro suave é bem aproximado em uma malha grossa.

De acordo com estes princípios, um esquema iterativo apropriado, com várias malhas de diferentes graus de refinamento, promoverá uma rápida redução das componentes de alta frequência correspondentes e, como este processo passa por várias malhas, uma rápida redução do erro global pode ser alcançada, acelerando a convergência do método iterativo.

A taxa de convergência ideal (teórica) do método *multigrid* independe do número de pontos da malha (FERZIGER E PERIC, 2002).

3.4.1 Equação residual e esquema de correção

Seja Φ a solução exata do sistema linear dado pela Eq. (1.1). Se $\tilde{\phi}$ é uma aproximação para Φ , então o erro algébrico (\mathbf{e}), ou apenas erro, pode ser calculado por

$$\mathbf{e} = \Phi - \tilde{\phi}. \quad (3.25)$$

A magnitude do vetor erro, dado pela Eq. (3.25), pode ser medida por alguma norma de vetores, sendo as mais utilizadas a norma l_1 , a norma euclidiana e a norma infinito, definidas, respectivamente, pelas equações

$$\|\mathbf{e}\|_{l_1} = \sum_{i=1}^n |e_i|, \quad (3.26)$$

$$\|\mathbf{e}\|_2 = \left(\sum_{i=1}^n e_i^2 \right)^{1/2}, \quad (3.27)$$

e

$$\|\mathbf{e}\|_{\infty} = \max_i |e_i|, \quad (3.28)$$

onde e_i representa a i -ésima componente do vetor erro. Outras normas vetoriais podem ser encontradas em Burden e Faires (2003).

Uma vez que, em geral, a solução exata do sistema linear não é conhecida, também o erro não pode ser conhecido *a priori*. Para verificar o quanto a aproximação \mathbf{v} dista da solução exata do sistema linear ao longo do processo iterativo, calcula-se o resíduo \mathbf{r} da Eq. (1.1), definido por

$$\mathbf{r} = \mathbf{f} - \mathbf{A}\tilde{\phi}. \quad (3.29)$$

O resíduo mede o quanto a aproximação falha ao satisfazer o sistema dado pela Eq. (1.1). Teoricamente, se $\tilde{\phi} = \Phi$ então $\Phi - \tilde{\phi} = \mathbf{e} = 0$ e, portanto, $\tilde{\phi}$ satisfaz o sistema linear.

A partir das equações do erro e do resíduo, Eqs. (3.25) e (3.29), respectivamente, pode-se estabelecer uma importante relação, que pode ser obtida substituindo-se a Eq. (1.1) na equação do resíduo, Eq. (3.29). Assim, obtém-se:

$$\mathbf{r} = \mathbf{A}\Phi - \mathbf{A}\tilde{\phi}, \quad (3.30)$$

que, para o caso linear, pode ser escrito como

$$\mathbf{r} = \mathbf{A}(\Phi - \tilde{\phi}). \quad (3.31)$$

Com a substituição da Eq. (3.25) na Eq. (3.31), tem-se um sistema linear, no qual o vetor das incógnitas é o erro e o vetor dos termos independentes é o resíduo, ou seja

$$\mathbf{A}\mathbf{e} = \mathbf{r}. \quad (3.32)$$

A Eq. (3.32) é chamada de equação residual e desempenha um papel importante no âmbito do método *multigrid*. Ela indica que o erro dado pela Eq. (3.25) satisfaz o mesmo conjunto de equações que a solução Φ quando \mathbf{f} é substituído pelo resíduo \mathbf{r} .

Para o caso não linear, adota-se a seguinte notação para a equação residual

$$\mathbf{r} = \mathbf{A}(\Phi) - \mathbf{A}(\tilde{\phi}) \quad (3.33)$$

É importante notar que, diferentemente do que acontece no caso linear, no caso não linear $\mathbf{A}(\mathbf{e}) = \mathbf{A}(\Phi - \tilde{\phi}) \neq \mathbf{A}(\Phi) - \mathbf{A}(\tilde{\phi})$.

Considerando-se que uma aproximação $\tilde{\phi}$ tenha sido obtida para o sistema linear dado pela Eq. (1.1), por meio de algum método iterativo, o resíduo pode ser calculado com a Eq. (3.29). Para melhorar essa aproximação, calcula-se o erro com a Eq. (3.32) e depois se corrige a solução atual obtendo uma nova aproximação, através da expressão

$$\phi = \tilde{\phi} + \mathbf{e}. \quad (3.34)$$

Assim, a solução é a aproximação atual acrescida do erro. O procedimento descrito acima é o chamado esquema de correção, um dos princípios fundamentais do *multigrid*.

No Capítulo 1 deste texto, são mencionados dois tipos de esquemas com os quais o método *multigrid* pode ser implementado: os esquemas CS e FAS. O esquema de correção, apresentado anteriormente, descreve o modo como o sistema de equações algébricas é operado nas malhas mais grossas no esquema CS. Neste esquema, o sistema de equações original, Eq. (1.1), é resolvido apenas na malha mais fina. O resíduo, Eq.(3.29), é transferido para a malha grossa subsequente, onde é resolvida a equação residual, Eq. (3.32). Na malha mais grossa, o erro é transferido para a malha fina subsequente para corrigir a solução.

Outro esquema segundo o qual o *multigrid* pode ser implementado é o FAS. O princípio fundamental do FAS é semelhante ao do CS sendo que, além do resíduo, a aproximação da solução também é transferida para a malha grossa subsequente. Na malha mais grossa, o erro é transferido para aproximar a solução na malha fina subsequente. Em contraponto ao CS, no FAS não se resolve a equação residual explicitamente; resolve-se a equação discretizada na malha grossa. De acordo com estas características, os esquemas CS e FAS são indicados, respectivamente, para problemas lineares e não lineares (Brandt, 1977).

3.4.2 Razão de refinamento

Várias razões de refinamento de malha podem ser usadas com o método *multigrid*. Briggs et al. (2000) afirmam que a razão $r = 2$ é uma prática universal e que o uso de diferentes razões de refinamento geralmente não apresenta vantagens. Nessa razão, se considerado o caso unidimensional, o número de elementos de uma malha fina é o dobro do número de elementos da malha grossa subsequente; no caso bidimensional, o número de elementos duplica em cada direção. A Fig. 3.11 apresenta uma sequência de malhas com razão de refinamento $r = 2$ para o caso bidimensional.

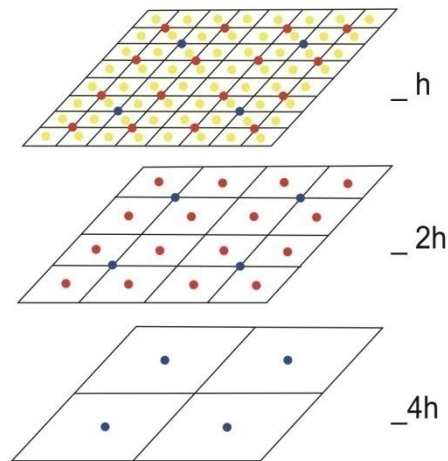


Figura 3.11: Sequência de malhas com razão de refinamento $r = 2$, $N = 2 \times 2$, 4×4 e 8×8

Brandt (1977) fez comparações com razões de refinamento $r = 2$, 3 e $3/2$ e concluiu que a razão $r = 2$ é a recomendável. Seu estudo envolveu diversos problemas de transferência de calor e escoamento de fluidos, unidimensionais, bidimensionais, lineares e não lineares. Pinto *et al.* (2005) também analisaram várias razões de engrossamento, para o problema de Burgers unidimensional, e não encontraram vantagens significativas em relação à razão de refinamento $r = 2$.

Dada a razão de refinamento, pode-se determinar o maior número de malhas (L) possível para cada tamanho de problema ($L_{máx}$). Por exemplo, para $r = 2$, o problema de tamanho $N = 128 \times 128$ pode ser resolvido no conjunto de malhas $N = 2 \times 2$, 4×4 , 8×8 , 16×16 , 32×32 , 64×64 e 128×128 . Assim, para este tamanho de problema $L_{máx} = 7$.

3.4.3 Operadores de restrição

A transferência de informações da malha fina Ω^h para a malha imediatamente mais grossa Ω^{2h} requer a definição do operador de restrição, denotado genericamente por \mathbf{I}_h^{2h} . Vários operadores de restrição são apresentados na literatura: injeção, ponderação completa e meia ponderação (WESSELING, 1992; BRIGGS *et al.*, 2000; TROTTERBERG *et al.*, 2001). Pela sua simplicidade, o operador de restrição por injeção é um dos procedimentos mais utilizados. Ele é definido como

$$\phi^{2h} = \mathbf{I}_h^{2h} \phi^h. \quad (3.35)$$

Este operador transfere informações dos valores nodais da malha fina Ω^h para os nodos coincidentes da malha grossa Ω^{2h} .

Considerando-se o MVF, com razão de refinamento $r = 2$, cada volume de controle da malha grossa corresponde a quatro volumes de controle na malha fina, conforme Fig. 3.12. Neste caso, não existem nodos coincidentes entre as duas malhas e, portanto, não é possível aplicar o operador por injeção.

Um operador de restrição frequentemente empregado neste caso é obtido tomando-se a média aritmética dos valores da propriedade dos quatro volumes da malha fina (TROTTERBERG *et al.*, 2001):

$$\mathbf{I}_h^{2h} \phi^h = \frac{1}{4} (\phi_A^h + \phi_B^h + \phi_C^h + \phi_D^h). \quad (3.36)$$

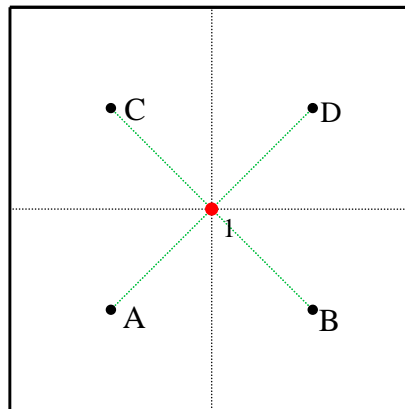


Figura 3.12: Disposição do volume da malha grossa (nodos vermelhos) e dos volumes da malha fina (nodos pretos)

No MVF, uma prática comum para restringir o resíduo da malha fina para a malha grossa, é através da soma dos resíduos relativos aos volumes de controle da malha fina que correspondem àquele da malha grossa (FERZIGER e PERIC, 2002; KUMAR *et al.*, 2009):

$$\mathbf{I}_h^{2h} \phi^h = \phi_A^h + \phi_B^h + \phi_C^h + \phi_D^h. \quad (3.37)$$

Como o resíduo pode ser encarado como uma propriedade extensiva, este procedimento proporciona o balanço corretamente. Porém, não é uma prática unânime. Em Versteeg e Malalasekera (2007) pode ser encontrado um exemplo do uso do operador definido na Eq. (3.36) para restringir os resíduos.

3.4.4 Prolongação

Os operadores que transferem informações da malha grossa para a malha fina são chamados operadores de prolongação e são denotados genericamente por \mathbf{I}_{2h}^h . A interpolação bilinear é um dos operadores de prolongação mais comuns na literatura (WESSELING, 1992; BRIGGS *et al.*, 2000 ; TROTTERBERG *et al.*, 2001). Em Drikakis *et al.*, 1998 pode ser encontrado um exemplo do emprego do operador trilinear.

As expressões para a prolongação bilinear da propriedade ϕ da malha grossa para a malha fina, ilustradas na Fig. 3.13, são dadas por

$$\mathbf{I}_{2h}^h \phi^{2h} = \begin{cases} \frac{1}{16} (9\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + \phi_D^{2h}) & \text{para } \phi_1^h \\ \frac{1}{16} (3\phi_A^{2h} + 9\phi_B^{2h} + \phi_C^{2h} + 3\phi_D^{2h}) & \text{para } \phi_2^h \\ \frac{1}{16} (3\phi_A^{2h} + \phi_B^{2h} + 9\phi_C^{2h} + 3\phi_D^{2h}) & \text{para } \phi_3^h \\ \frac{1}{16} (\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + 9\phi_D^{2h}) & \text{para } \phi_4^h \end{cases}. \quad (3.38)$$

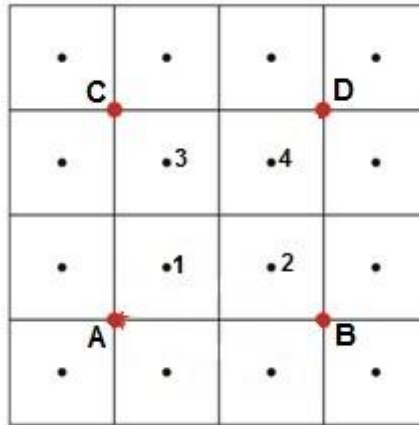


Figura 3.13: Esquema de malhas para operador de prolongação- malha grossa (nodos vermelhos) e malha fina (nodos pretos)

Em Trottenberg *et al.*(2001), são apresentados outros operadores de prolongação, como o quadrático e o cúbico, para os casos unidimensional e bidimensional.

De um modo geral, os operadores de restrição e prolongação são definidos por interpolações. A ordem de uma interpolação é igual a $k + 1$ se é exata para todo polinômio de

grau k (BURDEN e FAIRES, 2003). Por exemplo, o operador bilinear tem ordem 2. Uma regra básica para boa convergência do método *multigrid* é que

$$m_r + m_p > m, \quad (3.39)$$

onde m_r e m_p são, respectivamente, as ordens das interpolações relativas aos operadores de restrição e prolongação e m é a ordem da equação diferencial (TROTTERBERG, 2001). Por outro lado, de acordo com Brandt (2011), não há significativa melhora na taxa de convergência do método se $m_r + m_p > m + 1$.

3.4.5 Ciclos

Durante a execução do algoritmo do método *multigrid*, diferentes malhas são visitadas. A sequência com que as malhas são percorridas é denominada ciclo. O ciclo V é um dos mais populares. Wesseling (1992) apresentou uma generalização do ciclo V, conhecida como ciclo μ , da qual podem ser deduzidos outros tipos de ciclos. Por exemplo, $\mu = 1$ corresponde ao ciclo V e $\mu = 2$ corresponde ao ciclo W. A Figs. 3.14 e 3.15 ilustram cada um destes ciclos.

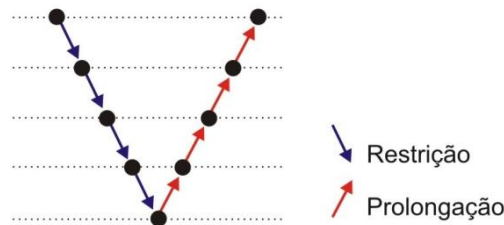


Figura 3.14: Ciclo V com 5 malhas

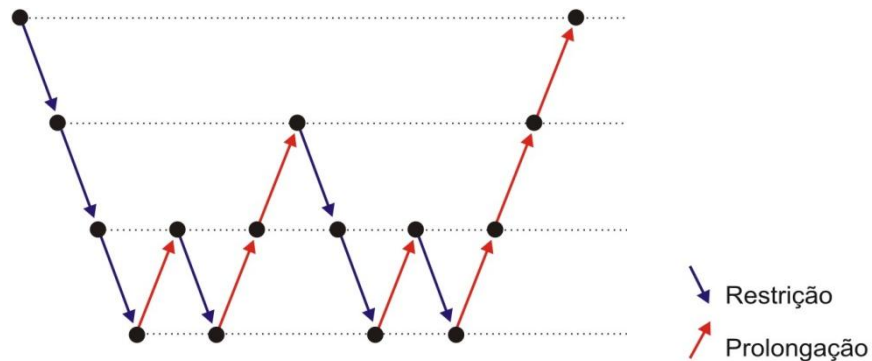


Figura 3.15: Ciclo W com 4 malhas

O número de suavizações, ou iterações do *solver* (v), geralmente depende do *solver* e não precisa ser necessariamente o mesmo em todos os níveis de malha, tanto no processo de restrição quanto no de prolongação. Briggs *et al.* (2000) afirmam que em problemas que não apresentam dificuldades de convergência, são empregadas entre 1 e 3 suavizações.

3.4.6 FMG

Uma boa estimativa inicial pode reduzir consideravelmente o tempo de CPU, dado que a resolução de um problema com uma boa estimativa inicial requer poucas iterações na malha fina.

Uma forma de obter uma boa estimativa inicial é interpolando-se uma solução da malha grossa para a malha fina, isto é

$$(\phi^h)_0 = \mathbf{I}_{2h}^h (\phi^{2h})_{\text{convergida}}, \quad (3.40)$$

onde \mathbf{I}_{2h}^h é o operador de prolongação da malha grossa para a malha fina. A estratégia mencionada acima é denominada *Full Multigrid* (FMG) e pode ser associada a qualquer tipo de ciclo do *multigrid*.

Uma ilustração esquemática para um algoritmo com quatro malhas e ciclo V é dada na Fig.3.16. Neste caso, ciclos V são usados como blocos básicos na composição do algoritmo.

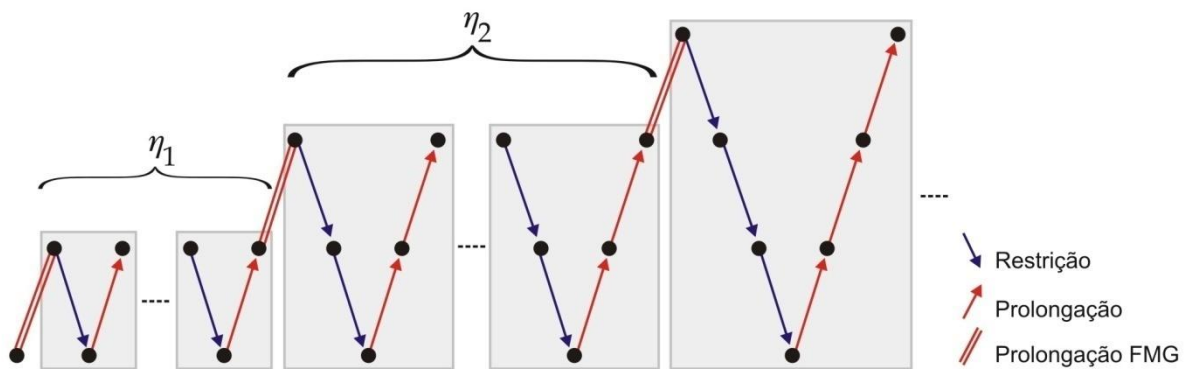


Figura 3.16: FMG com 4 malhas (Adaptado de Thekale *et al.*, (2010))

Vários ciclos V podem ser executados em cada nível η_i do FMG, onde i varia de 1 até $L_{\text{máx}} - 2$. A prática comum é que sejam empregados 1 ou 2 ciclos (TROTTEBERG, 2001;

BRANDT, 2011). Porém, existem estudos que buscam determinar o número ótimo de ciclos a cada nível (THEKALE *et al.*, 2010).

O símbolo // denota a prolongação da solução obtida no nível η_i como estimativa inicial para a solução do nível η_{i+1} . O operador de prolongação da solução do FMG, indicado por // na Fig. 3.16, não é necessariamente o mesmo usado para prolongar as correções nos ciclos do *multigrid*. De acordo com Brandt (2011), frequentemente este operador deve ser de ordem mais alta.

A vantagem do FMG é que uma boa estimativa inicial para a malha fina é produzida ao custo de poucas iterações na malha grossa. Além disso, os erros são suavizados eficientemente, o que garante uma ótima taxa de convergência. De acordo com Trottenberg (2001) estas propriedades fazem do FMG a mais eficiente versão do MG e Thekale *et al.* (2010) e Zhang *et al.* (2010) consideram o FMG o método preferido para acelerar o *multigrid*.

3.4.7 Algoritmos

Os passos básicos do algoritmo FAS, para um único ciclo V, generalizado para L malhas, $2 \leq L \leq L_{\text{máx}}$, são descritos no algoritmo a seguir. Na prática, vários ciclos V são realizados, sucessivamente, até que o critério de parada seja alcançado.

Os parâmetros ν_1 e ν_2 representam o número de suavizações realizadas no processo de restrição e prolongação. Eles são chamados de pré e pós-suavização, respectivamente. Os sobrescritos $h, 2h, 4h, \dots$ indicam a malha onde se definem as informações (operadores, resíduos, correções, etc.) e os operadores $\mathbf{A}^{2h}, \mathbf{A}^{4h}, \mathbf{A}^{8h} \dots$ podem ser obtidos de forma semelhante à usada para obter a matriz \mathbf{A}^h , ou seja, discretizando-se a equação original nas respectivas malhas (BRIGGS *et al.*, 2000).

Algoritmo 3.3: Esquema FAS com ciclo V para L malhas.

FAS($\tilde{\phi}_0^h, \nu_1, \nu_2$)

- 1) Iterar $\mathbf{A}^h(\tilde{\phi}^h) = \mathbf{f}^h$ ν_1 vezes com estimativa inicial $\tilde{\phi}_0^h$;
- 2) Calcular $\mathbf{r}^{2h} = \mathbf{I}_h^{2h}(\mathbf{f}^h - \mathbf{A}^h(\tilde{\phi}^h))$ e $\tilde{\phi}_0^{2h} = \mathbf{I}_h^{2h}\tilde{\phi}^h$;
- 3) Definir $\mathbf{f}^{2h} = \mathbf{A}^{2h}(\tilde{\phi}^{2h}) + \mathbf{r}^{2h}$;

- 4) Iterar $\mathbf{A}^{2h}(\tilde{\phi}^{2h}) = \mathbf{f}^{2h}$ ν_1 vezes com estimativa inicial $\tilde{\phi}_0^{2h}$;
- 5) Calcular $\mathbf{r}^{4h} = \mathbf{I}_{2h}^{4h}(\mathbf{f}^{2h} - \mathbf{A}^{2h}(\tilde{\phi}^{2h}))$ e $\tilde{\phi}_0^{4h} = \mathbf{I}_{2h}^{4h}\tilde{\phi}^{2h}$;
- 6) Definir $\mathbf{f}^{4h} = \mathbf{A}^{4h}(\tilde{\phi}^{4h}) + \mathbf{r}^{4h}$;
- 7) Iterar $\mathbf{A}^{4h}(\tilde{\phi}^{4h}) = \mathbf{f}^{4h}$ ν_1 vezes com estimativa inicial $\tilde{\phi}_0^{4h}$;
- 8) Calcular $\mathbf{r}^{8h} = \mathbf{I}_{4h}^{8h}(\mathbf{f}^{4h} - \mathbf{A}^{4h}(\tilde{\phi}^{4h}))$ e $\tilde{\phi}_0^{8h} = \mathbf{I}_{4h}^{8h}\tilde{\phi}^{4h}$;
- 9) Definir $\mathbf{f}^{8h} = \mathbf{A}^{8h}(\tilde{\phi}^{8h}) + \mathbf{r}^{8h}$;
- \vdots
- 10) Resolver $\mathbf{A}^{kh}(\tilde{\phi}^{kh}) = \mathbf{f}^{kh}$;
- 11) Aproximar o erro $\mathbf{e}^{kh} = \tilde{\phi}^{kh} - \tilde{\phi}_0^{kh}$;
- \vdots
- 12) Corrigir a solução $\tilde{\phi}_0^{4h} \leftarrow \tilde{\phi}^{4h} + \mathbf{I}_{8h}^{4h}\mathbf{e}^{8h}$;
- 13) Iterar $\mathbf{A}^{4h}(\tilde{\phi}^{4h}) = \mathbf{f}^{4h}$ ν_2 vezes com estimativa inicial $\tilde{\phi}_0^{4h}$;
- 14) Aproximar o erro $\mathbf{e}^{4h} = \tilde{\phi}^{4h} - \tilde{\phi}_0^{4h}$;
- 15) Corrigir a solução $\tilde{\phi}_0^{2h} \leftarrow \tilde{\phi}^{2h} + \mathbf{I}_{4h}^{2h}\mathbf{e}^{4h}$;
- 16) Iterar $\mathbf{A}^{2h}(\tilde{\phi}^{2h}) = \mathbf{f}^{2h}$ ν_2 vezes com estimativa inicial $\tilde{\phi}_0^{2h}$;
- 17) Aproximar o erro $\mathbf{e}^{2h} = \tilde{\phi}^{2h} - \tilde{\phi}_0^{2h}$;
- 18) Corrigir a solução $\tilde{\phi}_0^h \leftarrow \tilde{\phi}^h + \mathbf{I}_{2h}^h\mathbf{e}^{2h}$;
- 19) Iterar $\mathbf{A}^h(\tilde{\phi}^h) = \mathbf{f}^h$ ν_2 vezes com estimativa inicial $\tilde{\phi}_0^h$.

