

MARCIO AUGUSTO VILLELA PINTO

**COMPORTAMENTO DO *MULTIGRID GEOMÉTRICO* EM
PROBLEMAS DE TRANSFERÊNCIA DE CALOR**

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

Orientador: Carlos Henrique Marchi.

Curitiba

2006

TERMO DE APROVAÇÃO

MARCIO AUGUSTO VILLELA PINTO

COMPORTAMENTO DO MULTIGRID GEOMÉTRICO EM PROBLEMAS DE TRANSFERÊNCIA DE CALOR

Tese aprovada como requisito parcial para a obtenção do grau de Doutor em Ciências, na área de concentração Programação Matemática, no Programa de Pós-Graduação em Métodos Numéricos em Engenharia, da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. Dr. Carlos Henrique Marchi

(orientador)

Departamento de Engenharia Mecânica,
UFPR

Profª Dra. Neida Maria Patias Volpi

Departamento de Matemática,
UFPR

Prof. Ph.D. Clóvis Raimundo Maliska
Departamento de Engenharia Mecânica,
UFSC

Profª Dra. Sandra Mara Cardoso Malta
Departamento de Matemática e Estatística,
UNIRIO

Prof. Dr. Luis Carlos Matioli
Departamento de Matemática,
UFPR

Profª Dra. Viviana Cocco Mariani
Departamento de Engenharia Mecânica,
PUC-PR

Curitiba, 20 de dezembro de 2006.

A Deus, por tudo.

Agradecimentos

Agradeço ao meu orientador, Prof. Dr. Carlos Henrique Marchi, por ter aceitado orientar-me neste trabalho e pelo conhecimento recebido ao longo desta orientação.

Agradeço aos membros da banca examinadora, Prof. Ph.D. Clóvis Raimundo Maliska, Prof. Dr. Luiz Carlos Matioli, Prof^a Dr^a Neida Maria Patias Volpi, Prof^a Dr^a Sandra Mara Cardoso Malta e Prof^a Dr^a Viviana Cocco Mariani, pelo tempo despendido na leitura deste trabalho e pelas importantes sugestões apontadas.

Agradeço ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da Universidade Federal do Paraná (UFPR) pela oportunidade de cursar o doutorado e ao Departamento de Matemática e Estatística da Universidade Estadual de Ponta Grossa (UEPG) pela licença concedida para a dedicação ao curso.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à UEPG pelo suporte financeiro.

Agradeço aos meus familiares e amigos pela ajuda e compreensão durante este meu percurso.

Agradeço ao meu amigo Geremias Vieira por estar sempre próximo.

Resumo. Sobre o tempo de CPU necessário para resolver problemas unidimensionais lineares e não-linear e um problema bidimensional linear, verifica-se o efeito causado por diversos valores de razão de engrossamento, vários tamanhos de malha, número de iterações internas, número de níveis de malhas, tolerâncias, estimativas iniciais, solvers, esquemas de correção (CS) e aproximação completa (FAS), diversos valores de razão de aspecto de malha e algoritmos multigrid para problemas anisotrópicos. Os problemas considerados são lineares unidimensionais (equação de difusão e de advecção-difusão), não-linear unidimensional (equação de Burgers) e linear bidimensional (equação de Laplace), todos com condições de contorno de Dirichlet. O método de diferenças finitas é usado para discretizar as equações diferenciais. Os sistemas de equações algébricas são resolvidos com diversos solvers associados ao método multigrid geométrico com ciclo V. Para os problemas isotrópicos são feitas comparações entre os esquemas CS e FAS. Para o problema anisotrópico, quatro tipos de algoritmos de engrossamento são considerados, envolvendo engrossamento padrão e semi-engrossamento. Alguns resultados confirmam os da literatura e outros novos são apresentados. Entre outros, verificou-se que: o esquema FAS é mais rápido do que o esquema CS e que o algoritmo do tipo semi-engrossamento seguido de engrossamento padrão é o mais rápido entre os quatro testados para o problema anisotrópico.

Palavras-chave: Solver, CFD, diferenças finitas, métodos numéricos, anisotropia, multigrid.

Abstract. On the necessary CPU time to solve one-dimensional linear and nonlinear problems and a two-dimensional linear problem, one verifies the effect considered by several coarsening ratios values, several number of nodes, number of inner iterations, number of grid levels, tolerances, initial estimates, solvers, correction (CS) and full approximation schemes (FAS), several grid aspect ratios values and multigrid algorithms to anisotropic problems. The considered problems are one-dimensional linear (diffusion and advection-diffusion equations), one-dimensional nonlinear (Burgers's equation) and two-dimensional linear (Laplace's equation) problems with Dirichlet's boundary conditions. The finite difference method is used to discretize the differential equations. The algebraic systems are solved by several solvers associated to geometric multigrid method with V-cycle. Comparisons between CS and FAS schemes are made to isotropic problems. Four types of coarsening algorithms are considered, involving standard coarsening and semicoarsening to anisotropic problem. Some literature results are confirmed and some new ones are presented. The main conclusions are: the FAS scheme is faster than CS scheme and the SE-EP (partial semicoarsening) algorithm is faster than other studied algorithms to the anisotropic problem.

Key-word: Solver, CFD, finite difference, numerical methods, anisotropy, multigrid.

Sumário

Listas de figuras	10
Listas de tabelas	13
Listas de símbolos	15
1 Introdução.....	19
1.1 Generalidades em dinâmica dos fluidos computacional.....	19
1.2 Generalidades do método <i>multigrid</i>	22
1.3 Motivação	25
1.4 Objetivos.....	