No algoritmo a seguir, são apresentados os passos básicos do procedimento FMG, para três malhas:

Algoritmo 3.4: FASFMG para três malhas.

FASFMG($\tilde{\phi}_0^{4h}, \nu_1, \nu_2$)

- 1) Resolver $\mathbf{A}^{4h}(\tilde{\phi}^{4h}) = \mathbf{f}^{4h}$ com estimativa inicial $\tilde{\phi}_0^{4h}$;

- 2) Calcular $\tilde{\phi}_0^{2h} = \mathbf{I}_{4h}^{2h} \tilde{\phi}^{4h}$;
- 3) Iterar $\mathbf{FAS}(\tilde{\phi}_0^{2h}, \nu_1, \nu_2)$;
- 4) Calcular e $\tilde{\phi}_0^h = \mathbf{I}_{2h}^h \tilde{\phi}^{2h}$;
- 5) Iterar $\mathbf{FAS}(\tilde{\phi}_0^h, \nu_1, \nu_2)$ até atingir algum critério de parada estabelecido.

Para o caso de modelos com dois ou mais sistemas de equações algébricas, como o caso das Equações de Burgers, o *multigrid* pode ser aplicado de formas diferentes. Uma maneira é aquela em que cada equação percorre o ciclo V separadamente, uma depois a outra, ao longo do processo iterativo. Na outra maneira, as equações percorrem o ciclo V simultaneamente, ou seja, em cada malha todos os passos descritos no Algoritmo 3.3 e no Algoritmo 3.4 são executados para todas as equações do sistema. Santiago (2010) mostrou que esta segunda maneira de implementar o *multigrid* para sistemas de equações diferenciais é mais eficiente.

CAPÍTULO 4

MODELOS MATEMÁTICOS E NUMÉRICOS

Neste capítulo apresentam-se os modelos matemáticos e numéricos empregados no desenvolvimento deste trabalho. Foram escolhidos três modelos matemáticos que fazem parte do escopo da dinâmica de fluidos computacional: as equações de Laplace, de Advecção-Difusão e de Burgers. Desta forma, têm-se um modelo linear puramente difusivo e dois modelos que apresentam advecção e difusão, um linear e outro não linear.

Inicialmente apresenta-se cada modelo matemático com suas respectivas condições de contorno, o domínio é dado por $\{(x, y) \in \mathbb{R}^2 : 0 \leq x, y \leq 1\}$ e ilustrado na Fig. 4.1. Em seguida, apresentam-se a configuração do problema teste usado com as equações de Burgers e os detalhes dos modelos numéricos.

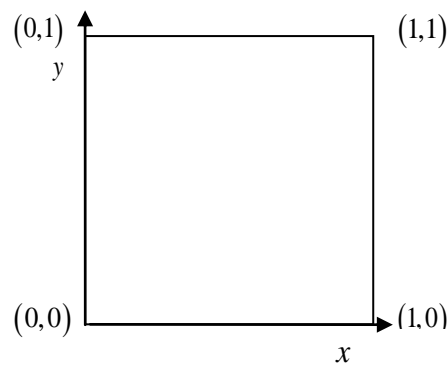


Figura 4.1: Domínio de cálculo

4.1 Modelos matemáticos

4.1.1 Equação de Laplace

Considera-se o problema linear de condução de calor bidimensional, em regime permanente, sem geração de calor, em coordenadas cartesianas, modelado pela equação de Laplace

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, \quad (4.1)$$

na qual $T(x, y)$ representa a temperatura em graus Celsius ($^{\circ}\text{C}$) e x e y são as coordenadas espaciais.

As condições de contorno, do tipo Dirichlet, são dadas por

$$T(x, 0) = T(1, y) = T(0, y) = 0 \quad (4.2)$$

e

$$T(x, 1) = \text{sen}(\pi x). \quad (4.3)$$

E a solução analítica,

$$T(x, y) = \text{sen}(\pi x) \frac{\text{senh}(\pi y)}{\text{senh}(\pi)}. \quad (4.4)$$

4.1.2 Equação de Advecção-Difusão

A equação de Advecção-Difusão é uma equação representativa de fenômenos de transporte. O modelo bidimensional, em regime permanente e em coordenadas cartesianas é dado por

$$Pe_x \frac{\partial T}{\partial x} + Pe_y \frac{\partial T}{\partial y} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + S(x, y), \quad (4.5)$$

onde Pe_x e Pe_y representam o número de Peclet nas direções das coordenadas x e y , respectivamente, dado por $Pe_x = \frac{\rho u c_p l}{k}$ e $Pe_y = \frac{\rho v c_p l}{k}$ sendo ρ a massa específica, u e v as velocidades nas direções x e y , respectivamente, c_p o calor específico, l o comprimento do domínio de cálculo e k a condutividade térmica. As simulações realizadas neste estudo tiveram como parâmetro $Pe_x = Pe_y = 2$. O termo fonte $S(x, y)$ é dado por (SCHNEIDER, 2007)

$$S(x, y) = \frac{\pi \left[(\sin(\pi x) \pi - Pe_x \cos(\pi x)) (e^{Pe_x x} - 1) - 2Pe_x \cos(\pi x) \right] (e^{Pe_y y} - 1)}{(e^{Pe_x} - 1)(e^{Pe_y} - 1)}, \quad (4.6)$$

as condições de contorno, do tipo Dirichlet, por

$$T(0, y) = T(x, 0) = T(1, y) = 0 \quad (4.7)$$

e

$$T(x, 1) = \sin(\pi x) \frac{(e^{Pe_x x} - 1)}{(e^{Pe_x} - 1)}. \quad (4.8)$$

e a solução analítica, por

$$T(x, y) = \sin(\pi x) \frac{(e^{Pe_x x} - 1)(e^{Pe_y y} - 1)}{(e^{Pe_x} - 1)(e^{Pe_y} - 1)}. \quad (4.9)$$

4.1.3 Equações de Burgers

As equações de Burgers são dadas por um sistema de equações diferenciais parciais não lineares, do tipo advecção-difusão. Consistem em uma forma simplificada das equações de Navier-Stokes, reduzidas às equações QML (Quantidade de Movimento Linear), pois uma vez que o campo de pressões é prescrito, a equação de conservação da massa não é necessária. De acordo com as hipóteses consideradas, podem ser classificadas como equações diferenciais elípticas, parabólicas ou hiperbólicas. Devido a estas características, são consideradas um importante problema teste. O modelo considerado neste trabalho tem solução analítica e campo de pressões prescrito, dados por Shih *et al.* (1989), os quais serão apresentados na sequência.

Considerando-se propriedades constantes, regime permanente e coordenadas cartesianas, as equações de Burgers na forma conservativa são escritas como

$$\frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (4.10)$$

$$\frac{\partial(uv)}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - S(x, y, \text{Re}), \quad (4.11)$$

onde p é a pressão estática, u e v são as componentes da velocidade nas direções coordenadas x e y , respectivamente, Re é o número de Reynolds, dado por $Re = \frac{U_{\infty} l}{\nu_c}$, sendo U_{∞} e l a velocidade e o comprimento de referência, $\nu_c = \frac{\mu_d}{\rho}$ a viscosidade cinemática, onde μ_d é a viscosidade dinâmica e ρ a massa específica. Neste estudo, as simulações foram realizadas com $Re = 1$. O termo fonte $S(x, y, Re)$ e o campo de pressões p , dados por Shih *et al.* (1989), são, respectivamente,

$$S(x, y, Re) = -\frac{8}{Re} [24F(x) + 2f'(x)g''(y) + f'''(x)g(y)] - 64[F_2(x)G_1(x) - g(x)g'(y)F_1(x)] \quad (4.12)$$

e

$$p(x, y, Re) = \frac{8}{Re} [F(x)g'''(y) + f'(x)g'(y)] + 64F_2 \{g(y)g''(y) - [g'(y)]^2\}, \quad (4.13)$$

com

$$\begin{cases} f(x) = x^4 - 2x^3 + x^2 \\ g(y) = y^4 - y^2 \\ F(x) = 0,2x^5 - 0,5x^4 + \frac{1}{3}x^3 \\ F_1(x) = -4x^6 + 12x^5 - 14x^4 + 8x^3 - 2x^2 \\ F_2(x) = 0,5(x^4 - 2x^3 + x^2)^2 \\ G_1(y) = -24y^5 + 8y^3 - 4y \end{cases} \quad (4.14)$$

As condições de contorno, do tipo Dirichlet, são

$$v(x, 0) = v(0, y) = v(1, y) = v(x, 1) = 0, \quad (4.15)$$

$$u(x, 0) = u(0, y) = u(1, y) = 0, \quad (4.16)$$

e

$$u(x, 1) = 16(x^4 - 2x^3 + x^2). \quad (4.17)$$

A solução analítica, obtida pelo método das soluções manufaturadas (ROY *et al.*, 2004), é dada pelas expressões

$$u(x, y) = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y) \quad (4.18)$$

e

$$v(x, y) = -8(4x^3 - 6x^2 + 2x)(y^4 - y^2). \quad (4.19)$$

Neste trabalho, as equações de Burgers são resolvidas em uma região fechada e o problema é conhecido como o problema da cavidade quadrada com tampa móvel. Neste caso, o fluido se desloca para a direita do contorno superior com uma distribuição de velocidades na direção x dada pela Eq. (4.18), enquanto que nos outros contornos permanece com velocidade nula (condição de não deslizamento). A Fig. 4.2 ilustra o domínio de cálculo deste problema. O deslocamento provocado no contorno superior cria dentro da cavidade um movimento circular das partículas do fluido, em torno de um ponto, chamado vórtice central.

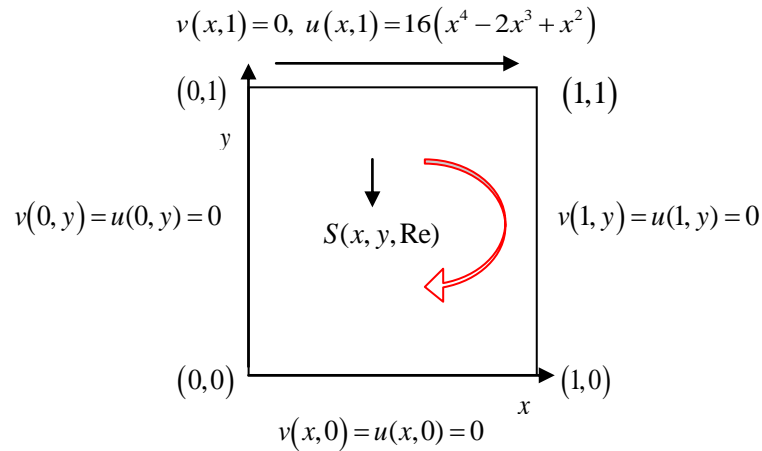


Figura 4.2: Domínio de cálculo e condições de contorno para o problema da cavidade quadrada com tampa móvel

4. 2 Modelos numéricos

Nesta seção são descritos os métodos, esquemas e procedimentos adotados para resolver numericamente os modelos matemáticos apresentados na seção anterior. São detalhados a discretização das equações, o tipo de malha, as aproximações numéricas, o critério de convergência, os operadores de transferência e o *solver* empregado.

4.2.1 Discretização das equações algébricas

Para obter as equações algébricas adota-se o MVF. A discretização do domínio é realizada mediante o uso de malhas estruturadas e uniformes por direção. Neste procedimento, os termos difusivos são aproximados com esquema CDS de segunda ordem. Os termos advectivos, por sua vez são aproximados implicitamente com UDS de primeira ordem, através do procedimento da correção adiada, Eq. (3.12). As condições de contorno, de Dirichlet, são aplicadas de acordo com a técnica dos volumes fictícios. Esta técnica, além de ser de fácil aplicação e respeitar o princípio de conservação para todo domínio (MALISKA, 2004), facilita a implementação dos procedimentos de restrição e prolongação no método *multigrid*. Considerando-se estas escolhas, são apresentados na sequência alguns passos realizados para a obtenção dos coeficientes e termos fontes que formam os sistemas de equações algébricas, para cada um dos modelos matemáticos.

4.2.1.1 Equação de Laplace

Realizando-se a integração da Eq. (4.1), obtém-se

$$\left(\frac{\partial T}{\partial x} \Big|_e - \frac{\partial T}{\partial x} \Big|_w \right) \Delta y + \left(\frac{\partial T}{\partial y} \Big|_n - \frac{\partial T}{\partial y} \Big|_s \right) \Delta x = 0. \quad (4.20)$$

Aproximando-se as derivadas com CDS de 2ª ordem, Eqs.(3.4)-(3.7), e rearranjando os termos da equação, chega-se a

$$2T_P \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) = \frac{1}{\Delta x^2} T_W + \frac{1}{\Delta x^2} T_E + \frac{1}{\Delta y^2} T_S + \frac{1}{\Delta y^2} T_N. \quad (4.21)$$

Colocando-a na forma linearizada

$$a_P T_P = a_W T_W + a_E T_E + a_S T_S + a_N T_N + b_P, \quad (4.22)$$

obtém-se os coeficientes para os volumes de controle internos ao domínio, dados por

$$a_P = 2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right), \quad a_W = \frac{1}{\Delta x^2}, \quad a_E = \frac{1}{\Delta x^2}, \quad a_S = \frac{1}{\Delta y^2}, \quad a_N = \frac{1}{\Delta y^2}, \quad (4.23)$$

e o termo fonte

$$b_P = 0. \quad (4.24)$$

Para os contornos, mediante a técnica dos volumes fictícios, descrita na seção 3.2.2, obtém-se

Leste

$$a_P = 1, \quad a_W = -1, \quad a_E = a_S = a_N = 0, \quad b_P = 0; \quad (4.25)$$

Oeste

$$a_P = 1, \quad a_E = -1, \quad a_W = a_S = a_N = 0, \quad b_P = 0; \quad (4.26)$$

Norte

$$a_P = 1, \quad a_S = -1, \quad a_W = a_E = a_N = 0, \quad b_P = 2 \sin(\pi x); \quad (4.27)$$

Sul

$$a_P = 1, \quad a_N = -1, \quad a_W = a_E = a_S = 0, \quad b_P = 0. \quad (4.28)$$

4.2.1.2 Advecção-Difusão

Realizando-se a integração da Eq.(4.5) e, considerando-se as aproximações descritas anteriormente, obtém-se

$$\begin{aligned} & Pe_x \left(T_P + \frac{1}{2} (T_E^* - T_P^*) \beta - T_W - \frac{1}{2} (T_P^* - T_W^*) \beta \right) \Delta y + \\ & Pe_y \left(T_P + \frac{1}{2} (T_N^* - T_P^*) \beta - T_S - \frac{1}{2} (T_P^* - T_S^*) \beta \right) \Delta x = \\ & \left(\frac{(T_E - T_P)}{\Delta x} - \frac{(T_P - T_W)}{\Delta x} \right) \Delta y + \left(\frac{(T_N - T_P)}{\Delta y} - \frac{(T_P - T_S)}{\Delta y} \right) \Delta x + S_P \Delta x \Delta y, \end{aligned} \quad (4.29)$$

onde β é o coeficiente da mistura entre os esquemas de aproximação UDS ($\beta = 0$) e CDS ($\beta = 1$), sendo que este último valor efetiva a correção adiada e o * faz referência aos valores obtidos na iteração anterior.

Colocando a Eq. (4.29) na forma linearizada

$$a_P T_P = a_W T_W + a_E T_E + a_S T_S + a_N T_N + b_P, \quad (4.30)$$

obtem-se os coeficientes dos volumes de controle internos ao domínio, dados por

$$a_P = a_W + a_E + a_S + a_N, \quad a_W = \frac{\Delta y}{\Delta x} + Pe_x \Delta y, \quad a_E = \frac{\Delta y}{\Delta x}, \quad a_S = \frac{\Delta x}{\Delta y} + Pe_y \Delta x, \quad (4.31)$$

e o termo fonte

$$S_P \Delta x \Delta y + 0,5\beta \left[Pe_x \Delta y (2T_P^* - T_W^* - T_E^*) + Pe_y \Delta x (2T_P^* - T_S^* - T_N^*) \right]. \quad (4.32)$$

Para os contornos, mediante a técnica dos volumes fictícios, obtém-se

Leste

$$a_P = 1, \quad a_W = -1, \quad a_E = a_S = a_N = 0, \quad b_P = 0; \quad (4.33)$$

Oeste

$$a_P = 1, \quad a_E = -1, \quad a_W = a_S = a_N = 0, \quad b_P = 0; \quad (4.34)$$

Norte

$$a_P = 1, \quad a_S = -1, \quad a_W = a_E = a_N = 0, \quad b_P = 2sen(\pi x)(e^{Pe_x} - 1)/(e^{Pe_x} - 1); \quad (4.35)$$

Sul

$$a_P = 1, \quad a_N = -1, \quad a_W = a_E = a_S = 0, \quad b_P = 0. \quad (4.36)$$

4.2.1.3 Equações de Burgers

Realizando-se a integração da equação QML para a propriedade ϕ (u ou v), com as aproximações descritas no início desta seção, chega-se a

$$\begin{aligned}
& \dot{M}_e [(1/2 + \alpha_e)\phi_P + (1/2 - \alpha_e)\phi_E] - \dot{M}_w [(1/2 + \alpha_w)\phi_W + (1/2 - \alpha_w)\phi_P] + \\
& \dot{M}_n [(1/2 + \alpha_n)\phi_P + (1/2 - \alpha_n)\phi_N] - \dot{M}_s [(1/2 + \alpha_s)\phi_S + (1/2 - \alpha_s)\phi_P] = \\
& \left\{ L[P^\phi]_P + L[S^\phi]_P \right\} \Delta x \Delta y \Delta z + \mu \left[\frac{(\phi_E - \phi_P)}{\Delta x} - \frac{(\phi_P - \phi_W)}{\Delta x} \right] \Delta y \Delta z + \\
& \mu \left[\frac{(\phi_N - \phi_P)}{\Delta y} - \frac{(\phi_P - \phi_W)}{\Delta x} \right] \Delta x \Delta z.
\end{aligned} \tag{4.37}$$

Colocando a Eq. (4.37) na forma linearizada,

$$a_P^\phi \phi_P = a_W^\phi \phi_W + a_E^\phi \phi_E + a_S^\phi \phi_S + a_N^\phi \phi_N + b_P^\phi, \tag{4.38}$$

obtem-se os coeficientes para os volumes de controle internos ao domínio, dados por

$$\left\{ \begin{aligned} a_P &= a_W + a_E + a_S + a_N \\ a_W &= (1/2 + \alpha_w)\dot{M}_w + \mu \frac{\Delta y \Delta z}{\Delta x} \\ a_E &= -(1/2 - \alpha_e)\dot{M}_e + \mu \frac{\Delta y \Delta z}{\Delta x} \\ a_S &= (1/2 + \alpha_s)\dot{M}_s + \mu \frac{\Delta x \Delta z}{\Delta y} \\ a_N &= -(1/2 - \alpha_n)\dot{M}_n + \mu \frac{\Delta x \Delta z}{\Delta y} \end{aligned} \right. \tag{4.39}$$

os quais são válidos para u e v , observando-se que

$$\alpha_w = \frac{1}{2} \text{sign}(u_w), \quad \alpha_e = \frac{1}{2} \text{sign}(u_e), \quad \alpha_n = \frac{1}{2} \text{sign}(v_n), \quad \alpha_s = \frac{1}{2} \text{sign}(u_s), \tag{4.40}$$

$$\dot{M}_w = \rho u_w \Delta y \Delta z, \dot{M}_e = \rho u_e \Delta y \Delta z, \dot{M}_n = \rho v_n \Delta x \Delta z, \dot{M}_s = \rho v_s \Delta x \Delta z, \quad (4.41)$$

$$u_w = \frac{u_p + u_w}{2}, u_e = \frac{u_p + u_e}{2}, v_s = \frac{v_p + v_s}{2}, v_n = \frac{v_p + v_n}{2}. \quad (4.42)$$

Os termos fonte de u e v são diferentes, dados por

$$b_p^u = -\frac{(p_e - p_w)}{2} \Delta y \Delta z \quad (4.43)$$

e

$$b_p^v = -\frac{(p_n - p_s)}{2} \Delta x \Delta z - S_p \Delta x \Delta y \Delta z + \beta \left[u_w \alpha_w (v_p^* - v_w^*) \Delta y - u_e \alpha_e (v_e^* - v_p^*) \Delta y + v_s \alpha_s (v_p^* - v_s^*) \Delta x - v_n \alpha_n (v_n^* - v_p^*) \Delta x \right]. \quad (4.44)$$

Para os contornos, mediante a técnica dos volumes fictícios, obtêm-se os coeficientes e termos fonte para u e v :

Leste

$$a_p = 1, a_w = -1, a_e = a_s = a_n = 0, b_p = 0; \quad (4.45)$$

Oeste

$$a_p = 1, a_e = -1, a_w = a_s = a_n = 0, b_p = 0; \quad (4.46)$$

Sul

$$a_p = 1, a_n = -1, a_w = a_e = a_s = 0, b_p = 0; \quad (4.47)$$

Norte

$$a_p = 1, a_s = -1, a_w = a_e = a_n = 0, \quad (4.48)$$

$$b_p^v = 0, b_p^u = 32(x^4 - 2x^3 + x^2). \quad (4.49)$$

4.2.2 Resolução dos sistemas de equações algébricas

Os sistemas de equações algébricas obtidos para cada um dos modelos matemáticos (Laplace, Advecção-Difusão e Burgers) são resolvidos com o método *multigrid* geométrico com esquema FAS-FMG e ciclo V, conforme algoritmos 3.3 e 3.4. Em um primeiro momento, o *solver* empregado é o GS-Lex, dado no algoritmo 3.1. Posteriormente, para o modelo dado pelas equações de Burgers, o desempenho do GS-Lex é comparado com o do GS-RB, algoritmo 3.2. Para obtenção das malhas, no *multigrid*, é empregada razão de refinamento $r = 2$.

O critério de parada adotado para as iterações externas (ciclos V) é baseado na razão entre a norma l_1 do resíduo da k -ésima iteração, $\mathbf{r}_{(k)}$, e a norma l_1 do resíduo da estimativa inicial, $\mathbf{r}_{(0)}$, (FERZIGER e PERIC, 2002):

$$\frac{\|\mathbf{r}_{(k)}\|_{l_1}}{\|\mathbf{r}_{(0)}\|_{l_1}} < \varepsilon. \quad (4.50)$$

Para estabelecer a tolerância ε é adotado o seguinte procedimento: na malha mais fina possível, considerando-se as limitações de memória física computacional, o programa é executado até eliminar o erro de iteração. Com a norma l_1 do resíduo, calculada em cada iteração, verifica-se o número de dígitos significativos, que não variam com as iterações (faixa do erro de arredondamento). Com essa verificação, obtém-se uma tolerância da ordem de 10^{-11} , que é duas ordens de grandeza acima do número de dígitos significativos verificados no erro de arredondamento. Para todos os problemas, adota-se a estimativa inicial com valor nulo para as variáveis dependentes.

4.2.3 Operadores de transferência

Em todos os problemas resolvidos, na restrição da solução da malha fina para a grossa é tomada a média aritmética das soluções nos quatro volumes de controle da malha fina que formam o volume de controle da malha grossa, conforme descrito na seção 3.4.3. Para o problema de Laplace, os resíduos são transferidos da malha fina para a malha grossa desta mesma forma. Já a restrição dos resíduos nos problemas Advecção-Difusão e Burgers é feita

mediante a soma dos resíduos dos quatro volumes de controle da malha fina que formam o volume de controle da malha grossa, também apresentado na seção 3.4.3.

O operador de prolongação adotado como padrão para os três problemas é o operador de prolongação bilinear, apresentado na seção 3.4.4. Ele é aplicado para prolongar as soluções de cada malha para o nível seguinte, no ciclo FMG, Fig. 3.16, e também para prolongar as correções de malha grossa para malha fina no ciclo V.

Para o problema descrito pelas equações de Burgers, o desempenho dos *solvers*, GS-Lex e GS-RB é considerado mediante diferentes operadores de prolongação.

Na prolongação das soluções para o nível seguinte, no FMG, costumam ser recomendados operadores de alta ordem (TROTTERBERG, 2001). Assim, são realizados testes com os operadores biquadrático e bicúbico. Para comparação, outros esquemas de interpolação descritos na literatura são testados: bilíres, *upwind*, *mixed* e *licen*, os quais serão apresentados na sequência. Para prolongar as correções entre as malhas no ciclo V não são necessários operadores de ordem muito alta (BRANDT, 2011), de forma que os esquemas selecionados, além do operador bilinear, são: biquadrático, bilíres, *upwind*, *mixed* e *licen*.

O operador de prolongação biquadrático é obtido de forma análoga ao operador bilinear, ou seja, interpolando-se polinômios quadráticos nas direções x e y . Alguns passos da obtenção das expressões para este operador são descritos a seguir.

Para determinar um polinômio que passe pelos pontos A, B e C na Fig. 4.3, consideram-se os polinômios de Lagrange (BURDEN e FAIRES, 2003):

$$L_0(x) = \frac{(x-x_B)(x-x_C)}{(x_A-x_B)(x_A-x_C)}, \quad L_1(x) = \frac{(x-x_A)(x-x_C)}{(x_B-x_A)(x_B-x_C)}, \quad L_2(x) = \frac{(x-x_A)(x-x_B)}{(x_C-x_A)(x_C-x_B)}. \quad (4.51)$$

O polinômio interpolador que passa por A, B e C é da forma

$$\phi(x) = \phi(x_A)L_0(x) + \phi(x_B)L_1(x) + \phi(x_C)L_2(x). \quad (4.52)$$

Substituindo-se a Eq.(4.51) na Eq. (4.52) e considerando-se as distâncias $(x_B - x_A) = (x_C - x_B) = h$ e $(x_C - x_A) = 2h$, obtém-se

$$\phi(x) = \frac{1}{2h^2} (\phi(x_A)(x-x_B)(x-x_C) - 2\phi(x_B)(x-x_A)(x-x_C) + \phi(x_C)(x-x_A)(x-x_B)). \quad (4.53)$$

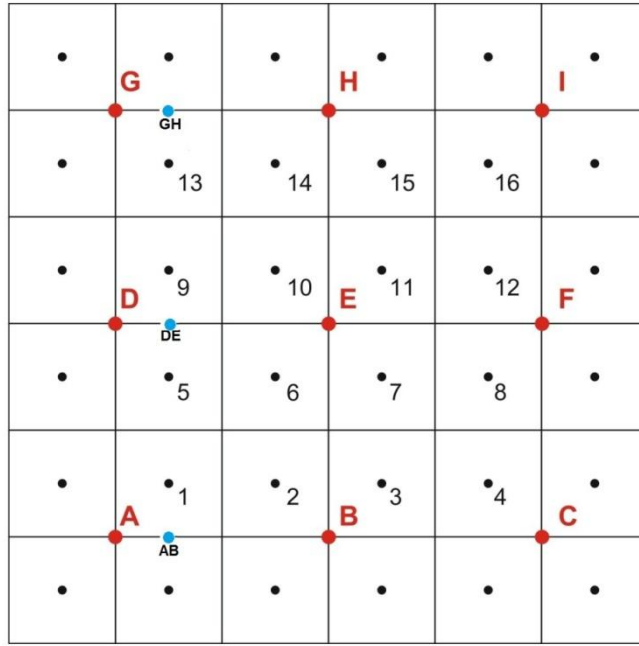


Figura 4.3: Esquema de malhas para operador biquadrático: malha grossa (nodos vermelhos) e malha fina (nodos pretos)

Com o polinômio dado na Eq. (4.53), interpola-se o ponto AB, em azul na Fig. 4.2

$$\phi(x_{AB}) = \frac{1}{2h^2} (\phi(x_A)(x_{AB} - x_B)(x_{AB} - x_C) - 2\phi(x_B)(x_{AB} - x_A)(x_{AB} - x_C) + \phi(x_C)(x_{AB} - x_A)(x_{AB} - x_B)). \quad (4.54)$$

Considerando-se que $(x_{AB} - x_A) = \frac{1}{4}h$, $(x_{AB} - x_B) = -\frac{3}{4}h$ e $(x_{AB} - x_C) = -\frac{7}{4}h$, obtém-se

se

$$\phi(x_{AB}) = \frac{21}{32}\phi(x_A) + \frac{14}{32}\phi(x_B) - \frac{3}{32}\phi(x_C). \quad (4.55)$$

Este procedimento também é realizado para os dois conjuntos de pontos D, E F e G, H e I, resultando nos respectivos polinômios interpoladores:

$$\phi(x) = \frac{1}{2h^2} (\phi(x_D)(x - x_E)(x - x_F) - 2\phi(x_E)(x - x_D)(x - x_F) + \phi(x_F)(x - x_D)(x - x_E)), \quad (4.56)$$

$$\phi(x) = \frac{1}{2h^2} (\phi(x_A)(x - x_B)(x - x_C) - 2\phi(x_B)(x - x_A)(x - x_C) + \phi(x_C)(x - x_A)(x - x_B)). \quad (4.57)$$

Com as Eqs. (4.56) e (4.57) são interpolados os pontos DE e GH, respectivamente, indicados na Fig. 4.3:

$$\phi(x_{DE}) = \frac{21}{32}\phi(x_D) + \frac{14}{32}\phi(x_E) - \frac{3}{32}\phi(x_F); \quad (4.58)$$

$$\phi(x_{GH}) = \frac{21}{32}\phi(x_G) + \frac{14}{32}\phi(x_H) - \frac{3}{32}\phi(x_I). \quad (4.59)$$

Em seguida, com procedimento análogo àquele considerado na obtenção da Eq. (4.53), obtém-se o polinômio que passa pelos pontos AB, DE e GH:

$$\begin{aligned} \phi(x) = \frac{1}{2h^2} & (\phi(x_{AB})(x - x_{DE})(x - x_{GH}) - 2\phi(x_{DE})(x - x_{AB})(x - x_{GH}) + \\ & \phi(x_{GH})(x - x_{AB})(x - x_{GH})) \end{aligned} \quad (4.60)$$

E, finalmente, pode-se obter o valor interpolado para o ponto 1 da malha fina, indicado na Fig.4.3:

$$\begin{aligned} \phi(x_1) = \frac{1}{2h^2} & (\phi(x_{AB})(x_1 - x_{DE})(x_1 - x_{GH}) - 2\phi(x_{DE})(x_1 - x_{AB})(x_1 - x_{GH}) + \\ & \phi(x_{GH})(x_1 - x_{AB})(x_1 - x_{GH})). \end{aligned} \quad (4.61)$$

Considerando-se as distâncias $(x_1 - x_{DE})$, $(x_1 - x_{GH})$ e $(x_1 - x_{AB})$, obtém-se

$$\phi(x_1) = \frac{21}{32}\phi(x_{AB}) + \frac{14}{32}\phi(x_{DE}) - \frac{3}{32}\phi(x_{GH}). \quad (4.62)$$

Substituindo-se as expressões dadas nas Eqs. (4.55), (4.58) e (4.59) na Eq. (4.62), tem-se o valor da propriedade ϕ no ponto 1 da malha fina, em termos de seus respectivos valores nos pontos A, B, C, D, E, F, G, H e I da malha grossa:

$$\begin{aligned} \phi^h(x_1) = \frac{1}{1024} & (441\phi^{2h}(x_A) + 294\phi^{2h}(x_B) - 63\phi^{2h}(x_C) + 294\phi^{2h}(x_D) + \\ & 196\phi^{2h}(x_E) - 42\phi^{2h}(x_F) - 63\phi^{2h}(x_G) - 42\phi^{2h}(x_H) + 9\phi^{2h}(x_I)). \end{aligned} \quad (4.63)$$

Os demais pontos da malha fina (2 a 16) são interpolados de maneira semelhante. No quadro a seguir são apresentadas essas interpolações. Para simplificar a notação, adota-se a convenção $\phi^k(x_i) = \phi_i^k$.

$$\begin{aligned}\phi_1^h &= \frac{1}{1024} (441\phi_A^{2h} + 294\phi_B^{2h} - 63\phi_C^{2h} + 294\phi_D^{2h} + 196\phi_E^{2h} - 42\phi_F^{2h} - 63\phi_G^{2h} - 42\phi_H^{2h} + 9\phi_I^{2h}) \\ \phi_2^h &= \frac{1}{1024} (105\phi_A^{2h} + 630\phi_B^{2h} - 63\phi_C^{2h} + 70\phi_D^{2h} + 420\phi_E^{2h} - 42\phi_F^{2h} - 15\phi_G^{2h} - 90\phi_H^{2h} + 9\phi_I^{2h}) \\ \phi_3^h &= \frac{1}{1024} (-63\phi_A^{2h} + 630\phi_B^{2h} + 105\phi_C^{2h} - 42\phi_D^{2h} + 420\phi_E^{2h} + 70\phi_F^{2h} + 9\phi_G^{2h} - 90\phi_H^{2h} - 15\phi_I^{2h}) \\ \phi_4^h &= \frac{1}{1024} (-63\phi_A^{2h} + 294\phi_B^{2h} + 441\phi_C^{2h} - 42\phi_D^{2h} + 196\phi_E^{2h} + 294\phi_F^{2h} + 9\phi_G^{2h} - 42\phi_H^{2h} - 63\phi_I^{2h}) \\ \phi_5^h &= \frac{1}{1024} (105\phi_A^{2h} + 70\phi_B^{2h} - 15\phi_C^{2h} + 630\phi_D^{2h} + 420\phi_E^{2h} - 90\phi_F^{2h} - 63\phi_G^{2h} - 42\phi_H^{2h} + 9\phi_I^{2h}) \\ \phi_6^h &= \frac{1}{1024} (25\phi_A^{2h} + 150\phi_B^{2h} - 15\phi_C^{2h} + 150\phi_D^{2h} + 900\phi_E^{2h} - 90\phi_F^{2h} - 15\phi_G^{2h} - 90\phi_H^{2h} + 9\phi_I^{2h}) \\ \phi_7^h &= \frac{1}{1024} (-15\phi_A^{2h} + 150\phi_B^{2h} + 25\phi_C^{2h} - 90\phi_D^{2h} + 900\phi_E^{2h} + 150\phi_F^{2h} + 9\phi_G^{2h} - 90\phi_H^{2h} - 15\phi_I^{2h}) \\ \phi_8^h &= \frac{1}{1024} (-15\phi_A^{2h} + 70\phi_B^{2h} + 105\phi_C^{2h} - 90\phi_D^{2h} + 420\phi_E^{2h} + 630\phi_F^{2h} + 9\phi_G^{2h} - 42\phi_H^{2h} - 63\phi_I^{2h}) \\ \phi_9^h &= \frac{1}{1024} (-63\phi_A^{2h} - 42\phi_B^{2h} + 9\phi_C^{2h} + 630\phi_D^{2h} + 420\phi_E^{2h} - 90\phi_F^{2h} + 105\phi_G^{2h} + 70\phi_H^{2h} - 15\phi_I^{2h}) \\ \phi_{10}^h &= \frac{1}{1024} (-15\phi_A^{2h} - 90\phi_B^{2h} + 9\phi_C^{2h} + 150\phi_D^{2h} + 900\phi_E^{2h} - 90\phi_F^{2h} + 25\phi_G^{2h} + 150\phi_H^{2h} - 15\phi_I^{2h}) \\ \phi_{11}^h &= \frac{1}{1024} (9\phi_A^{2h} - 90\phi_B^{2h} - 15\phi_C^{2h} - 90\phi_D^{2h} + 900\phi_E^{2h} + 150\phi_F^{2h} - 15\phi_G^{2h} + 150\phi_H^{2h} + 25\phi_I^{2h}) \\ \phi_{12}^h &= \frac{1}{1024} (9\phi_A^{2h} - 42\phi_B^{2h} - 63\phi_C^{2h} - 90\phi_D^{2h} + 420\phi_E^{2h} + 630\phi_F^{2h} - 15\phi_G^{2h} + 70\phi_H^{2h} + 105\phi_I^{2h}) \\ \phi_{13}^h &= \frac{1}{1024} (-63\phi_A^{2h} - 42\phi_B^{2h} + 9\phi_C^{2h} + 294\phi_D^{2h} + 196\phi_E^{2h} - 42\phi_F^{2h} + 441\phi_G^{2h} + 294\phi_H^{2h} - 63\phi_I^{2h}) \\ \phi_{14}^h &= \frac{1}{1024} (-15\phi_A^{2h} - 90\phi_B^{2h} + 9\phi_C^{2h} + 70\phi_D^{2h} + 420\phi_E^{2h} - 42\phi_F^{2h} + 105\phi_G^{2h} + 630\phi_H^{2h} - 63\phi_I^{2h}) \\ \phi_{15}^h &= \frac{1}{1024} (9\phi_A^{2h} - 90\phi_B^{2h} - 15\phi_C^{2h} - 42\phi_D^{2h} + 420\phi_E^{2h} + 70\phi_F^{2h} - 63\phi_G^{2h} + 630\phi_H^{2h} + 105\phi_I^{2h}) \\ \phi_{16}^h &= \frac{1}{1024} (9\phi_A^{2h} - 42\phi_B^{2h} - 63\phi_C^{2h} - 42\phi_D^{2h} + 196\phi_E^{2h} + 294\phi_F^{2h} - 63\phi_G^{2h} + 294\phi_H^{2h} + 441\phi_I^{2h})\end{aligned}$$

Para obtenção das expressões para o operador de prolongação bicúbico são realizados os mesmos passos, isto é, interpolações de polinômios cúbicos nas direções x e y . A representação das malhas relativas a este operador, bem como as expressões obtidas, constam no Apêndice B deste texto.

Além dos operadores biquadrático e bicúbico, foram buscados na literatura outros operadores de interpolação. O operador que denominamos bilires é um operador baseado no resíduo (r), adaptado de Liu (2010). A Fig. 4.4 mostra a disposição das malhas para a aplicação deste operador, dado por

$$\mathbf{I}_{2h}^h \phi^{2h} = \begin{cases} \frac{1}{16+h^2} (9\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + \phi_D^{2h} + h^2 r_1^h) & \text{para } \phi_1^h \\ \frac{1}{16+h^2} (3\phi_A^{2h} + 9\phi_B^{2h} + \phi_C^{2h} + 3\phi_D^{2h} + h^2 r_2^h) & \text{para } \phi_2^h \\ \frac{1}{16+h^2} (3\phi_A^{2h} + \phi_B^{2h} + 9\phi_C^{2h} + 3\phi_D^{2h} + h^2 r_3^h) & \text{para } \phi_3^h \\ \frac{1}{16+h^2} (\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + 9\phi_D^{2h} + h^2 r_4^h) & \text{para } \phi_4^h \end{cases}. \quad (4.64)$$

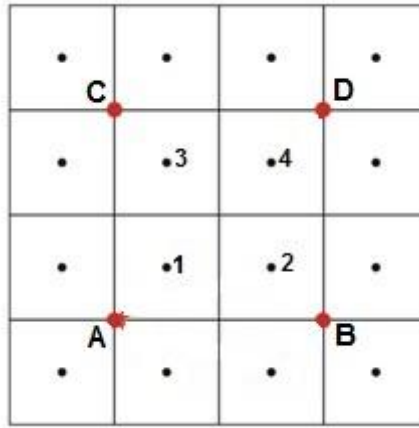


Figura 4.4: Esquema de malhas para o bilires

O operador *upwind* é aplicado conforme Drikakis (1998), ilustrado na Fig. 4.5, onde $\phi_A^{2h} > 0$ e é dado por

$$\mathbf{I}_{2h}^h \phi^{2h} = \begin{cases} \phi_A^{2h} & \text{se } \phi_A^{2h} > 0 & \text{para } \phi_1^h, \phi_2^h, \phi_3^h \text{ e } \phi_4^h \\ \phi_B^{2h} & \text{se } \phi_A^{2h} < 0 & \text{para } \phi_1^h, \phi_2^h, \phi_3^h \text{ e } \phi_4^h \end{cases} \quad (4.65)$$

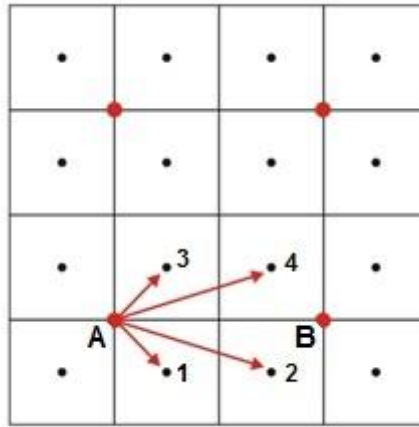


Figura 4.5: Aplicação do *upwind* para $\phi_A^{2h} > 0$

O operador *mixed*, proposto por Drikakis (1998), consiste na interpolação linear (na direção vertical) seguida de *upwind*. Para os pontos 1 e 2 na Fig. 4.6, as expressões são

$$\mathbf{I}_{2h}^h \phi^{2h} = \begin{cases} \frac{3}{4} \phi_A^{2h} + \frac{1}{4} \phi_C^{2h} & \text{se } \phi_I^{2h} > 0 \quad \text{para } \phi_1^h \text{ e } \phi_2^h \\ \frac{3}{4} \phi_B^{2h} + \frac{1}{4} \phi_D^{2h} & \text{se } \phi_I^{2h} < 0 \quad \text{para } \phi_1^h \text{ e } \phi_2^h \end{cases}. \quad (4.66)$$

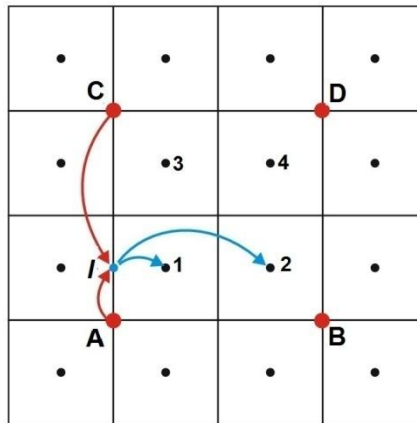


Figura 4.6: Aplicação do *mixed* para $\phi_I^{2h} > 0$

Como uma adaptação do *mixed*, apresentado por Drikakis (1998), considera-se o esquema que designamos por *licen* que consiste em uma interpolação linear na direção vertical seguida da atribuição do valor interpolado para seus vizinhos a oeste e a leste na malha fina, conforme pode ser visualizado na Fig.4.7. As expressões para este operador são

$$\mathbf{I}_{2h}^h \phi^{2h} = \begin{cases} \frac{1}{4}(3\phi_A^{2h} + \phi_C^{2h}) & \text{para } \phi_1^h \text{ e } \phi_2^h \\ \frac{1}{4}(\phi_A^{2h} + 3\phi_C^{2h}) & \text{para } \phi_3^h \text{ e } \phi_4^h \end{cases}. \quad (4.67)$$

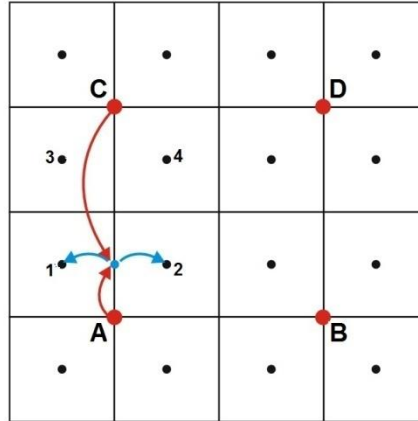


Figura 4.7: Aplicação do licen

4.3 Dados da implementação computacional

Todos os códigos deste trabalho foram desenvolvidos na linguagem FORTRAN/95 com o compilador Compaq Visual Fortran 2005. Para cada problema, foi escrito um código computacional compilado na versão *release*, projeto tipo Console Application.

Todas as simulações foram realizadas no microcomputador CFD-14 do Laboratório de Experimentação Numérica (LENA) da UFPR, com processador AMD Athlon Core 2 Duo com 2.2 GHz e 2 GB RAM, usando aritmética de dupla precisão em sistema operacional Windows XP de 64 bits.

Mesmo nas malhas mais refinadas, as simulações foram realizadas sem a necessidade do uso de memória virtual. A memória computacional foi monitorada através do gerenciador de tarefas do Windows. O tempo de CPU foi medido usando-se a função CPU_time do FORTRAN 95 e corresponde ao tempo para gerar malhas, atribuir estimativa inicial, calcular coeficientes e termos fonte, resolver os sistemas de equações e verificar o critério de convergência.

CAPÍTULO 5

RESULTADOS

Neste capítulo é apresentada a análise dos parâmetros investigados para o método *multigrid*. Dentre as centenas de simulações realizadas, são apresentados os resultados mais representativos. Como objetivo principal, investiga-se a influência causada pelo número de iterações internas (v) do *solver*, do número de malhas (L), do número de incógnitas (N) e dos operadores de prolongação sobre o tempo de CPU. Em todas as malhas, as simulações são realizadas com um número de malhas L tal que $1 \leq L \leq L_{máx}$, onde $L = 1$ corresponde ao *singlegrid* e $L = L_{máx}$ corresponde ao método *multigrid* percorrendo todas as malhas possíveis do ciclo V. O parâmetro $L_{máx}$ representa o maior número possível de malhas que podem ser empregadas para um determinado problema, com apenas quatro volumes de controle reais na malha mais grossa (2×2). Para exemplificar, se $N = 256 \times 256$ volumes de controle, o maior número possível de malhas é $L_{máx} = 8$ e o método *multigrid* percorre as malhas 128×128 , 64×64 , 32×32 , 16×16 , 8×8 , 4×4 e 2×2 . A análise do desempenho do método *multigrid*, com relação ao *singlegrid*, é geralmente medida pelo fator de aceleração S , definido no Capítulo 1 deste texto. Nos casos em que o tempo de CPU é inferior a 10 segundos, as simulações são repetidas até atingir 10 segundos, por meio da sub-rotina *repete* introduzida no programa. O tempo de CPU é então calculado tomando-se a média aritmética dos tempos obtidos em cada repetição.

As análises são divididas em duas etapas, descritas a seguir.

Etapa I – Concentra-se em um estudo preliminar em que são feitas análises do método *multigrid* nos problemas bidimensionais de Laplace, Advecção-Difusão e Burgers. Neste estudo preliminar, busca-se verificar se a eficiência do *multigrid* sofre degradação devido às características de cada problema, como não linearidade e número de equações. Todos os problemas são resolvidos com o *solver* GS-Lex e algoritmo FAS-FMG com ciclo V. Nesta etapa, os operadores de restrição para solução e resíduos são a soma e a média aritmética dos valores da propriedade nos quatro volumes de controle da malha fina que formam o volume de controle da malha grossa, apresentados na seção 3.4.3. O operador de prolongação adotado como padrão é o operador de prolongação bilinear. Ele é aplicado para prolongar as soluções de cada malha para o nível seguinte, no ciclo FMG, e também para prolongar as correções de

malha grossa para malha fina no ciclo V. São investigados os valores ótimos para os parâmetros v , L e N . No estudo dos valores ótimos para os parâmetros v e L são consideradas malhas de três tamanhos: pequeno, médio e grande, com 128×128 , 512×512 e 2048×2048 incógnitas, respectivamente. Nas análises sobre o tamanho do problema (N) são consideradas malhas de 16×16 a 2048×2048 incógnitas, que é a malha mais fina possível, dadas as limitações de memória computacional. Os resultados do tempo de CPU empregado na resolução dos problemas com o algoritmo FAS-FMG são comparados com os respectivos resultados obtidos com os algoritmos SG e FAS e o desempenho do *multigrid* é avaliado para os três problemas resolvidos.

Etapa II – Concentra-se na análise dos parâmetros do método *multigrid* aplicado ao modelo bidimensional das equações de Burgers. O melhor algoritmo para este problema, definido na Etapa I, com *solver* GS-Lex e FAS-FMG com ciclo V e parâmetros ótimos é adotado como padrão para os testes adicionais desta etapa. O primeiro estudo consiste na comparação do desempenho deste algoritmo com os *solvers* GS-Lex e GS-RB. Os valores dos parâmetros ótimos, v e L , para o GS-RB são definidos e o desempenho dos dois *solvers* é comparado. O segundo estudo consiste na investigação do efeito de operadores de prolongação sobre o tempo de CPU. Com o objetivo de melhorar o desempenho do FAS-FMG são testados diferentes operadores de prolongação da solução, ao final de cada ciclo do FMG, que será adotada como estimativa inicial para o ciclo seguinte. Os operadores testados, comparados com o operador definido como padrão na etapa anterior (bilinear) são: o biquadrático, o bicúbico, o *upwind*, o *mixed*, o bilires e o *licen*. Todos os operadores escolhidos, combinados com os operadores de restrição descritos na etapa anterior, e adotados também nesta etapa, satisfazem a Eq. (3.39). As ordens de aproximação destes operadores, conforme definido na seção 3.4.4 são: 1ª ordem (*upwind*, *mixed* e *licen*), 2ª ordem (bilinear e bilires), e 3ª ordem (biquadrático) e 4ª ordem (bicúbico). Todos os operadores são testados com os *solvers* GS-Lex e GS-RB. O terceiro estudo consiste na investigação do efeito dos operadores de prolongação para as correções entre as malhas sobre o desempenho do FAS-FMG. Nestes testes são empregados os seguintes operadores: bilinear, biquadrático, *upwind*, *mixed*, bilires e *licen*.

5.1 ETAPA I: Laplace, Advecção-Difusão e Burgers

5.1.1 – Número de iterações internas (v)

Vários critérios podem ser adotados para definir o número de iterações internas (v) ou de suavizações do *solver* em cada malha. O mais comum é o critério de ciclo, que consiste em fixar o número de iterações em cada malha. Oliveira *et al.* (2008) apresentaram um estudo em que investigaram vários critérios apontados na literatura, como Hortmann, dente de serra e variações. Deste estudo concluíram que o melhor tempo de CPU é obtido quando se usa um número fixo de iterações do *solver*. Nesta seção é apresentado o estudo do número ótimo de iterações internas (v) do *solver* sobre o tempo de CPU. Para todas as malhas consideradas, são realizadas simulações com um número fixo de iterações internas do *solver* variando de $v = 1$ até 10, e dois valores isolados $v = 15$ e $v = 20$ para confirmar a tendência do parâmetro. O valor de v que resulta no menor tempo de CPU é considerado o número ótimo de iterações internas, denotado por $v_{ótimo}$. Em todas as simulações realizadas para este parâmetro empregou-se $L = L_{máx}$ e critério de parada dado pela Eq.(4.50).

A Fig. 5.1 ilustra a influência do número de iterações internas do *solver* para as equações de Laplace, Advecção-Difusão e Burgers.

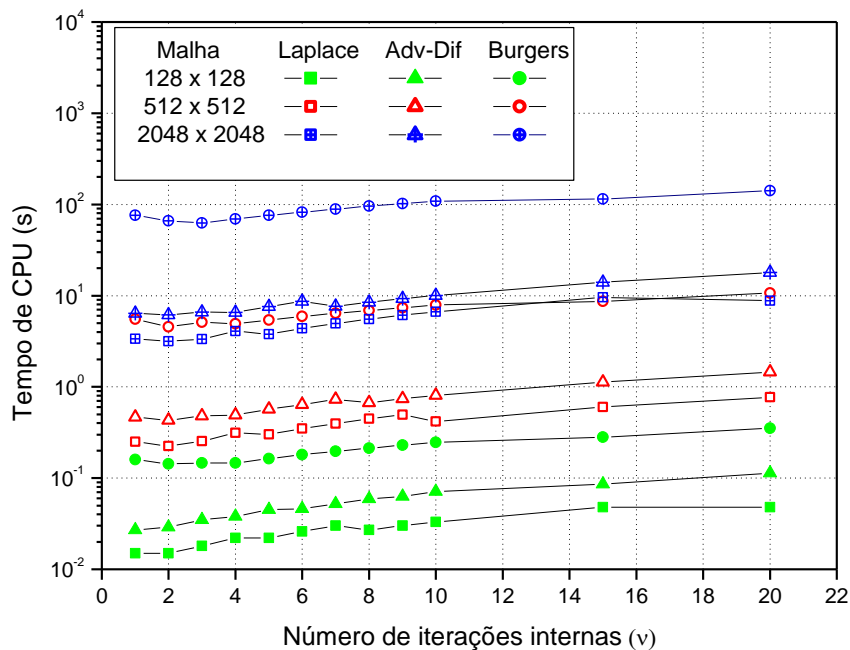


Figura 5.1: Efeito de v sobre o tempo de CPU – FAS-FMG e *solver* GS-Lex

Como pode ser observado na Fig. 5.1, o tempo de CPU exibe uma tendência de crescimento com o número de iterações internas. Nos problemas de Laplace e Advecção-Difusão o menor tempo de CPU é obtido com duas iterações internas, ou seja, $v_{\text{ótimo}} = 2$ para as duas malhas mais finas e $v_{\text{ótimo}} = 1$ para a malha mais grossa. Para o problema de Burgers, o menor tempo de CPU é obtido com 2 ou 3 iterações internas. Portanto $v_{\text{ótimo}} = 2$ ou 3 para este caso.

Alguns resultados sobre o valor de $v_{\text{ótimo}}$ para a equação de Laplace bidimensional, encontrados na literatura, são: Tannehill et al. (1997) com o MDF e esquema CS obtiveram $v_{\text{ótimo}} = 3$ ou 4 com o esquema FAZ; Pinto e Marchi (2006) com o MDF encontraram $v_{\text{ótimo}} = 1$ ou 2 com esquema CS e $v_{\text{ótimo}} = 4$ com esquema FAZ; Oliveira *et al.*, (2008) com MDF e esquema FAS e Suero (2010) com MVF e AMG obtiveram $v_{\text{ótimo}} = 2$. Para a equação de Advecção-Difusão bidimensional, Rabi e De Lemos (2001) obtiveram, com o MVF e esquema CS, $v_{\text{ótimo}} = 1$ com ciclo V e $v_{\text{ótimo}} = 2$ com ciclo W. Santiago (2010), com o MDF e esquema FAS, obteve $v_{\text{ótimo}} = 5$ com esquema FAS, para o problema de Burgers bidimensional.

Considerando-se os casos citados, observa-se a sensibilidade deste parâmetro com relação a outros aspectos do modelo numérico, como método de discretização, esquema de aproximação (CS ou FAS) e tipo de ciclo (V ou W). Assim, deve-se observar a importância da definição deste parâmetro. Quando se diminui ou aumenta o valor de v em relação ao ótimo, há um acréscimo no tempo de CPU, que pode ser significativo de acordo com o valor empregado. Por exemplo, para o problema de Laplace, o emprego de $v = 4$ ao invés do valor encontrado neste trabalho $v_{\text{ótimo}} = 2$ implica em um aumento de aproximadamente 30% no tempo de CPU na malha 2048×2048 . A Tab. 5.1 resume os resultados obtidos para este parâmetro neste trabalho.

Tabela 5.1 – Valor de $v_{\text{ótimo}}$ para cada problema

Problema	$v_{\text{ótimo}}$
Laplace	2
Advecção-Difusão	2
Burgers	2 ou 3

Deste ponto em diante, para o problema de Burgers, o valor ótimo adotado nos estudos seguintes é $v_{\text{ótimo}} = 3$. Esta escolha tem por base testes adicionais considerando-se outras malhas, dos quais se conclui que o parâmetro $v_{\text{ótimo}}$ tende assintoticamente para este valor.

5.1.2 Número de malhas (L)

Outro aspecto importante na composição do algoritmo do *multigrid* é a seleção das malhas. Esta seleção é baseada no tamanho do problema e na razão de refinamento adotada. Considerando-se um problema com razão de refinamento $r = 2$ com $N = 64 \times 64$ incógnitas, o maior número possível de malhas ($L_{\text{máx}}$) é 6. São elas: 2×2 , 4×4 , 8×8 , 16×16 , 32×32 e 64×64 .

Para a equação de Laplace, Oliveira *et al.*, (2008) e Suero (2010) com o MDF, obtiveram $L = L_{\text{máx}}$ como valor ótimo para este parâmetro, empregado o GMG e o AMG, respectivamente. Tannehill *et al.* (1997), com o MDF, verificaram, para a equação de Laplace com 128×128 incógnitas, que não há redução significativa no tempo de CPU com o emprego do número máximo de malhas e sugerem que sejam empregadas apenas 3 ou 4. Este resultado é confirmado pelo trabalho de Santiago (2010), que obteve com o MDF, para os problemas de Laplace, Navier e Burgers, $L_{\text{ótimo}} = L_{\text{máx}} - (0 \text{ a } 4)$. Com o MVF, Rabi e De Lemos (2001) resolveram o problema de Advecção-Difusão com número máximo de seis malhas e sugerem o emprego de pelo menos 4 malhas para uma boa suavização dos erros. Por outro lado, Kumar *et al.* (2009) afirmaram que não há ganho com mais do que 4 malhas para o problema da cavidade com tampa móvel e MVF com 513×513 incógnitas.

No estudo da influência do número de malhas sobre o desempenho do *multigrid*, considera-se o $v_{\text{ótimo}}$ obtido na seção anterior. A Fig. 5.2 ilustra o efeito do número de malhas sobre o tempo de CPU para as equações de Laplace, Advecção-Difusão e Burgers.

Pode-se observar na Fig. 5.2 que, independente do tipo e do tamanho do problema, o valor ótimo para o número de malhas é $L_{\text{ótimo}} = L_{\text{máx}} - 1$. Além disso, para $L < L_{\text{ótimo}} - 1$ o tempo de CPU tende a crescer significativamente com a redução do número de malhas. Nota-se que a diferença, em todos os casos (tipo de malha ou problema), está sempre entre, aproximadamente, 1 e 1,5 ordem de grandeza, ou seja, entre 10 e 50 vezes.

De acordo com os resultados obtidos neste estudo, o problema de Laplace na malha 128×128 com emprego de $L_{\text{ótimo}} = L_{\text{máx}} = 7$ converge cerca de 23 vezes mais rápido do que com o emprego de apenas 4 malhas. Para o problema de Burgers, na malha 512×512 , o

emprego de $L_{\text{ótimo}} = L_{\text{máx}} = 8$ converge aproximadamente 190 vezes mais rápido do que com apenas 4 malhas.

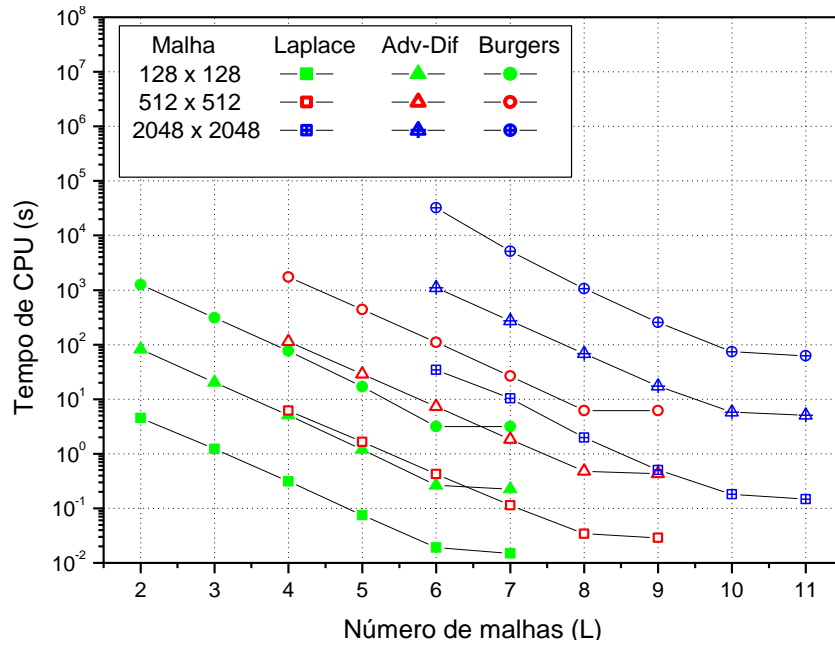


Figura 5.2: Efeito de L sobre o tempo de CPU – FAS-FMG e *solver* GS-Lex

5.1.3 Número de incógnitas (N)

De acordo com Ferziger e Peric (2001), quanto mais refinada a malha, isto é, quanto maior o número de incógnitas N , maior é a “vantagem” do método *multigrid* com relação ao método *singlegrid*. Com o intuito de avaliar o desempenho do algoritmo FAS-FMG perante o SG, inicialmente é analisada a influência do número de incógnitas no sistema de equações sobre o tempo de CPU. Para esta análise, são considerados os valores do número ótimo de iterações internas obtidos na seção 5.1.1 e o número ótimo de malhas obtido na seção 5.1.2. Na Fig. 5.3 constam os resultados do tempo de CPU empregado na resolução dos problemas de tamanho 16×16 a 2048×2048 , com os algoritmos FAS-FMG e SG, sendo que os resultados deste último são relativos às malhas 16×16 até, no máximo 1024×1024 , conforme o problema. Na Fig. 5.3 pode-se observar que as inclinações das curvas do FAS-FMG, para cada problema, são menores do que as das respectivas curvas do SG, o que evidencia a afirmação citada anteriormente, sobre a melhora no desempenho do FAS-FMG com o aumento do número de incógnitas. Considerando-se a malha mais fina, o tempo de CPU observado para o problema de Burgers com o FAS-FMG é de aproximadamente 10 a 20

vezes maior que o tempo observado para os problemas de Advecção-Difusão e de Laplace, respectivamente.

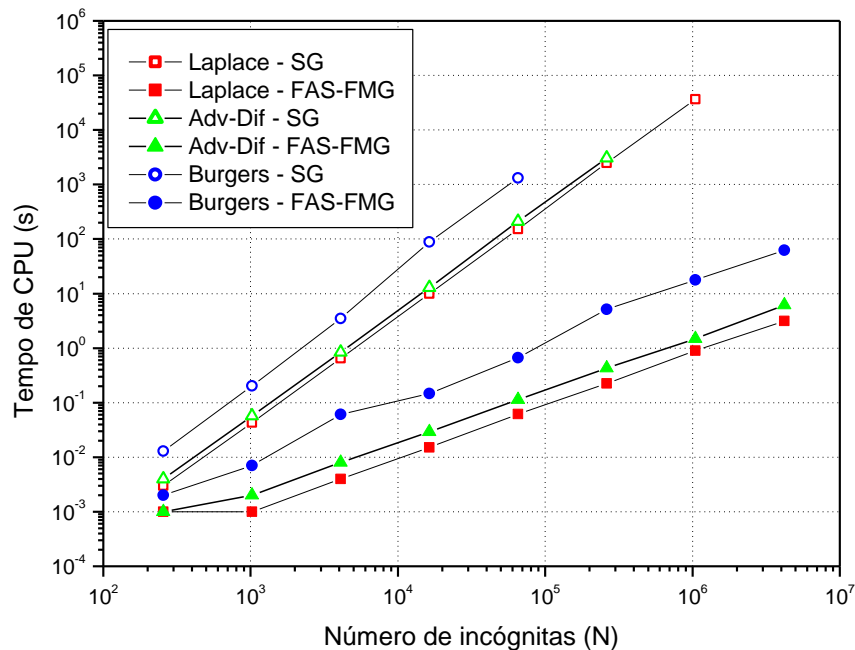


Figura 5.3: Efeito do número de incógnitas sobre o tempo de CPU

Nas próximas seções, 5.1.4 e 5.15, os resultados do FAS-FMG são comparados com o FAS padrão e o esforço computacional para os três algoritmos é apresentado.

5.1.4 Comparação entre os esquemas FAS e FAS-FMG

Segundo Trottenberg *et al.* (2001), o FMG é a mais eficiente versão do *multigrid*. Lien e Leschziner (1994) e Zhang *et al.* (2010) observaram que, de um modo geral, o FAS-FMG é aproximadamente 2 vezes mais rápido do que o FAS padrão.

Com o intuito de verificar o quanto a técnica FMG melhora o desempenho do MG nos problemas de Laplace, Advecção-Difusão e Burgers, são comparados os algoritmos FAS e FAS-FMG. Na Fig. 5.4, são apresentados os resultados do tempo de CPU obtidos com o FAS e com o FAS-FMG para os problemas de tamanho 16×16 a 2048×2048 .

Conforme o esperado, em todos os casos o FAS-FMG é significativamente mais rápido do que o FAS padrão. Porém, o efeito de sua aplicação é diferente para cada problema. De acordo com os resultados obtidos neste trabalho, para o problema de Laplace a redução no tempo de CPU relativo ao emprego do FMG varia de 45 a 96%, enquanto que para o problema da Advecção-Difusão varia de 31% a 73% e para o problema de Burgers de 28 a 105%, considerando-se todas as malhas testadas. Em todos os casos, quanto mais refinada é a

malha, maior é a redução no tempo de CPU com a aplicação do FAS-FMG, em relação ao FAS.

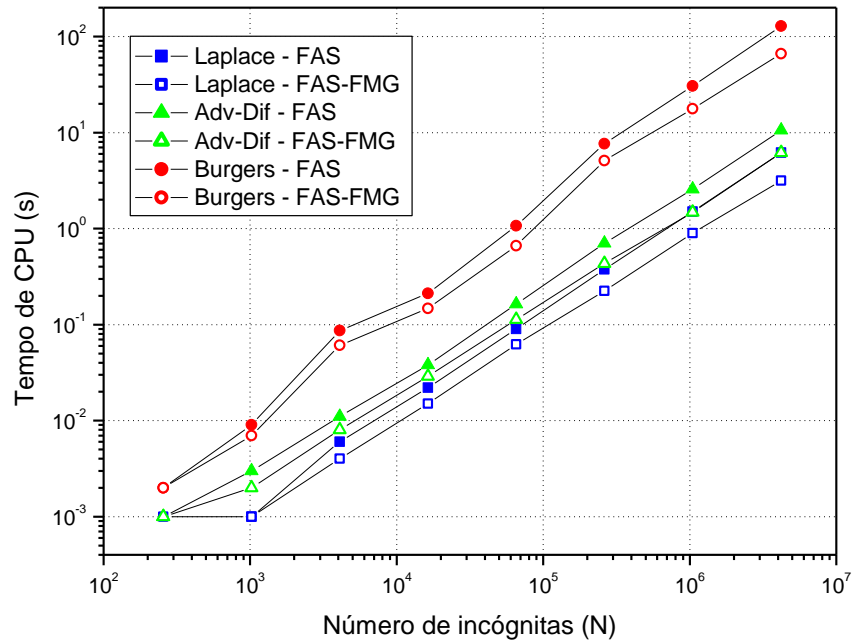


Figura 5.4: Efeito do número de malhas sobre o tempo de CPU

A Tab. 5.2 traz os resultados do tempo de CPU dispensado na resolução de cada problema, segundo cada algoritmo (SG, FAS ou FAS-FMG) para as malhas 128×128 , 512×512 e 2048×2048 . Adicionalmente, apresenta o fator de aceleração⁵ obtido em cada caso, com relação ao SG. Os valores indicados por * são estimados a partir da expressão tempo de $CPU(N) = cN^p$. O procedimento para obtenção desta curva, assim como os coeficientes obtidos para cada problema estão descritos na seção 5.1.5, a seguir.

⁵ De uma forma mais geral do que aquela definida no Capítulo 1, redefine-se o fator de aceleração (S), como a razão entre o tempo de CPU de um algoritmo e o tempo de CPU de outro algoritmo.

Tabela 5.2: Fatores de aceleração S do *multigrid* frente ao *singlegrid* com os esquemas FAS e FAS-FMG

Problema	Malha	SG	Tempo de CPU (s)		S	
			FAS	FAS-FMG	FAS	FAS-FMG
Laplace	128 × 128	9,98	0,022	0,015	454	666
Advecção-Difusão		12,91	0,038	0,029	340	445
Burgers		69,33	0,212	0,147	327	472
Laplace	512 × 512	2478,27	0,375	0,225	6609	11014
Advecção-Difusão		3049,77	0,703	0,432	4338	7059
Burgers		25281,72	7,633	5,117	3312	4940
Laplace	2048 × 2048	541354,51 [*]	6,188	3,156	87484	171531
Advecção-Difusão		698888,72 [*]	10,578	6,125	66070	114104
Burgers		858559,78 [*]	128,620	62,625	6675	13709

* valor estimado

Para os três problemas estudados, os resultados mostram que o desempenho do *multigrid* melhora significativamente nas malhas mais refinadas e o fator de aceleração, com relação ao SG, é maior com o emprego do FAS-FMG do que com o do FAS.

5.1.5 Esforço computacional

Para determinar a ordem de complexidade dos algoritmos relativos a cada método e o comportamento da curva tempo de CPU x N é feito o ajuste de uma curva do tipo tempo de $CPU(N) = cN^p$, onde p representa a ordem do algoritmo, ou a inclinação da curva e c é uma constante que depende do método. Quanto menor for o valor de p , melhor é o desempenho do algoritmo empregado. No caso ideal, o método *multigrid* apresenta $p = 1$, o que significa que o tempo de CPU aumenta proporcionalmente com o crescimento do número de incógnitas (N). A Tab.5.3 mostra os valores de c e p obtidos pelo ajuste de curva, descrito anteriormente, para os três problemas estudados, empregando-se os esquemas FAS e FAS-FMG com GS-Lex. No cálculo destes coeficientes para o *multigrid* são consideradas as malhas a partir de $N > 32 \times 32$ e para o SG são considerados os seis problemas de maior tamanho, com solução disponível, para cada modelo matemático.

Tabela 5.3: Coeficientes com *solver* GS-Lex

Problema	SG		MG			
			FAS		FAS-FMG	
	c	p	c	p	c	p
Laplace	$4,86 \times 10^{-8}$	1,97	$1,32 \times 10^{-6}$	1,00	$1,29 \times 10^{-6}$	0,96
Advecção-Difusão	$7,30 \times 10^{-8}$	1,96	$2,56 \times 10^{-6}$	0,99	$2,77 \times 10^{-6}$	0,95
Burgers	$1,03 \times 10^{-7}$	2,10	$6,94 \times 10^{-6}$	1,11	$6,94 \times 10^{-6}$	1,06

Os resultados confirmam que o tempo de CPU do método *multigrid* e *solver* GS-Lex cresce linearmente com o aumento de N , conforme o esperado. Sendo que o algoritmo FAS-FMG é o que apresenta o melhor desempenho. Por outro lado, os valores de p próximos de dois para o *singlegrid* concordam com os valores teóricos (BURDEN e FAIRES, 2003).

Dadas às características dos problemas estudados, Laplace, Advecção-Difusão, e Burgers, nota-se que o valor de p é pouco afetado pelo número de equações ou sua complexidade. O efeito mais relevante na eficiência do algoritmo é o emprego ou não do *multigrid*.

5.2 Etapa II – Equações de Burgers

5.2.1 Comparativo GS-Lex e GS-RB

Tendo por motivação os bons resultados obtidos por Oliveira (2010) com o *solver* GS-RB, optou-se por investigar o efeito deste *solver* no algoritmo FAS-FMG, aplicando-o ao modelo das equações de Burgers na Etapa I. No trabalho citado, verificou-se que GS-RB é cerca de duas vezes mais rápido do que o GS-Lex, considerando-se a resolução da equação de Laplace em uma malha com 513×513 pontos. Adicionalmente, Larsson *et al.* (2005) afirmam que em estudos envolvendo o algoritmo PSC (*Partial semicoarsening*), os *solvers* GS-RB e Gauss-Seidel *four-colors* foram mais robustos que GS-Lex para anisotropias moderadas, ou seja, problemas próximos do isotrópico.

Assim, o *solver* GS-RB é implementado no algoritmo FAS-FMG e seu desempenho é comparado com os resultados obtidos com o emprego do *solver* GS-Lex. Inicialmente, com procedimentos semelhantes aos adotados nas seções 5.1.1 e 5.1.2, são determinados os

parâmetros $v_{\text{ótimo}}$ e $L_{\text{ótimo}}$ para o método *multigrid* com este *solver*. Conforme pode ser observado nas Figs. 5.5 e 5.6, os parâmetros ótimos para o GS-RB são $v_{\text{ótimo}} = 2$ para a malha 512×512 , $v_{\text{ótimo}} = 1$ ou 2 para as de menor tamanho e $L_{\text{ótimo}} = L_{\text{máx}}$. Novamente, observa-se a sensibilidade do $v_{\text{ótimo}}$ com relação aos parâmetros algorítmicos. Por outro lado, o valor de $L_{\text{ótimo}}$ não sofre alteração.

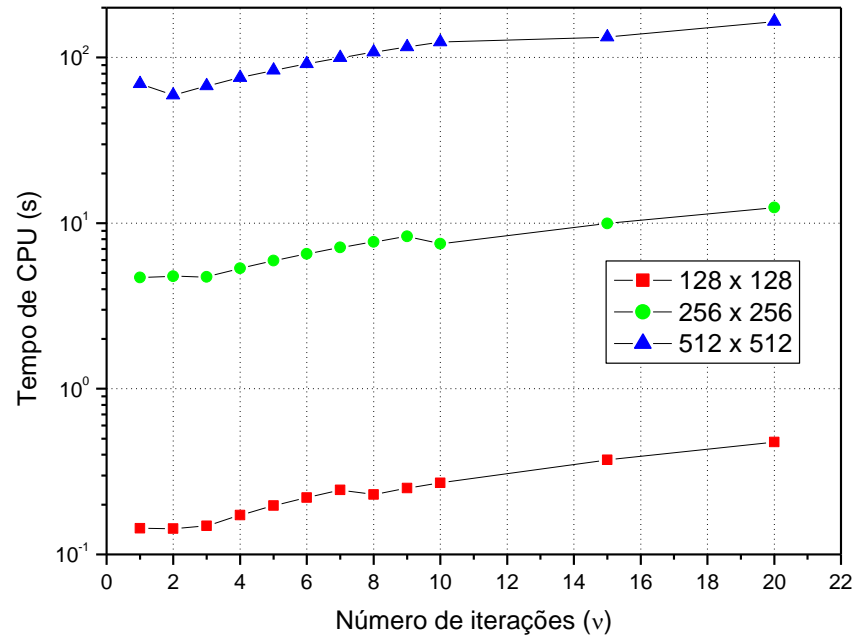


Figura 5.5: Efeito de v sobre o tempo de CPU – Burgers, *solver* GS-RB

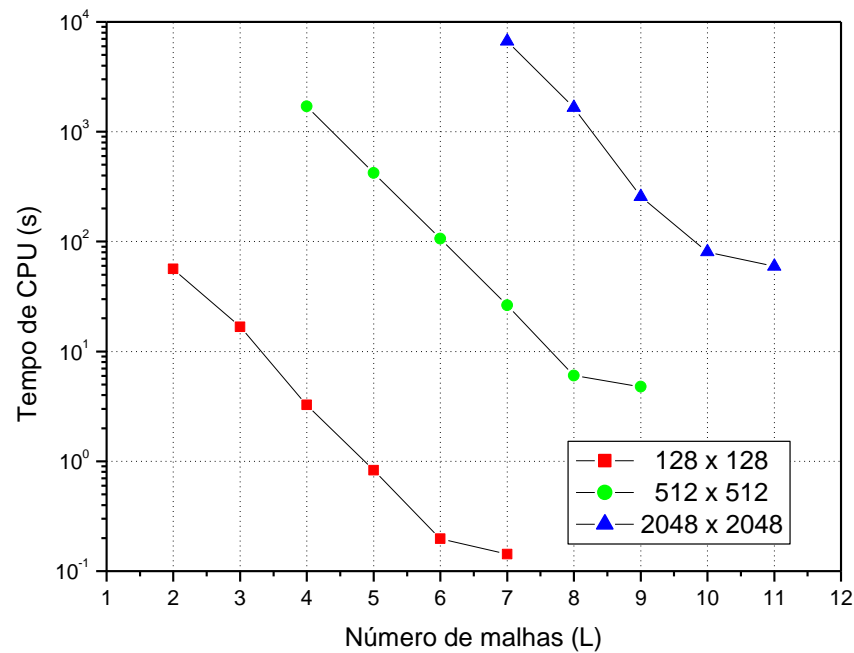


Figura 5.6: Efeito de L sobre o tempo de CPU – Burgers, *solver* GS-RB

Posteriormente, é avaliado o efeito do GS-RB sobre o tempo de CPU. A Fig.5.7 mostra um comparativo do tempo de CPU empregado na solução do problema de Burgers com os *solvers* GS-Lex e GS-RB com os métodos *singlegrid* e *multigrid*.

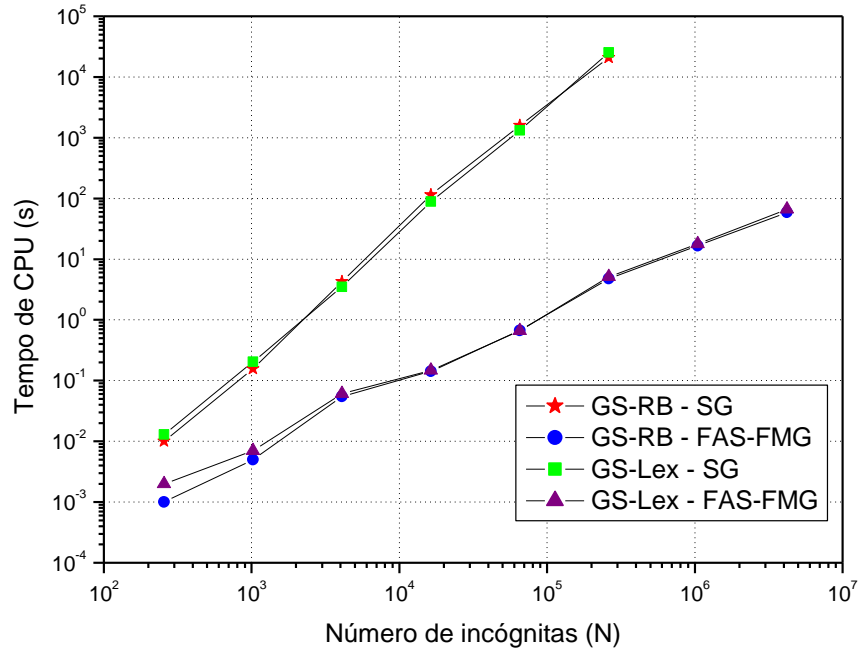


Figura 5.7: Comparativo tempo de CPU *versus* N – Burgers, *solvers* GS-Lex e GS-RB

Observando-se a Fig. 5.7, aparentemente, os valores são muito aproximados, não permitindo a análise dos resultados. A Tab. 5.4 apresenta os valores, para uma análise mais clara.

Tabela 5.4: Tempo de CPU (s) com os *solvers* GS-Lex e GS-RB

Malha	SG		FAS-FMG	
	GS-Lex	GS-RB	GS-Lex	GS-RB
16×16	0,013	0,010	0,002	0,001
32×32	0,203	0,156	0,007	0,005
64×64	3,484	4,240	0,061	0,055
128×128	88,500	114,828	0,147	0,143
256×256	1323,250	1584,281	0,664	0,672
512×512	25281,722	20793,90	5,117	4,771
1024×1024	-	-	17,750	16,656
2048×2048	-	-	62,625	59,234

Como pode ser observado na Tab. 5.4, considerando-se os esquemas FAS e FAS-FMG, os resultados do tempo de CPU com o *solver* GS-RB são ligeiramente menores do que os resultados com o GS-Lex, sendo 5,7% mais rápido na malha mais refinada. Como exceção, a malha 256×256 , onde foi mais ligeiramente mais lento. Comparando-se os resultados do SG, o GS-RB foi mais rápido nas malhas $N = 16 \times 16$ a 64×64 e, posteriormente na malha 512×512 que foi a malha mais fina calculada. Os resultados obtidos estão aquém da expectativa inicial para com GS-RB. Uma hipótese a ser considerada, na tentativa de justificar o seu mau desempenho, é que a não linearidade do problema de Burgers possa afetar a taxa de convergência do *solver*.

Os trabalhos a seguir, encontrados na literatura, apontam comportamento do GS-RB frente ao GS-Lex semelhante ao obtido neste estudo. Em um estudo de operadores de restrição para a equação de Poisson 3D, Kouatchou (1999) verificou que o GS-Lex foi de 4 a 34% mais rápido do que o GS-RB, de acordo com o problema resolvido. Como justificativa para este comportamento do *solver*, afirma que a ordenação do GS-Lex facilita o acesso aos dados da memória do computador, quando comparada à ordenação do GS-RB. Borba (2004) obteve resultados muito próximos entre si em termos de tempo de CPU, do GS-RB e do GS-Lex para o problema de Laplace e mais seis outros problemas envolvendo difusão pura ou advecção-difusão. Em todos os casos investigados nesse estudo, a redução no tempo de CPU obtida com o emprego do GS-RB foi menor do que 1%.

5.2.2 Comparativo de operadores de prolongação

Nesta seção são apresentados os resultados dos estudos acerca do efeito de alguns operadores de prolongação sobre o tempo de CPU do algoritmo FAS-FMG com os dois *solvers* testados na seção anterior.

5.2.2.1 Operadores de prolongação para a solução de cada nível do FMG

No algoritmo do FMG, a cada mudança de nível, Fig.3.16, a solução obtida é prolongada como estimativa inicial para o nível seguinte. Com o objetivo de melhorar a estimativa inicial prolongada e, conseqüentemente, reduzir o tempo de CPU do algoritmo, são testados diferentes operadores de prolongação. O operador mais comum na literatura, aplicado para este procedimento, é o operador bicúbico, de 4ª ordem (TROTTERBERG *et al.*, 2001).

Este operador é comparado com o operador bilinear, adotado como padrão, e com outros selecionados ou adaptados da literatura. São eles: o biquadrático, de 3ª ordem, o bilires, de 2ª ordem, e os operadores de 1ª ordem *licen*, *mixed* e *upwind*, todos descritos na seção 4.2.3. Como operador de prolongação para as correções é adotado o operador bilinear. Os operadores de restrição são aqueles estabelecidos como padrão na Etapa I.

Considerando-se os parâmetros ótimos estabelecidos na seção anterior para cada um dos *solvers*, $v_{ótimo} = 3$ para o GS-RB e $v_{ótimo} = 2$ para o GS-Lex, é avaliada a influência do operador de prolongação sobre o tempo de CPU.

Em alguns dos casos testados, o número ótimo de iterações se mostrou sensível à mudança do operador de prolongação. Para todos os operadores indicados na Tab. 5.5, que mostra os resultados para o tempo de CPU obtido com o *solver* GS-Lex, o valor do parâmetro é $v_{ótimo} = 3$, com exceção do *mixed* para o qual o valor é $v_{ótimo} = 2$.

Na Tab. 5.6 podem ser visualizados os resultados com o emprego do *solver* GS-RB. Para os operadores relacionados nesta tabela, os valores de $v_{ótimo}$ são $v_{ótimo} = 1$ para o biquadrático, $v_{ótimo} = 3$ para o bicúbico e para o *licen* e $v_{ótimo} = 4$ para o *mixed*. Para os demais operadores nesta tabela, o valor é $v_{ótimo} = 2$.

Tabela 5.5: Comparativo do tempo de CPU (s) e do número de ciclos V (V) para cada operador com *solver* GS-Lex.

Malha	Operador de prolongação da solução no FMG													
	bilinear		bilires		licen		biquadrático		bicúbico		<i>mixed</i>		<i>upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
32 x 32	0,007	9	0,007	9	0,007	9	0,007	9	0,007	9	0,007	11	0,008	10
64 x 64	0,061	8	0,062	8	0,068	9	0,068	9	0,068	9	0,073	11	0,075	10
128 x 128	0,147	8	0,156	8	0,165	9	0,148	8	0,148	8	0,174	11	0,173	9
256 x 256	0,664	7	0,606	7	0,751	8	0,752	8	0,752	8	0,835	10	0,770	9
512 x 512	5,117	7	5,258	7	5,117	7	5,117	7	5,125	7	6,242	10	5,898	8
1024 x 1024	17,750	6	18,281	6	20,453	7	17,734	6	17,750	6	22,703	9	23,609	8
2048 x 2048	62,625	5	64,266	5	85,344	7	62,625	5	62,703	5	96,312,	9	87,438	7

Tabela 5.6: Comparativo do tempo de CPU (s) e do número de ciclos V(V) para cada operador com *solver* GS-RB.

Malha	Operador de prolongação da solução no FMG													
	bilinear		bilires		licen		biquadrático		bicúbico		<i>mixed</i>		<i>upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
32x32	0,005	9	0,005	9	0,005	8	0,006	12	0,006	9	0,006	8	0,006	11
64x64	0,055	8	0,056	8	0,062	8	0,065	11	0,062	8	0,069	8	0,069	10
128x128	0,143	8	0,155	8	0,151	7	0,145	10	0,151	7	0,199	8	0,191	10
256x256	0,672	7	0,613	7	0,776	7	0,719	9	0,776	7	0,880	7	0,865	10
512x512	4,771	7	4,990	7	4,760	6	4,635	8	4,766	6	6,180	7	6,297	9
1024x1024	16,656	6	17,344	6	19,016	6	16,391	7	19,031	6	21,391	6	25,219	9
2048x2048	59,234	5	61,188	5	79,812	6	69,438	7	67,578	5	89,562	6	94,047	8

De um modo geral, são recomendados operadores de alta ordem para interpolar as soluções entre os níveis do FMG, sendo o operador bicúbico uma escolha comum na literatura (TROTTEBERG *et al.*, 2001; BRANDT, 2011). De acordo com os resultados das Tabs. 5.5 e 5.6, para os dois *solvers* testados, considerando-se o tempo de CPU, o operador bilinear, em geral, é o que apresenta o melhor desempenho.

Considerando-se os resultados para o GS-Lex, os operadores de ordem mais alta, biquadrático e bicúbico, e o operador de 2ª ordem bilires tiveram resultados bem próximos àqueles do operador bilinear, tanto no que diz respeito ao tempo de CPU quanto ao número de ciclos V empregados. Os operadores de 1ª ordem *licen*, *mixed* e *upwind* tiveram desempenho muito inferior, sendo de 36 a 54% mais lentos do que o operador bilinear.

O operador bilires, apesar de seu bom desempenho, reduzindo o número de ciclos V, requer mais recursos de memória, pois necessita que sejam armazenados os resíduos de cada volume de controle, os quais são incorporados na interpolação.

Os operadores biquadrático e bicúbico transferem as soluções com melhor acurácia, de um nível para outro do FMG (BRANDT, 2011). Contudo, o número de ciclos V se manteve igual àquele do operador bilinear. Além disso, estes operadores são computacionalmente mais complexos e resultam em soluções tão acuradas quanto às soluções obtidas pelo operador bilinear, conforme pode ser observado na Tab. 5.7, que mostra o erro numérico da solução de u para as malhas 128 x 128, 512 x 512 e 2048 x 2048. Os resultados do erro numérico da solução de v seguem o mesmo padrão. Neste sentido, os testes realizados não evidenciaram as vantagens do emprego destes operadores diante do emprego do operador bilinear padrão. É conveniente enfatizar que, no presente estudo, as equações de Burgers são aproximadas com

CDS de 2ª ordem. Em outros estudos, em que sejam utilizadas aproximações de alta ordem, como o CDS de 4ª ordem, os resultados podem ser diferentes.

Tabela 5.7: Norma infinita do erro numérico da solução u – GS-Lex.

Malha	bilinear	biquadrático	bicúbico
128×128	$1,7334252343060769 \times 10^{-4}$	$1,7334252343059929 \times 10^{-4}$	$1,7334252343060484 \times 10^{-4}$
512×512	$1,3257007788118337 \times 10^{-5}$	$1,3257007788119506 \times 10^{-5}$	$1,3257007811721680 \times 10^{-5}$
2048×2048	$8,9550920596308427 \times 10^{-7}$	$8,9550969695985452 \times 10^{-7}$	$8,9550969695977320 \times 10^{-7}$

Nos testes realizados com o GS-RB observa-se que o operador bilinear manteve o seu desempenho com relação aos demais operadores, ou seja, é o operador que apresenta o menor tempo de CPU, seguido do bilíres que tem desempenho próximo. Os operadores de 1ª ordem permanecem com o pior desempenho. Já os operadores de alta ordem, biquadrático e bicúbico, têm desempenho pior ao apresentado com o GS-Lex, sendo aproximadamente 16% mais lentos na malha mais refinada. De acordo com Brandt (2011), interpolações de alta ordem no FMG podem ocasionar aumento no tempo de CPU, sendo que, nestes casos, uma interpolação de ordem mais baixa pode ser a escolha mais prática. Drikakis (1998) empregou uma versão tridimensional do operador *mixed*, verificando uma redução de cerca de 30% no número de ciclos necessários para atingir o critério de convergência quando comparado ao operador trilinear⁶. Observa-se que nesse estudo, as equações são discretizadas com esquemas de 3ª e 4ª ordens.

5.2.2.2 Operadores de prolongação para as correções

Um dos principais fatores que podem melhorar o desempenho do *multigrid* é uma boa interpolação das informações entre as malhas (Darbandi *et al.* (2010)). Nesta seção são investigados os efeitos do operador de prolongação para interpolar as correções das malhas grossas para as malhas finas, no algoritmo do ciclo V, sobre o tempo de CPU. Neste estudo o operador bilinear é adotado como padrão para prolongar as soluções entre os níveis do FMG. São testados os operadores bilíres, biquadrático, *licen*, *mixed* e *upwind* para prolongar as correções, considerando-se novamente os dois *solvers* GS-Lex e GS-RB. Com relação ao número de iterações internas, o parâmetro $v_{ótimo}$ é sensível à mudança de operador, em alguns casos. Com o GS-Lex, os valores do parâmetro ótimo são $v_{ótimo} = 4$ para os operadores *mixed*

⁶ Versão tridimensional do operador linear.

e *upwind*, $v_{ótimo} = 2$ para o operador biquadrático e $v_{ótimo} = 3$ para todos os demais. Com o solver GS-RB o valor $v_{ótimo} = 2$ é padrão para todos os operadores considerados.

A Tab. 5.8 apresenta os resultados do tempo de CPU para cada um destes operadores, considerando-se o *solver* GS-Lex, com malhas de 32×32 a 2048×2048 .

Tabela 5.8: Comparativo do tempo de CPU (CPU) e do número de ciclos V (V) para cada operador de prolongação das correções com *solver* GS-Lex.

Malha	bilinear		bilires		licen		biquadrático		<i>mixed</i>		<i>upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
32×32	0,007	9	0,007	9	0,007	9	0,006	8	0,010	12	0,011	15
64×64	0,061	8	0,062	8	0,061	8	0,061	8	0,105	13	0,130	18
128×128	0,147	8	0,157	8	0,129	7	0,131	7	0,319	15	0,339	18
256×256	0,664	7	0,608	7	0,664	7	0,670	7	1,484	16	1,732	21
512×512	5,117	7	5,305	7	4,464	6	4,495	6	13,422	17	15,469	22
1024×1024	17,750	6	18,438	6	17,719	6	15,062	5	56,875	18	62,031	22
2048×2048	62,625	5	64,984	5	62,453	5	62,766	5	251,875	22	284,969	24

Considerando-se os resultados da Tab. 5.8, para o GS-Lex, os operadores bilinear, *licen* e biquadrático são os que resultam nos menores valores para o tempo de CPU. Para todos estes operadores, os resultados são bastante aproximados, sendo que o *licen* é ligeiramente mais rápido do que os demais. Acredita-se que este fato se deva à menor complexidade deste operador, em termos de operações de ponto flutuante, e que essa vantagem possa ser ainda mais evidenciada considerando-se malhas ainda mais refinadas.

Com relação aos operadores *mixed* e *upwind*, que também exigem poucas operações, os resultados são muito piores do que os demais operadores analisados. Aparentemente, estas interpolações de baixa ordem introduzem erros de interpolação maiores, os quais demandam maior quantidade de ciclos V para serem eliminados, conforme pode ser observado nos resultados da Tab. 5.8. O operador biquadrático, por sua vez, apresenta um bom desempenho frente aos demais operadores. Para esse operador, observando-se o número de ciclos V na malha mais fina, percebe-se que o ganho com uma interpolação mais acurada resulta, em geral, em menos ciclos V, compensando assim a maior quantidade de cálculos envolvidos nesta interpolação.

A seguir, os resultados do estudo de operadores com o *solver* GS-RB podem ser observados na Tab. 5.9.

Tabela 5.9: Comparativo tempo de CPU (CPU) e número de ciclos V (V) para cada operador de prolongação das correções do FMG com *solver* GS-RB.

Malha	bilinear		bilires		Licen		biquadrático		<i>mixed</i>		<i>upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
32×32	0,005	9	0,005	9	0,005	9	0,005	9	0,009	18	0,009	17
64×64	0,055	8	0,056	8	0,055	8	0,055	8	0,117	18	0,131	20
128×128	0,143	8	0,156	8	0,143	8	0,144	8	0,376	21	0,438	24
256×256	0,672	7	0,615	7	0,666	7	0,676	7	2,072	23	2,425	27
512×512	4,771	7	5,031	7	4,771	7	4,786	7	16,094	25	18,078	28
1024×1024	16,656	7	17,500	6	16,500	6	16,688	6	72,391	28	84,969	33
2048×2048	59,236	6	61,797	5	69,328	6	59,391	5	326,312	30	358,234	33

Considerando-se GS-RB, os resultados se assemelham qualitativamente aos obtidos com o GS-Lex. Ou seja, os operadores *mixed* e *upwind* resultam no pior desempenho e os demais têm desempenho semelhante entre si. Porém, neste caso, o operador *licen* não é tão eficiente quanto no caso anterior. Já o operador *bilires* se mostra mais eficiente com o GS-RB, mas os operadores *bilinear* e *biquadrático* são ainda as melhores opções, em termos de tempo de CPU com este *solver*. Liu (2010) empregou um operador baseado no resíduo (*full local*), assim como o operador *bilires*, em conjunto com o *solver* GS-Lex e verificou que o número de ciclos é reduzido em 36% com relação ao operador *bilinear*, mas não faz menção ao tempo de CPU. Observa-se que, nos testes com o *bilires*, nas malhas mais refinadas, houve redução do número de ciclos V. Contudo, essa redução não está relacionada a um algoritmo mais rápido. Neste caso, também é possível observar que o operador *biquadrático*, assim como o *bilires*, é o operador que demanda menor quantidade de ciclos V para atingir o critério de parada do problema.

De acordo com os resultados apresentados nesta seção, é possível fazer algumas considerações a respeito do efeito dos operadores de interpolação sobre o tempo de CPU:

- De acordo com o esperado, a escolha do operador de prolongação empregado para interpolar as soluções entre os níveis do FMG, ou entre as malhas grossa e fina no algoritmo do ciclo V, pode ter efeito significativo na aceleração do *multigrid*.
- De uma forma geral, o operador *bilinear* apresenta o melhor desempenho, seja na prolongação das soluções ou das correções.

- Observa-se que o efeito de cada operador sobre o tempo de CPU varia com o *solver* empregado.
- O *solver* GS-RB é mais sensível à mudança dos operadores, sendo que o número de iterações internas varia bastante de um operador para outro quando aplicados com este *solver*, particularmente no caso da prolongação das soluções.
- Para ambos os *solvers*, os operadores *upwind* e *mixed* são os que têm o pior desempenho em todos os casos, sendo este efeito mais pronunciado para o GS-RB, na prolongação das correções.
- Além do operador bilinear, que apresenta bom desempenho em todos os casos estudados, o operador biquadrático é o mais eficiente, de um modo geral.
- Para a prolongação das correções e GS-Lex, o operador *licen* é mais eficiente.
- Considerando-se a prolongação da solução no FMG, os operadores de ordem mais alta, biquadrático e bicúbico, não correspondem necessariamente à redução no tempo de CPU. Ressalta-se que o esquema de aproximação empregado no processo de discretização das equações, CDS de 2ª ordem, pode não ser o mais apropriado e, que no caso de estudos onde sejam empregados operadores de ordem mais alta, os resultados podem ser diferentes dos obtidos neste estudo.

CAPÍTULO 6

CONCLUSÃO

Este trabalho teve como principal objetivo a investigação de parâmetros do método *multigrid* geométrico para três modelos matemáticos bidimensionais do escopo da CFD, com diferentes características: a equação de Laplace (problema linear), equação de Advecção-Difusão (problema linear) e as equações de Burgers (problema não linear com duas equações). Para alcançar este objetivo foi desenvolvido um código computacional para cada problema, sendo que cada um deles pode ser resolvido por três diferentes algoritmos: SG, FAS e FAS-FMG. Em seguida, o trabalho foi desenvolvido em duas etapas.

Na primeira etapa, foi realizado um estudo preliminar em que se analisou a influência de vários parâmetros do método *multigrid* sobre o tempo de CPU necessário para resolver cada equação. Os parâmetros analisados nesta etapa foram: o número de iterações internas (v), o número de malhas (L) e o número de incógnitas (N). Além disso, foi analisado o desempenho e a eficiência dos esquemas FAS e FAS-FMG para os três problemas em estudo.

Na segunda etapa, definidos os parâmetros ótimos para as equações de Burgers, foram realizados estudos procurando identificar a melhor escolha de *solver*, entre o GS-Lex e GS-RB, bem como dos operadores de prolongação, objetivando obter as componentes algorítmicas que otimizem o desempenho do FAS-FMG.

Como resultado deste estudo, verificou-se que:

- O número ótimo de iterações internas para a equação de Laplace e equação de Advecção-Difusão foi $v_{\text{ótimo}} = 2$, e para as equações de Burgers foi $v_{\text{ótimo}} = 2$ ou 3 , com *solver* GS-RB e GS-Lex.
- O tempo de CPU pode aumentar consideravelmente, com o emprego de valores que não o $v_{\text{ótimo}}$ para o número de iterações internas. Para o problema de Laplace, o

emprego de $v = 4$, por exemplo, corresponde a um aumento de cerca de 30% no tempo de CPU.

- Para todos os problemas estudados, em todas as malhas investigadas, o valor ótimo do número de malhas é $L_{\text{ótimo}} = L_{\text{máx}} - 1$. Para $L < L_{\text{ótimo}}$, o tempo de CPU aumenta significativamente, tanto mais quanto menor for o número de malhas empregadas.
- O esquema FAS-FMG é cerca de 2 vezes mais rápido do que o esquema FAS, o que está de acordo com o resultado previsto na literatura (ZHANG *et al*, 2010; LESCHZINER, 1994).
- A ordem p do algoritmo GS-Lex e GS-RB varia entre 1,97 e 2,10 para o método *singlegrid* e entre 0,95 e 1,06 para o método FAS-FMG, dependendo do problema e do esquema. O que está de acordo com a eficiência teórica do *multigrid*.
- Para o *multigrid*, o número de equações envolvidas em cada problema, assim como a presença de não linearidades, não afeta significativamente a eficiência do FAS-FMG.
- O *solver* GS-RB foi apenas um pouco mais rápido do que o GS-Lex para o problema de Burgers, sendo apenas 5,7% mais rápido na malha 2048 x 2048.
- Os operadores de prolongação possuem efeitos significativos na aceleração do *multigrid*. Considerando-se a malha mais fina estudada e a prolongação das soluções do FMG, com *solver* GS-Lex, a diferença entre o pior e o melhor desempenho dos operadores estudados é de aproximadamente 53%.
- Para a prolongação das correções entre as malhas, os melhores resultados são obtidos com os operadores *licen* e *bilinear* (GS-Lex) e *biquadrático* e *bilinear* (GS-RB).
- Para a prolongação das soluções do FMG, considerando-se o GS-Lex, os operadores *bilinear*, *biquadrático* e *bicúbico* tiveram desempenho muito próximo, sendo a diferença entre o maior e o menor tempo de CPU inferior a 0,1 s.

- No caso do GS-RB, o operador bilinear foi o que apresentou o menor tempo de CPU.
- A melhor escolha algorítmica do *multigrid* para o problema modelado pelas equações de Burgers constitui-se do esquema FAS-FMG, com *solver* GS-RB e operador de prolongação bilinear, para interpolar as soluções entre os níveis do FMG e correções da malha grossa para a malha fina no algoritmo do ciclo V.

Contribuições:

Neste trabalho foram desenvolvidos códigos computacionais com base no método *multigrid* geométrico e realizaram-se estudos de parâmetros para importantes problemas de CFD. Assim, contribui-se com a literatura existente no sentido de que:

- As equações de Laplace, Advecção-difusão e Burgers foram resolvidas em malhas de até 2048 x 2048 volumes de controle; o tempo de CPU empregado na obtenção das soluções é, no máximo, da ordem de dezenas de segundos.
- O esforço computacional proporcional ao número de incógnitas, para o método *multigrid*, na forma do algoritmo FAS-FMG, foi confirmado para os três problemas investigados.
- Apresentou-se um código eficiente para o problema das Equações de Burgers. A eficiência foi obtida pelo estudo dos parâmetros ótimos do MG e pela seleção das melhores componentes como *solver* e operadores de interpolação.
- Verificou-se que o emprego de operadores de alta ordem, como o operador bicúbico, para interpolação da solução no esquema FAS-FMG, pode não ser a escolha mais vantajosa na redução do tempo de CPU.
- Apresentou-se um estudo sistemático de determinados parâmetros (esquemas *multigrid*, operadores de restrição, operadores de prolongação para correção e solução, *solvers*, número de iterações internas e número de níveis) para a equação de Burgers utilizando-se o método de volumes finitos.

Desdobramentos deste trabalho:

- Aplicar a técnica de aceleração de Krylov para acelerar o FMG.
- Aplicar técnicas de extrapolação para acelerar o FMG.
- Investigar o efeito de outros tipos de ciclos como os ciclos W e F, sobre o tempo de CPU.
- Investigar o efeito do *solver* Gauss-Seidel com outros esquemas, como o esquema *black-white*, sobre o tempo de CPU.
- Fazer um estudo comparativo de operadores de restrição.
- Aplicar o método *multigrid* para as equações de Navier-Stokes 2D, na formulação em variáveis primitivas e volumes finitos e investigar os parâmetros ótimos.

REFERÊNCIAS BIBLIOGRÁFICAS

AKSAN, E. N., ÖZDES, A. A numerical solution of Burger's equation. **Applied Mathematics and Computation**, v. 156, p. 395-402, 2004.

BORBA, A. A. **Métodos iterativos e *multigrid***. Dissertação de mestrado. Universidade Federal de Santa Catarina. Florianópolis, SC, 2004.

BRANDT, A. Multi-Level Adaptive Solutions to Boundary-Value Problems. **Mathematics of Computation**, v. 31, p. 333-390, 1977.

BRANDT, A. Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD. **ICASE Report**, n. 32, NASA/CR-1998-207647, 1998.

BRANDT, A., THOMAS, J. L. **Recent Advances in Efficiency for Computational Simulations**. ICASE Report, n. 16, NASA/CR-2002-11656, 2002.

BRANDT, A., LIVNE, O. **Multigrid Techniques: 1984 Guide with Applications to Applications to Fluid Dynamics**. Revised Edition. Philadelphia: Society for Industrial and Applied Mathematics, 2011.

BRIGGS, W. L., HENSON, V. E., MCCORMICK, S. F. **A *multigrid* tutorial**. 2. ed. Philadelphia: Society for Industrial and Applied Mathematics, 2000.

BURDEN, R. L., FAIRES, J. D. **Análise Numérica**. São Paulo: Pioneira Thomson Learning, 2003.

CLEARY, A. J., FALGOUT, R. D., HENSON, V. E., JONES, J. E., MANTEUFFEL, T. A., MCCORMICK, S. F., MIRANDA, G. N. E RUGE, J. W. Robustness and Scalability of Algebraic *Multigrid*. **SIAM Journal on Scientific Computing**, v. 21, n. 5, p. 1886, 2000.

DARBANDI, M., VAKILI, S., SCHNEIDER, G. Efficient multilevel restriction-prolongation expressions for hybrid finite volume element method. **International Journal of Computational Fluid Dynamics**, v. 22, n. 1, p. 29-38, 2008.

DHAWAN, S., KAPOOR, S., KUMAR, S., RAWAT, S. Contemporary review of techniques for the solution of nonlinear Burgers equation. **Journal of Computational Science**, v. 3, p. 405-419, 2012.

DRIKAKIS, D., ILIEV, O. P., VASSILEVA D. P. A Nonlinear Multigrid Method for the Three-Dimensional Incompressible Navier-Stokes Equations. **Journal of Computational Physics**, 146, p. 301-321, 1998.

EFE, M. O. Observer-based boundary control for 2D Burgers equation. **Transactions of the Institute of Measurement and Control**, v. 28, p. 177-185, 2006.

- FAURE, S.; LAMINIE, J. TEMAM, R. Finite volume discretization and multilevel methods in flow problems. **Journal of Scientific Computing**, v. 25, n. 112, p. 231-260, 2005.
- FENG, C., SHU, S., XU, J., ZHANG, C. Numerical Study of Geometric Multigrid Methods on CPU-GPU Heterogeneous Computers. **Proceedings of CoRR**, 2012.
- FERM, L.; LÖTSTEDT, P. Two-grid solution of shock problems. **SIAM J. Sci. Comput.**, v. 18, n. 6, p.1533-1552, 1997.
- FERZIGER, J. H., PERIC, M. **Computational Methods for Fluid Dynamics**. 3. ed. Berlin: Springer-Verlag, 2002.
- FORTUNA, A. O. **Técnicas Computacionais para Dinâmica de Fluidos**. São Paulo: Editora da Universidade de São Paulo, 2000.
- FOX, R. W., MCDONALD, A. T. **Introdução à Mecânica dos Fluidos**. Rio de Janeiro: Guanabara Koogan, 1995.
- GHIA, U., GHIA N., SHIN, C. T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a *Multigrid* method. **Journal of Computational Physics**, v.48, p. 387-411, 1982.
- HACKBUSCH, W. **Multi-grid Methods and Applications**. Berlin: Springer-Verlag, 1985.
- HIRSCH, C. **Numerical computational of internal and external flows**. Chichester: Wiley, 1988.
- HON, Y. C., MAO, X. Z. An efficient numerical scheme for Burger's equation. **Applied Mathematics and Computation**, v. 95, p.37-50, 1998.
- HORTMANN, M., PERIC, M., SCHEURER, G. Finite Volume Multigrid Prediction of Laminar Natural Convection: Bench-March solutions. **International Journal for Numerical Methods in Fluids**, v. 11, p. 189-207, 1990.
- IBRAHEEM, S. O.; DEMUREN, A. O. On Bi-Grid Local Mode Analysis of Solution Techniques for 3-D Euler and Navier-Stokes Equations. **Journal of Computational Physics**, v. 125, p.354-377, 1996.
- JIMAC, P. K. Applications of Multigrid Techniques in CFD. **International Journal for Numerical Methods in Fluids**, v. 01, p.1-12, 2007.
- KANNAN, R., WANG, Z. J. A high order spectral volume solution on the Burgers' equation using Hopf Cole transformation. **International Journal for Numerical Methods in Fluids**, v.69, p. 781-801, 2012.
- KOUATCHOU, J., ZHANG, J. Optimal injection operator in multigrid solution of the three dimensional Poisson equation. **International Journal of Computer Mathematics**, v. 76, p. 173-190, 2000.

KUMAR, D. S., KUMAR, K.S., DAS, M. K. A Fine Grid Solution for a Lid-Driven Cavity Flow Using *Multigrid* Method. **Engineering Applications of Computational Fluid Mechanics**, v.3, n. 3, p. 336-354, 2009.

KUNDU, P. K., **Fluid Mechanics**. San Diego: Academic Press, 1990.

LIAO, W. A fourth-order finite-difference method for solving the system of two-dimensional Burgers' equations. **International Journal for Numerical Methods in Fluids**, v.64, p.565-590, 2010.

LIEN, F. S., LESCHZINER, M. A. Multigrid acceleration for recirculating laminar and turbulent flows computed with a non-orthogonal, collocated finite volume scheme. **Computer Methods in Applied Mechanics and Engineering**, v. 118, p. 351-371, 1994.

LILEK, Z., MUZAFERIJA, S., PERIC, M. Efficiency and accuracy aspects of a full-multigrid simple algorithm for three-dimensional flows. **Numerical Heat Transfer, Part B**, v.31, p. 23-42, 1997.

LIU, F.; SHI, W. Numerical solutions of two-dimensional Burgers' equations by lattice Boltzmann method. **Commun Nonlinear Sci Numer Simulat**, v. 16, p. 150-157, 2011.

LIU, Z. Optimal multigrid methods with new transfer operators based on finite difference approximations. **Acta Appl. Math.**, v. 111, p.83-91, 2010.

LIU, Z. Multigrid method with a new interpolation operator. **International Journal of Computer Mathematics**, v. 88, n. 5, pp.982-993, 2011.

MALISKA, C. R. **Transferência de Calor e Mecânica dos Fluidos Computacional**. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004.

MARCHI, Carlos H. **Protocolo para Testes de Coerência: Versão 1.0**. Disponível em: <ftp://ftp.demec.ufpr.br/CFD/bibliografia/erros_numericos/>, acesso em 18 de janeiro de 2013.

MARTINS, M. A. **Estimativa de Erros de Iteração em Dinâmica dos Fluidos Computacional**. Dissertação de Mestrado. Universidade Federal do Paraná, Curitiba, PR, 2002.

MESQUITA, M. S., DE LEMOS, M. J. S. Optimal Multigrid Solutions of Two-dimensional Convection-conduction Problems. **Applied Mathematics and Computation**, v. 152, p. 725-742, 2004.

MONTERO, R. S., LLORENTE, I. M., SALAS, M. D. Robust Multigrid Algorithm for Navier-Stokes Equations. **Journal of Computational Physics**, v. 173, p.412-432, 2001.

MULDER, W. A. Multigrid for one-dimensional inviscid Burgers equation. **SIAM J. Sci. Stat. Comput.**, v.11, p.33-50, 1990.

NEVEU, E.; DEBREU, L.; LE DIMET F.X. Multigrid Methods and data assimilation. **ARIMA**, v. 14, p.63-80, 2011.

OLIVEIRA, F., PINTO, M. A. V., SANTIAGO, C. D., MARCHI, C. H. Efeito de Parâmetros do Método *Multigrid* CS e FAS sobre o Tempo de CPU em Problemas 1D lineares e não-lineares. **Proceedings of XXVII CILAMCE**, Iberian Latin American Congress on Computational Methods in Engineering. Belém, 2006.

OLIVEIRA, F., PINTO, M. A. V., SANTIAGO, C. D., MARCHI, C. H. Efeito de Roteiros do Método Multigrid sobre o Tempo de CPU para a Equação de Laplace 2D. **Proceedings of XXIX CILAMCE**, Iberian Latin American Congress on Computational Methods in Engineering. Maceió, 2008.

OLIVEIRA, F., PINTO, M. A. V., MARCHI, C. H., ARAKI, L. K. Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. **Applied Mathematical Modelling**, v. 36, p. 4665–4676, 2012.

PATANKAR, S. V. **Numerical Heat Transfer and Fluid Flow**. Washington DC: Hemisphere, 1980.

PEREIRA, F. H., VERARDI, S. L. L., NABETA, S. I. A fast algebraic *multigrid* preconditioned conjugate gradient solver. **Applied Mathematics and Computational**, v. 179, p. 344-351, 2006.

PERSSON, I., SAMUELSSON, K., SZEPESY, A. On the convergence of multigrid methods for flow problems. **Electronic Transactions on Numerical Analysis**, v. 8, p.46-87, 1999.

PINTO, M. A. V., SANTIAGO, C. D., MARCHI, C. H. Effect of Parameters of a Multigrid Method on the CPU Time for One-dimensional Problems. **Proceedings of COBEM 2005**, Ouro Preto, 2005.

PINTO, M. A. V., SANTIAGO, C. D., MARCHI, C. H. Efeito de Parâmetros do Método *Multigrid* sobre o Tempo de CPU para a Equação de Burgers Unidimensional. **Proceedings of XXVI CILAMCE**, Iberian Latin American Congresson Computational Methods in Engineering, Guarapari, 2005.

PINTO, M. A. V. **Comportamento do Método *Multigrid* Geométrico em problemas de Transferência de Calor**. Tese de doutorado. Universidade Federal do Paraná, Curitiba, PR, 2006.

PINTO, M. A. V., MARCHI, C. H. Optimum parameters of a geometric multigrid for the two-dimensional Laplace's Equations. **Proceedings of COBEM 2007**, Brasília, 2007.

RABI, J. A., DE LEMOS, M. J. S. Optimization of Convergence Acceleration in Multigrid Numerical Solutions of Conductive-convective problems. **Applied Mathematics and Computation**, v. 124, p. 215-226, 2001.

ROACHE, P. J. **Verification and Validation in Computational Science and Engineering**. Albuquerque, USA: Hermosa, 1998.

ROACHE, P. J. Code verification by the method of manufactured solutions, **J. Fluids Eng.**, v. 124 (1), p. 4-10, 2002.

ROY, C. J. Review of Code and Solution Verification Procedures for Computational Simulation. **Journal of Computational Physics**, v. 205, p. 131-156, 2005.

SANTIAGO, C. D., MARCHI, C. H. Optimum Parameters of a Geometric Multigrid for a Two-Dimensional Problem of Two-Equations. **Proceedings of COBEM 2007**, Brasília, 2007.

SANTIAGO, C. D., MARCHI, C. H. Parâmetros Ótimos do Método *Multigrid* Geométrico CS e FAS para Problemas 2D com duas Equações. **Proceedings of XXIX CILAMCE**, Iberian Latin American Congress on Computational Methods in Engineering, Maceió, 2008.

SANTIAGO, C. **Estudo de parâmetros do *Multigrid* para sistemas de equações 2D em CFD**. Tese de doutorado. Universidade Federal do Paraná, Curitiba, PR, 2010.

SANTIAGO, C. D., MARCHI, C. H., SOUZA, L. F., ARAKI, L. K. Performance of multigrid method whit alternative formulations for the Navier-Stokes equations. **Proceeding of COBEM 2011**, Natal, 2011.

SCHNEIDER, F. A. **Verificação de soluções numéricas em problemas difusivos e advectivos com malhas não-uniformes**. Tese de doutorado. Universidade Federal do Paraná, Curitiba, PR, 2007.

SCHNEIDER, G. E., ZEDAN, M. A modifiedStronglyImplicit Procedure for Numerical Solution of Field Problems. **Numerical Heat Transfer**, v. 4, p. 1-19, 1981.

SHAMARDAN, A. B., ABO ESSA, Y. M. Multi-level adaptative solutions to initial value problems. **Korean J. Comput.& Appl. Math.**, vol. 7. n. 1, p. 215-222, 2000.

SHIH, T. M., TAN, C. H., HWANG, B. C. Effects of grid staggering on numerical scheme. **International Journal for Numerical Methods in Fluids**, v. 9, p. 193-212, 1989.

SIRAJ-UL-ISLAM; SARLER, B.; VERTNIK, R.; KOSEC, G. Radial basis function collocation method for the numerical solution of the two-dimensional transient nonlinear coupled Burgers' equations. **Applied Mathematical Modelling**, v. 36, p.1148-1160, 2012.

SOUZA, L. F., OISHI, C. M., CUMINATO, J. A., NETO, A. S. Comparsion of Poisson solvers in a transient 2D fluid flow problem with Dirichlet boundary conditions. **Proceedings of ENCIT 2006**, Curitiba, 2006.

SUERO, R., PINTO, M. A. V., MARCHI, C. H., ARAKI, L. K., ALVES, A. C. Analysis of algebraic multigrid parameters for two-dimensional steady-state heat diffusion equations. **Applied Mathematical Modelling**, v. 36 p. 2996–3006, 2012.

STÜBEN, K. A review of algebraic *multigrid*. **Journal of Computational and Applied Mathematics**, v. 128, p. 281-309, 2001.

TANNEHILL, J. C., ANDERSON, D. A., PLETCHER, R. H. **Computational Fluid Mechanics and Heat Transfer**. Florence: Taylor & Francis, 1997.

THEKALE, A., GRADL, T., KLAMROTH, K., RÜDE U. Optimizing the number of multigrid cycles in full multigrid algorithm. **Numerical Linear Algebra whit Applications**, v. 17, p. 199-210, 2010.

TROTTEBERG, U., OOSTERLEE, C., SCHÜLLER, A. **Multigrid**. San Diego: Academic Press, 2001.

VERSTEEG, H. K., MALALASEKERA, W. **An introduction to Computational Fluid Dynamics The Finite Volume Method**. London: Pearson Longman, 2007.

XU, P., SUN, Z. A second-order accurate difference scheme for the two-dimensional Burger's System. **Numerical Methods for Partial Differential Equations**, v. 25, p.172-194, 2009.

YAN, J., THIELE, F. Performance and Accuracy of a Modified Multigrid Algorithm for Fluid Flow and Heat Transfer, **Numerical Heat Transfer**, Part B, v.34, p.323-338, 1998.

YAN, J., THIELE, F., XUE, L. A Modified Full Multigrid for the Navier-Stokes Equations. **Computers & Fluids**, v.34, p.445-454, 2007.

WESSELING, P. **An introduction to Multigrid Methods**. New York: John Wiley & Sons, 1992.

WALLIS, J. T. Methods toward better multigrid solver convergence. **Applied Mathematics Research Report**, Department of Mathematical Sciences, KAIST, Korea, June 1, 2008.

WANG, Y., ZHANG, J. Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation. **Journal of Computational Physics**, 228, p.137-146, 2009.

WESSELING, P., OOSTERLEE, C. W. Geometric Multigrid whit Applications to Computational Fluid Dynamics. **Journal of Computational and Applied Mathematics**, v. 128, p.311-334, 2001.

ZHANG, J. **Multigrid acceleration technics and applications to the numerical solution of partial differential equations**. Dissertação de mestrado. The George Whashington University, Whashington, DC, 1997.

ZHANG, W.; ZHANG, C. H.; XI, G. An explicit Chebyshev pseudospectral multigrid method for incompressible Navier-Stokes equations. **Computers & Fluids**, v. 39, pp.178-188, 2010.

ZHAO, G.; YU, X.; ZHANG, R. The new numerical method for solving the system of two-dimensional Burgers' equations. **Computers and Mathematics whit Applications**, v. 62, p. 3279-3291, 2011.

APÊNDICE A

VERIFICAÇÃO DOS CÓDIGOS

A.1 Verificação e validação

A validação e a verificação consistem em processos que permitem quantificar o nível de confiabilidade das soluções numéricas. A validação relaciona-se com o grau de fidelidade com que um determinado modelo representa um dado fenômeno físico. Nesta análise, os valores obtidos são comparados sistematicamente com resultados experimentais (FORTUNA, 2000). A verificação, por sua vez, relaciona-se com o grau de correção de um modelo implementado, isto é, deve-se confirmar que a implementação de um modelo (representada por equações, parâmetros e métodos numéricos adotados) representa sua descrição conceitual (VERSTEEG e MALALASEKERA, 2007).

O processo de verificação se distingue em verificação do código e verificação da solução. A verificação do código constitui-se na asserção, tanto quanto possível, de que não existem erros ou inconsistências no algoritmo implementado. A verificação da solução, por outro lado, consiste no processo de quantificação dos erros decorrentes de toda e qualquer simulação numérica (ARAKI, 2007).

Consistência e estabilidade são condições necessárias e suficientes para a convergência. Um método é consistente se o erro de truncamento tende a zero com o refinamento da malha, o que significa que a equação discretizada deve tender à equação original quando o espaçamento entre os pontos da malha tende a zero. A estabilidade em um método numérico é obtida quando o erro numérico tende a um valor finito quando o número de iterações tende ao infinito. A solução numérica é convergente quando é estável e tende para a solução das equações diferenciais quando a malha é refinada (MALISKA, 2004).

O erro numérico independe dos resultados experimentais, sendo avaliado, entretanto, somente quando a solução analítica do problema é conhecida. Deste modo, para a grande maioria dos problemas práticos, cuja solução analítica é desconhecida, o erro numérico não pode ser obtido. Neste caso, deve-se estimar o valor da solução analítica, calculando-se então

a incerteza (U) da solução numérica. A incerteza de uma solução numérica é calculada utilizando-se estimadores de erro, como os estimadores de Richardson e GCI (ROACHE, 1998).

No caso em que os erros de programação, de arredondamento e de iteração, descritos no Capítulo 2, são controlados a ponto de serem desprezíveis, observa-se que o erro da solução numérica obtida é causado pelos erros de truncamento. Neste caso, o erro numérico recebe a denominação de erro de discretização (FERZIGER e PERIĆ, 2001), podendo ser expresso como

$$E(\phi) = b_1 h^{p_L} + b_2 h^{p_2} + b_3 h^{p_3} + \dots, \quad (\text{A.1})$$

que é conhecida como equação geral do erro de discretização, na qual: os b_i são coeficientes independentes do espaçamento da malha (h), mas que são funções da variável dependente, e os p_v (isto é, $p_L, p_2, p_3 \dots$) são as ordens verdadeiras do erro. Por definição, as ordens verdadeiras (p_v) correspondem aos expoentes de h dos termos não nulos. O menor dos expoentes de h na equação geral do erro de truncamento é denominado ordem assintótica (p_L), que deve ser um número inteiro positivo, satisfazendo a condição $p_L \geq 1$.

As estimativas de erro de discretização podem ser classificadas como estimativas “*a priori*” e estimativas “*a posteriori*”. Enquanto as estimativas de erro “*a priori*” são utilizadas para estimar a ordem do erro de discretização, as estimativas de erro “*a posteriori*” são usadas efetivamente para avaliar a magnitude do erro de discretização. As estimativas de erro “*a posteriori*” podem ser divididas em dois grandes grupos (MARCHI, 2001): o primeiro, na qual a estimativa de erro é avaliada a partir da solução numérica em uma única malha (usualmente utilizada para o método de elementos finitos), e o segundo, na qual as estimativas de erros são realizadas utilizando-se resultados de várias malhas (normalmente utilizado para os métodos de diferenças e de volumes finitos).

As ordens efetiva e aparente do erro permitem verificar “*a posteriori*” se a ordem assintótica dos erros de discretização é atingida, uma vez que esta se trata de um resultado teórico, calculado “*a priori*” das soluções numéricas. A ordem efetiva (p_E) é definida por Marchi (2001) como a inclinação local da curva do erro de discretização E_h da solução numérica da variável de interesse (ϕ) em função do tamanho (h) dos elementos da malha em um gráfico logarítmico. Se a solução analítica Φ do problema é conhecida, pode-se determinar a ordem efetiva do erro utilizando-se a solução analítica e duas soluções

numéricas. Para duas malhas distintas, com elementos h_1 na malha fina e h_2 na malha grossa, a ordem efetiva do erro é dada por

$$p_E = \frac{\log\left(\frac{\Phi - \phi_2}{\Phi - \phi_1}\right)}{\log(q)} \quad (\text{A.2})$$

onde ϕ_1 e ϕ_2 são soluções numéricas de duas malhas, q é a razão de refino entre as malhas ($q = h_2/h_1$).

Se a solução analítica não é conhecida, a ordem assintótica do erro de truncamento é verificada através da ordem aparente (p_U), definida por Marchi (2001) como

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(q)} \quad (\text{A.3})$$

onde ϕ_1, ϕ_2 e ϕ_3 representam três soluções numéricas em três malhas distintas com elementos de tamanho h_1, h_2 e h_3 e razão de refino $q = h_2/h_1 = h_3/h_2$.

A.2 Verificação dos códigos

A verificação numérica dos códigos computacionais desenvolvidos neste trabalho foi realizada mediante a análise do comportamento do erro de discretização de algumas variáveis de interesse, selecionadas para cada problema. Para os problemas de Laplace e Advecção-Difusão, consideram-se como variáveis de interesse, a norma (infinito) do erro na solução T e a temperatura média, calculada pela regra do retângulo. Para o problema de Burgers, consideram-se as normas (infinito) das soluções u e v e a força da tampa sobre o fluido. Todos os problemas estudados têm solução analítica, cujas equações e respectivas condições de contorno foram especificadas no capítulo 4 deste trabalho.

Nesta verificação, foram consideradas as ordens efetiva e aparente do erro de discretização, o comportamento do erro de discretização mediante o refino da malha computacional e a ordem de convergência do método.

A.3 Erro de discretização

Nesta seção, apresenta-se um estudo do erro de discretização obtido nas soluções numéricas das equações de Laplace, Advecção-difusão e Burgers. São analisadas as ordens efetiva e aparente bem como seu comportamento mediante o refino da malha. Foram determinados os resultados analíticos e numéricos obtidos para cada variável de interesse e calculadas as respectivas ordens efetiva e aparente. Para minimizar o erro de iteração, cada programa foi executado até atingir o erro de máquina.

Esta análise busca verificar, para cada variável de interesse, se as ordens efetiva e aparente do erro de discretização tendem às respectivas ordens assintóticas com o refino da malha. A ordem assintótica do erro de discretização para problemas com aproximações UDS é $p_L = 1$ e com aproximações CDS é $p_L = 2$. Para os problemas considerados, os termos difusivos foram discretizados mediante o CDS. Já os termos advectivos foram discretizados mediante o CDS, com correção adiada sobre o UDS, de forma que a ordem assintótica do erro de discretização também é $p_L = 2$. As Figs. A.1 – A6 mostram o comportamento das ordens efetiva e aparente dos erros, mediante o refino da malha.

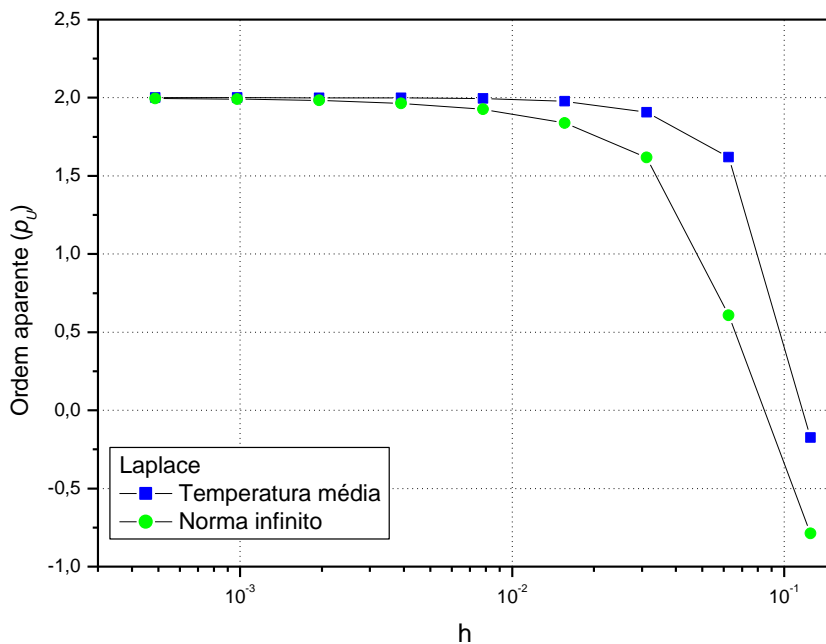


Figura A.1: Ordem aparente (P_U) em função do espaçamento da malha h – Laplace.

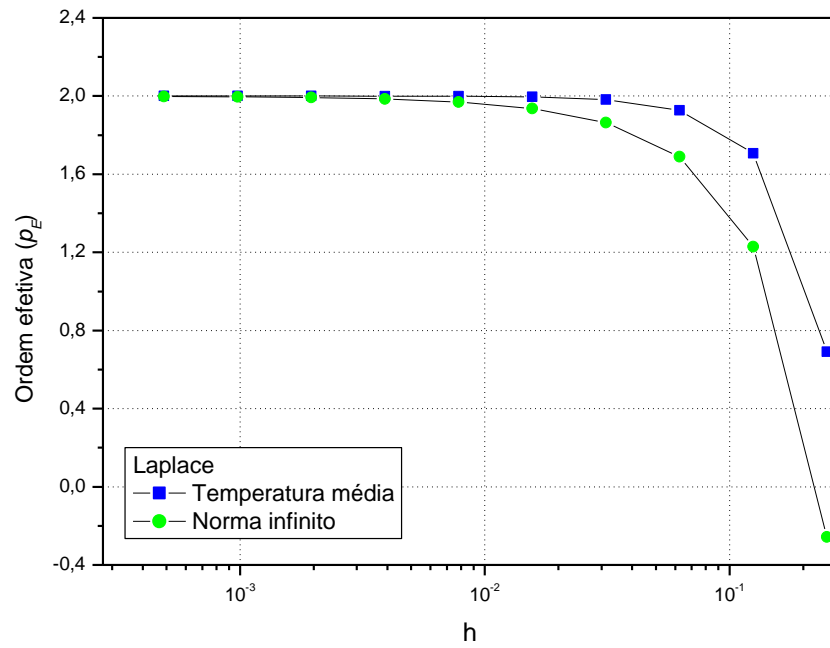


Figura A.2: Ordem efetiva (P_E) em função do espaçamento da malha h – Laplace.

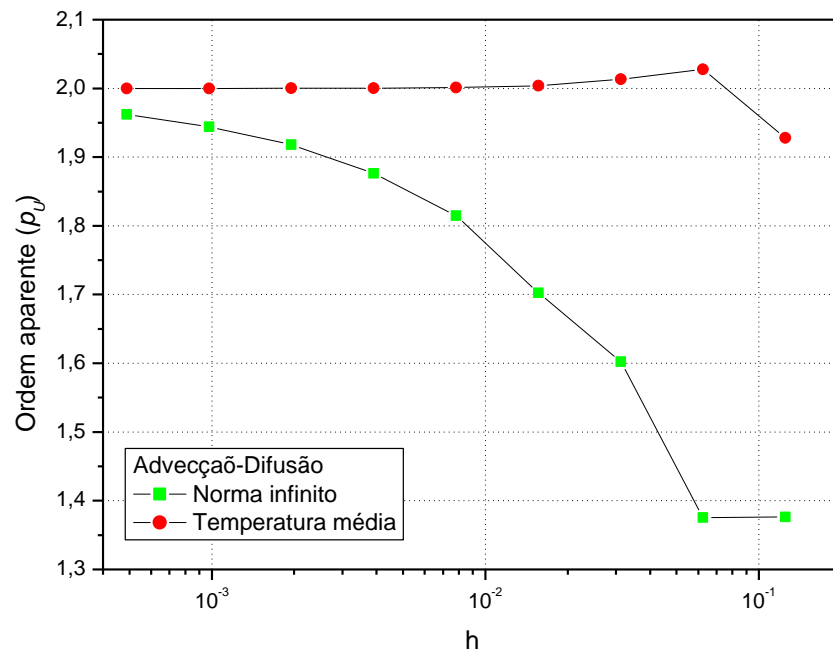


Figura A.3: Ordem aparente (P_U) em função do espaçamento da malha h – Advecção-Difusão.

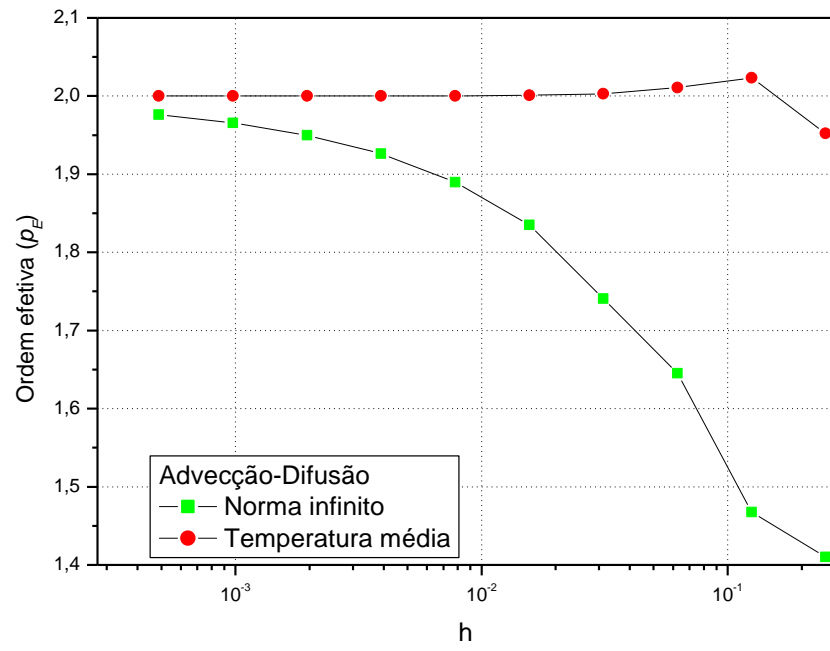


Figura A.4: Ordem efetiva (P_E) em função do espaçamento da malha h – Advecção-Difusão.

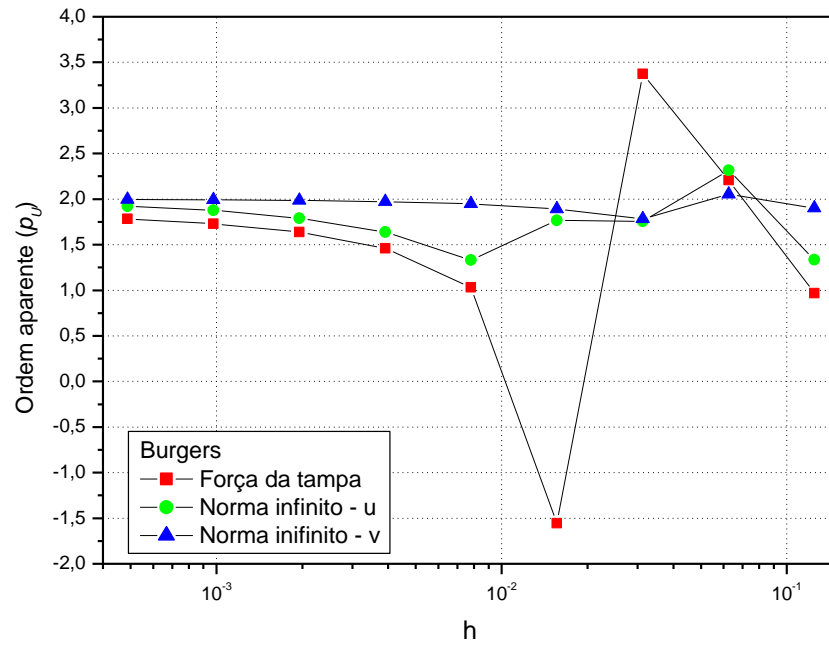


Figura A.5: Ordem aparente (P_U) em função do espaçamento da malha h – Burgers.

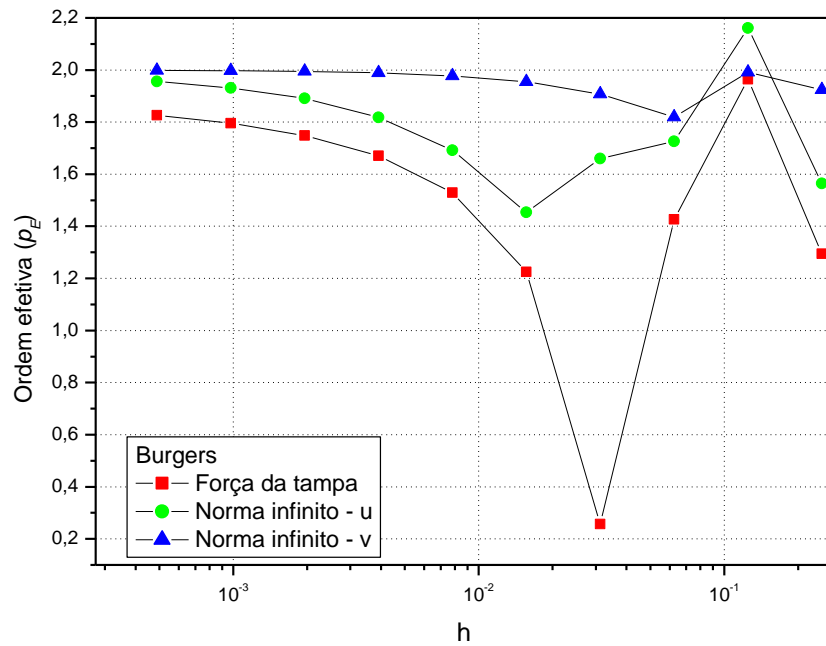


Figura A.6: Ordem efetiva (P_E) em função do espaçamento da malha h – Burgers.

Como pode ser observado nas Figs. A.1 – A6, as ordens de todas as variáveis de interesse referentes aos três problemas estudados tendem ao valor assintótico, mediante o refino da malha. Em todos os casos, as ordens efetiva e aparente do erro de discretização tendem à sua ordem assintótica $p_L = 2$.

Uma segunda análise na verificação da consistência das soluções numéricas consta da observação do comportamento do erro de discretização mediante o refino da malha. Neste caso, espera-se que o erro tenda a zero quando o espaçamento (h) da malha tende a zero, o que significa que o erro de discretização deve diminuir com o aumento do número de incógnitas. As Figs. A7- A9 mostram o comportamento do erro de discretização, para todas as variáveis de interesse, relativas aos três problemas estudados.

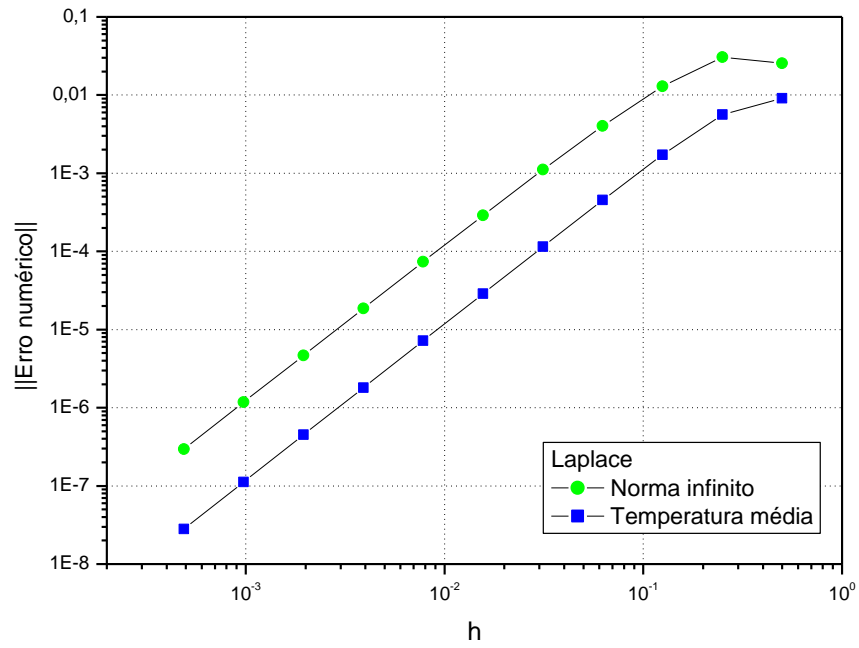


Figura A.7: Erro numérico *versus* espaçamento (h) da malha - Laplace

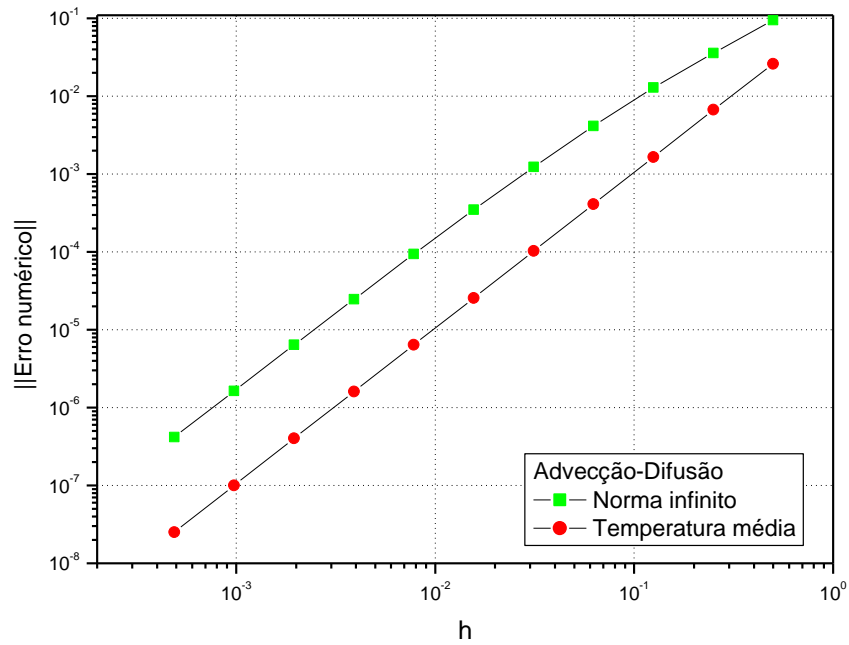


Figura A.8: Erro numérico *versus* espaçamento da malha (h) – Advecção-Difusão.

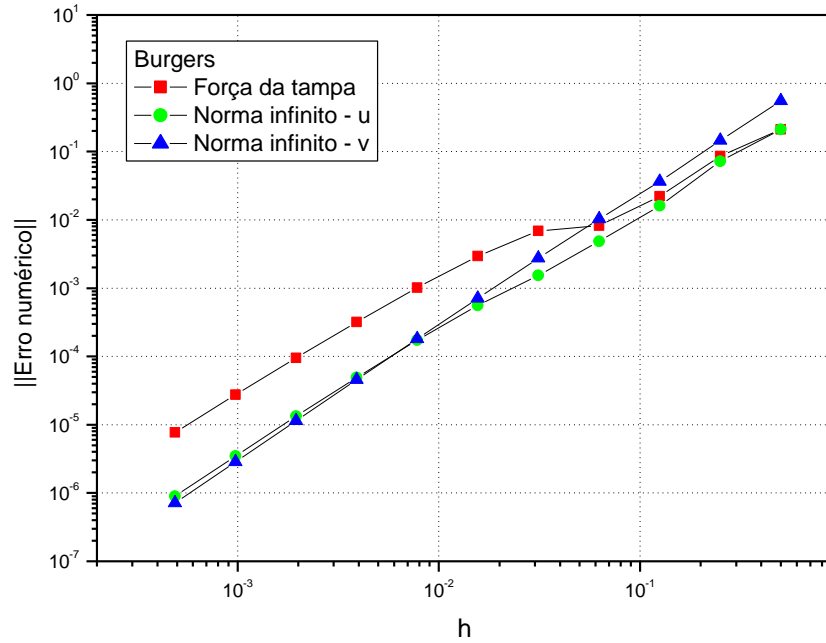


Figura A.9: Erro numérico *versus* espaçamento (h) da malha – Burgers.

De acordo com os resultados analisados, ilustrados nas Figs. A7-A9, o comportamento do erro de discretização está de acordo com o esperado. Isto é, o erro de discretização diminui com o refino da malha. Isto ocorre para todas as variáveis analisadas, correspondentes aos três problemas estudados.

A.4 Ordem de convergência dos métodos SG e MG

Para verificar a ordem de convergência do SG e do MG e avaliar o comportamento do tempo de CPU em função do número de incógnitas, para cada curva da Fig. A.10, a seguir, foi feito o ajuste tempo de $CPU(N) = cN^p$, onde p representa a ordem de convergência, ou a inclinação da curva e c é uma constante que depende do método. De acordo com a teoria, os valores esperados para o *singlegrid* e para o *multigrid* são $p = 2$ e $p = 1$, respectivamente.

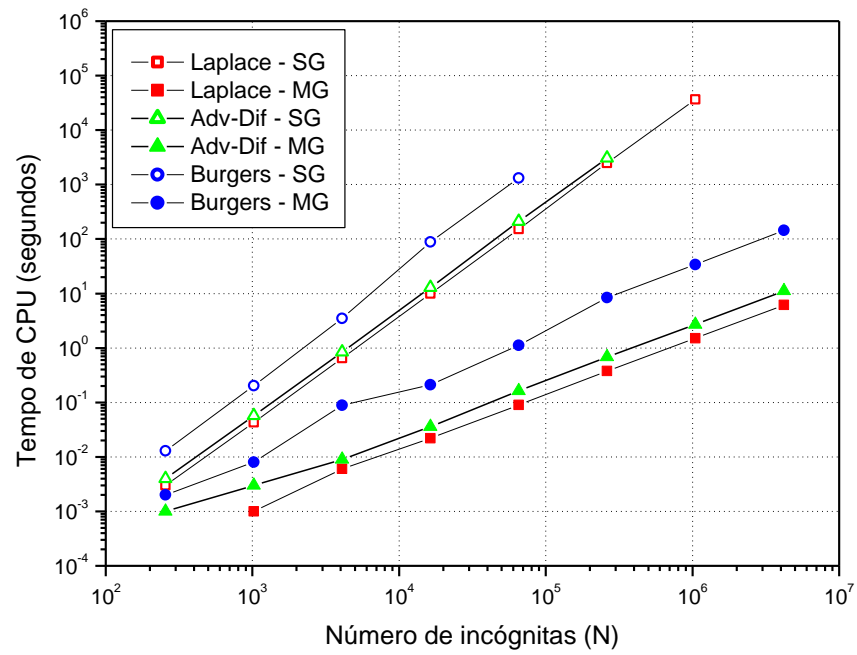


Figura A.10: Tempo de CPU *versus* número de incógnitas para o MG (FAS) e SG.

Os coeficientes c e p obtidos do ajuste de curva, para cada problema e cada método são apresentados na Tab. A.1. Como pode ser observado nesta tabela, os valores encontrados para o coeficiente p estão de acordo com os valores teóricos, para os três problemas analisados e para os dois métodos implementados. Estes resultados permitem concluir que os métodos *multigrid* e *singlegrid* foram implementados corretamente.

Tabela A.1: Coeficientes c e p para os métodos SG e MG.

Problema	SG		MG	
	c	p	c	p
Laplace	$5,26 \times 10^{-8}$	1,97	$9,40 \times 10^{-7}$	1,03
Advecção-Difusão	$7,30 \times 10^{-8}$	1,96	$2,83 \times 10^{-6}$	0,97
Burgers	$1,03 \times 10^{-7}$	2,10	$3,30 \times 10^{-6}$	1,16

APÊNDICE B

OPERADOR BICÚBICO

B.1 Equações do operador bicúbico

O operador de prolongação bicúbico é obtido de forma análoga ao operador biquadrático, ou seja, interpolando-se polinômios cúbicos nas direções x e y . Alguns passos da obtenção das expressões para este operador são descritos a seguir.

Para determinar um polinômio que passe pelos pontos A, B, C e D na Fig. B.1, consideram-se os polinômios de Lagrange (BURDEN e FAIRES, 2003). Com este polinômio interpola-se o ponto AB, em azul na Fig. B.1.

Este procedimento também é realizado para os três conjuntos de pontos: E, F, G e H; I, J, K e L e M, N, O e P, resultando em seus respectivos polinômios interpoladores. Com estas equações, são interpolados os pontos EF, IJ e MN, respectivamente, indicados na Fig. B.1.

Com procedimento análogo àquele considerado na obtenção do polinômio que passa pelos pontos A, B, C e D, obtêm-se o polinômio que passa pelos pontos AB, EF, IJ e MN. E, finalmente, pode-se obter o valor interpolado para o ponto 1 da malha fina (indicado na Fig. B.1), em termos de seus respectivos valores nos pontos A, B, C, D, E, F, G, H, I, J, K, L, M, N, O e P da malha grossa.

Os demais pontos da malha fina (2 a 36) são interpolados de maneira semelhante. A seguir, são apresentadas suas expressões.

$$\begin{aligned}
\phi_5^h &= \frac{1}{16384} (385\phi_A^{2h} - 2079\phi_B^{2h} + 10395\phi_C^{2h} + 1155\phi_D^{2h} + 385\phi_E^{2h} - 2079\phi_F^{2h} \\
&\quad + 10395\phi_G^{2h} + 1155\phi_H^{2h} - 165\phi_I^{2h} + 891\phi_J^{2h} - 4455\phi_K^{2h} - 495\phi_L^{2h} \\
&\quad + 35\phi_M^{2h} - 189\phi_N^{2h} + 945\phi_O^{2h} + 105\phi_P^{2h}) \\
\phi_6^h &= \frac{1}{16384} (539\phi_A^{2h} - 2541\phi_B^{2h} + 5929\phi_C^{2h} + 5929\phi_D^{2h} + 539\phi_E^{2h} - 2541\phi_F^{2h} \\
&\quad + 5929\phi_G^{2h} + 5929\phi_H^{2h} - 231\phi_I^{2h} + 1089\phi_J^{2h} + 10395\phi_K^{2h} + 1155\phi_L^{2h} \\
&\quad + 385\phi_M^{2h} - 2079\phi_N^{2h} + 10395\phi_O^{2h} + 1155\phi_P^{2h}) \\
\phi_7^h &= \frac{1}{16384} (1155\phi_A^{2h} + 1155\phi_B^{2h} - 495\phi_C^{2h} + 105\phi_D^{2h} + 10395\phi_E^{2h} + 10395\phi_F^{2h} \\
&\quad - 4455\phi_G^{2h} + 945\phi_H^{2h} - 2079\phi_I^{2h} - 2079\phi_J^{2h} + 891\phi_K^{2h} - 189\phi_L^{2h} \\
&\quad + 385\phi_M^{2h} + 385\phi_N^{2h} - 165\phi_O^{2h} + 35\phi_P^{2h}) \\
\phi_8^h &= \frac{1}{16384} (225\phi_A^{2h} + 2025\phi_B^{2h} - 405\phi_C^{2h} + 75\phi_D^{2h} + 2025\phi_E^{2h} + 18225\phi_F^{2h} \\
&\quad - 3465\phi_G^{2h} + 675\phi_H^{2h} - 405\phi_I^{2h} - 3645\phi_J^{2h} + 729\phi_K^{2h} - 135\phi_L^{2h} \\
&\quad + 75\phi_M^{2h} + 675\phi_N^{2h} - 135\phi_O^{2h} + 25\phi_P^{2h}) \\
\phi_9^h &= \frac{1}{16384} (-105\phi_A^{2h} + 1575\phi_B^{2h} + 525\phi_C^{2h} - 75\phi_D^{2h} - 945\phi_E^{2h} + 14175\phi_F^{2h} \\
&\quad + 4725\phi_G^{2h} - 675\phi_H^{2h} + 189\phi_I^{2h} - 2835\phi_J^{2h} - 945\phi_K^{2h} + 135\phi_L^{2h} \\
&\quad - 35\phi_M^{2h} + 525\phi_N^{2h} + 175\phi_O^{2h} - 25\phi_P^{2h}) \\
\phi_{10}^h &= \frac{1}{16384} (-75\phi_A^{2h} + 525\phi_B^{2h} + 1575\phi_C^{2h} - 105\phi_D^{2h} - 675\phi_E^{2h} + 4725\phi_F^{2h} \\
&\quad + 14175\phi_G^{2h} - 945\phi_H^{2h} + 135\phi_I^{2h} - 945\phi_J^{2h} - 2835\phi_K^{2h} + 189\phi_L^{2h} \\
&\quad - 25\phi_M^{2h} + 175\phi_N^{2h} + 525\phi_O^{2h} - 35\phi_P^{2h}) \\
\phi_{11}^h &= \frac{1}{16384} (75\phi_A^{2h} - 405\phi_B^{2h} + 2025\phi_C^{2h} + 225\phi_D^{2h} + 675\phi_E^{2h} - 3465\phi_F^{2h} \\
&\quad + 18225\phi_G^{2h} + 2025\phi_H^{2h} - 135\phi_I^{2h} + 729\phi_J^{2h} - 3645\phi_K^{2h} - 405\phi_L^{2h} \\
&\quad + 25\phi_M^{2h} - 135\phi_N^{2h} + 675\phi_O^{2h} + 75\phi_P^{2h}) \\
\phi_{12}^h &= \frac{1}{16384} (105\phi_A^{2h} - 495\phi_B^{2h} + 1155\phi_C^{2h} + 1155\phi_D^{2h} + 945\phi_E^{2h} - 4455\phi_F^{2h} \\
&\quad + 10395\phi_G^{2h} + 10395\phi_H^{2h} - 189\phi_I^{2h} + 891\phi_J^{2h} - 2079\phi_K^{2h} - 2079\phi_L^{2h} \\
&\quad + 35\phi_M^{2h} - 165\phi_N^{2h} + 385\phi_O^{2h} + 385\phi_P^{2h}) \\
\phi_{13}^h &= \frac{1}{16384} (-539\phi_A^{2h} - 539\phi_B^{2h} + 231\phi_C^{2h} - 49\phi_D^{2h} + 8085\phi_E^{2h} + 8085\phi_F^{2h} \\
&\quad - 3465\phi_G^{2h} + 735\phi_H^{2h} + 2695\phi_I^{2h} + 2695\phi_J^{2h} - 1155\phi_K^{2h} + 245\phi_L^{2h} \\
&\quad - 385\phi_M^{2h} - 385\phi_N^{2h} + 165\phi_O^{2h} - 35\phi_P^{2h})
\end{aligned}$$

$$\begin{aligned}
\phi_{14}^h &= \frac{1}{16384} (-105\phi_A^{2h} - 945\phi_B^{2h} + 189\phi_C^{2h} - 35\phi_D^{2h} + 1575\phi_E^{2h} + 14715\phi_F^{2h} \\
&\quad - 2835\phi_G^{2h} + 525\phi_H^{2h} + 525\phi_I^{2h} + 4725\phi_J^{2h} - 945\phi_K^{2h} \\
&\quad + 175\phi_L^{2h} - 75\phi_M^{2h} - 675\phi_N^{2h} + 135\phi_O^{2h} - 25\phi_P^{2h}) \\
\phi_{15}^h &= \frac{1}{16384} (49\phi_A^{2h} - 735\phi_B^{2h} - 245\phi_C^{2h} + 35\phi_D^{2h} - 735\phi_E^{2h} + 11025\phi_F^{2h} \\
&\quad + 3675\phi_G^{2h} - 525\phi_H^{2h} - 245\phi_I^{2h} + 3675\phi_J^{2h} + 1225\phi_K^{2h} \\
&\quad - 175\phi_L^{2h} + 35\phi_M^{2h} - 525\phi_N^{2h} - 175\phi_O^{2h} + 25\phi_P^{2h}) \\
\phi_{16}^h &= \frac{1}{16384} (35\phi_A^{2h} - 245\phi_B^{2h} - 735\phi_C^{2h} + 49\phi_D^{2h} - 525\phi_E^{2h} + 3675\phi_F^{2h} \\
&\quad + 11025\phi_G^{2h} - 735\phi_H^{2h} - 175\phi_I^{2h} + 1225\phi_J^{2h} + 3675\phi_K^{2h} \\
&\quad - 245\phi_L^{2h} + 25\phi_M^{2h} - 175\phi_N^{2h} - 525\phi_O^{2h} + 35\phi_P^{2h}) \\
\phi_{17}^h &= \frac{1}{16384} (-35\phi_A^{2h} - 189\phi_B^{2h} - 945\phi_C^{2h} - 105\phi_D^{2h} + 525\phi_E^{2h} - 2835\phi_F^{2h} \\
&\quad + 14175\phi_G^{2h} + 1575\phi_H^{2h} + 175\phi_I^{2h} - 945\phi_J^{2h} + 4725\phi_K^{2h} \\
&\quad + 525\phi_L^{2h} - 25\phi_M^{2h} + 135\phi_N^{2h} - 675\phi_O^{2h} - 75\phi_P^{2h}) \\
\phi_{18}^h &= \frac{1}{16384} (-49\phi_A^{2h} + 231\phi_B^{2h} - 539\phi_C^{2h} - 539\phi_D^{2h} + 735\phi_E^{2h} - 3465\phi_F^{2h} \\
&\quad + 8085\phi_G^{2h} + 8085\phi_H^{2h} + 245\phi_I^{2h} - 1155\phi_J^{2h} + 2695\phi_K^{2h} \\
&\quad + 2695\phi_L^{2h} - 35\phi_M^{2h} + 165\phi_N^{2h} - 385\phi_O^{2h} - 385\phi_P^{2h}) \\
\phi_{19}^h &= \frac{1}{16384} (-385\phi_A^{2h} - 385\phi_B^{2h} + 165\phi_C^{2h} - 35\phi_D^{2h} + 2695\phi_E^{2h} + 2695\phi_F^{2h} \\
&\quad - 1155\phi_G^{2h} + 245\phi_H^{2h} + 8085\phi_I^{2h} + 8085\phi_J^{2h} - 3465\phi_K^{2h} \\
&\quad + 735\phi_L^{2h} - 539\phi_M^{2h} - 539\phi_N^{2h} + 231\phi_O^{2h} - 49\phi_P^{2h}) \\
\phi_{20}^h &= \frac{1}{16384} (-75\phi_A^{2h} - 675\phi_B^{2h} + 135\phi_C^{2h} - 25\phi_D^{2h} + 525\phi_E^{2h} + 4725\phi_F^{2h} \\
&\quad - 945\phi_G^{2h} + 175\phi_H^{2h} + 1575\phi_I^{2h} + 14175\phi_J^{2h} - 2835\phi_K^{2h} \\
&\quad + 525\phi_L^{2h} - 105\phi_M^{2h} - 945\phi_N^{2h} + 189\phi_O^{2h} - 35\phi_P^{2h}) \\
\phi_{21}^h &= \frac{1}{16384} (35\phi_A^{2h} - 525\phi_B^{2h} - 175\phi_C^{2h} + 25\phi_D^{2h} - 245\phi_E^{2h} + 3675\phi_F^{2h} \\
&\quad + 1225\phi_G^{2h} - 175\phi_H^{2h} - 735\phi_I^{2h} + 11025\phi_J^{2h} + 3675\phi_K^{2h} \\
&\quad - 525\phi_L^{2h} + 49\phi_M^{2h} - 735\phi_N^{2h} - 245\phi_O^{2h} + 35\phi_P^{2h}) \\
\phi_{22}^h &= \frac{1}{16384} (25\phi_A^{2h} - 175\phi_B^{2h} - 525\phi_C^{2h} + 35\phi_D^{2h} - 175\phi_E^{2h} + 1225\phi_F^{2h} \\
&\quad + 3675\phi_G^{2h} - 245\phi_H^{2h} - 525\phi_I^{2h} + 3675\phi_J^{2h} + 11025\phi_K^{2h} \\
&\quad - 735\phi_L^{2h} + 35\phi_M^{2h} - 245\phi_N^{2h} - 735\phi_O^{2h} + 49\phi_P^{2h})
\end{aligned}$$

$$\begin{aligned}
\phi_{23}^h &= \frac{1}{16384} (-25\phi_A^{2h} + 135\phi_B^{2h} - 675\phi_C^{2h} - 75\phi_D^{2h} + 175\phi_E^{2h} - 945\phi_F^{2h} \\
&\quad + 4725\phi_G^{2h} + 525\phi_H^{2h} + 525\phi_I^{2h} - 2835\phi_J^{2h} + 14175\phi_K^{2h} \\
&\quad + 1575\phi_L^{2h} - 35\phi_M^{2h} + 189\phi_N^{2h} - 945\phi_O^{2h} - 105\phi_P^{2h}) \\
\phi_{24}^h &= \frac{1}{16384} (-35\phi_A^{2h} + 165\phi_B^{2h} - 385\phi_C^{2h} - 385\phi_D^{2h} + 245\phi_E^{2h} - 1155\phi_F^{2h} \\
&\quad + 2695\phi_G^{2h} + 2695\phi_H^{2h} + 735\phi_I^{2h} - 3465\phi_J^{2h} + 8085\phi_K^{2h} \\
&\quad + 8085\phi_L^{2h} - 49\phi_M^{2h} + 231\phi_N^{2h} - 539\phi_O^{2h} - 539\phi_P^{2h}) \\
\phi_{25}^h &= \frac{1}{16384} (385\phi_A^{2h} + 385\phi_B^{2h} - 165\phi_C^{2h} + 35\phi_D^{2h} - 2079\phi_E^{2h} - 2079\phi_F^{2h} \\
&\quad + 891\phi_G^{2h} - 189\phi_H^{2h} + 10395\phi_I^{2h} + 10395\phi_J^{2h} + 4455\phi_K^{2h} \\
&\quad + 945\phi_L^{2h} + 1155\phi_M^{2h} + 1155\phi_N^{2h} - 495\phi_O^{2h} + 105\phi_P^{2h}) \\
\phi_{26}^h &= \frac{1}{16384} (75\phi_A^{2h} + 675\phi_B^{2h} - 135\phi_C^{2h} + 25\phi_D^{2h} - 405\phi_E^{2h} + 3645\phi_F^{2h} \\
&\quad + 729\phi_G^{2h} - 135\phi_H^{2h} + 2025\phi_I^{2h} + 18225\phi_J^{2h} - 3645\phi_K^{2h} \\
&\quad + 675\phi_L^{2h} + 225\phi_M^{2h} + 2025\phi_N^{2h} - 405\phi_O^{2h} + 75\phi_P^{2h}) \\
\phi_{27}^h &= \frac{1}{16384} (-35\phi_A^{2h} + 525\phi_B^{2h} + 175\phi_C^{2h} - 25\phi_D^{2h} + 189\phi_E^{2h} - 2835\phi_F^{2h} \\
&\quad - 945\phi_G^{2h} + 135\phi_H^{2h} - 945\phi_I^{2h} + 14175\phi_J^{2h} + 4725\phi_K^{2h} \\
&\quad - 675\phi_L^{2h} - 105\phi_M^{2h} + 1575\phi_N^{2h} + 525\phi_O^{2h} - 75\phi_P^{2h}) \\
\phi_{28}^h &= \frac{1}{16384} (-25\phi_A^{2h} + 175\phi_B^{2h} + 525\phi_C^{2h} - 35\phi_D^{2h} + 135\phi_E^{2h} - 945\phi_F^{2h} \\
&\quad - 2835\phi_G^{2h} - 189\phi_H^{2h} - 675\phi_I^{2h} + 4725\phi_J^{2h} + 14175\phi_K^{2h} \\
&\quad - 945\phi_L^{2h} - 75\phi_M^{2h} + 525\phi_N^{2h} + 1575\phi_O^{2h} - 105\phi_P^{2h}) \\
\phi_{29}^h &= \frac{1}{16384} (25\phi_A^{2h} - 135\phi_B^{2h} + 675\phi_C^{2h} + 75\phi_D^{2h} - 135\phi_E^{2h} + 729\phi_F^{2h} \\
&\quad - 3645\phi_G^{2h} - 405\phi_H^{2h} + 675\phi_I^{2h} - 3645\phi_J^{2h} + 18225\phi_K^{2h} \\
&\quad + 2025\phi_L^{2h} + 75\phi_M^{2h} - 405\phi_N^{2h} - 2025\phi_O^{2h} + 225\phi_P^{2h}) \\
\phi_{30}^h &= \frac{1}{16384} (35\phi_A^{2h} - 165\phi_B^{2h} + 385\phi_C^{2h} + 385\phi_D^{2h} - 189\phi_E^{2h} - 891\phi_F^{2h} \\
&\quad + 2079\phi_G^{2h} - 2079\phi_H^{2h} + 945\phi_I^{2h} - 4455\phi_J^{2h} + 10395\phi_K^{2h} \\
&\quad + 10395\phi_L^{2h} + 105\phi_M^{2h} - 495\phi_N^{2h} + 1155\phi_O^{2h} + 1155\phi_P^{2h}) \\
\phi_{31}^h &= \frac{1}{16384} (539\phi_A^{2h} + 539\phi_B^{2h} - 231\phi_C^{2h} + 49\phi_D^{2h} - 2541\phi_E^{2h} - 2541\phi_F^{2h} \\
&\quad + 1089\phi_G^{2h} - 231\phi_H^{2h} + 5929\phi_I^{2h} + 5929\phi_J^{2h} - 2541\phi_K^{2h} \\
&\quad + 539\phi_L^{2h} + 5929\phi_M^{2h} + 5929\phi_N^{2h} - 2541\phi_O^{2h} + 539\phi_P^{2h})
\end{aligned}$$

$$\phi_{32}^h = \frac{1}{16384} (105\phi_A^{2h} + 945\phi_B^{2h} - 189\phi_C^{2h} + 35\phi_D^{2h} - 495\phi_E^{2h} - 4455\phi_F^{2h} \\ + 891\phi_G^{2h} - 165\phi_H^{2h} + 1155\phi_I^{2h} + 10395\phi_J^{2h} - 2079\phi_K^{2h} \\ + 385\phi_L^{2h} + 1155\phi_M^{2h} + 10395\phi_N^{2h} - 2079\phi_O^{2h} + 385\phi_P^{2h})$$

$$\phi_{33}^h = \frac{1}{16384} (-49\phi_A^{2h} + 735\phi_B^{2h} + 245\phi_C^{2h} - 35\phi_D^{2h} + 231\phi_E^{2h} - 3465\phi_F^{2h} \\ - 1155\phi_G^{2h} + 165\phi_H^{2h} - 539\phi_I^{2h} + 8085\phi_J^{2h} + 2695\phi_K^{2h} \\ - 385\phi_L^{2h} - 539\phi_M^{2h} + 8085\phi_N^{2h} + 2695\phi_O^{2h} - 385\phi_P^{2h})$$

$$\phi_{34}^h = \frac{1}{16384} (-35\phi_A^{2h} + 245\phi_B^{2h} + 735\phi_C^{2h} - 49\phi_D^{2h} + 165\phi_E^{2h} - 1155\phi_F^{2h} \\ - 3465\phi_G^{2h} + 231\phi_H^{2h} - 385\phi_I^{2h} + 2695\phi_J^{2h} + 8085\phi_K^{2h} \\ - 539\phi_L^{2h} - 385\phi_M^{2h} + 2695\phi_N^{2h} + 8085\phi_O^{2h} - 539\phi_P^{2h})$$

$$\phi_{35}^h = \frac{1}{16384} (35\phi_A^{2h} - 189\phi_B^{2h} + 945\phi_C^{2h} + 105\phi_D^{2h} - 165\phi_E^{2h} + 891\phi_F^{2h} \\ - 4455\phi_G^{2h} - 495\phi_H^{2h} + 385\phi_I^{2h} - 2079\phi_J^{2h} + 10395\phi_K^{2h} \\ + 1155\phi_L^{2h} + 385\phi_M^{2h} - 2079\phi_N^{2h} + 10395\phi_O^{2h} + 1155\phi_P^{2h})$$

$$\phi_{36}^h = \frac{1}{16384} (49\phi_A^{2h} - 231\phi_B^{2h} + 539\phi_C^{2h} + 539\phi_D^{2h} - 231\phi_E^{2h} + 1089\phi_F^{2h} \\ - 2541\phi_G^{2h} - 2541\phi_H^{2h} + 539\phi_I^{2h} - 2541\phi_J^{2h} + 5929\phi_K^{2h} \\ + 5929\phi_L^{2h} + 39\phi_M^{2h} - 2541\phi_N^{2h} + 5929\phi_O^{2h} + 5929\phi_P^{2h})$$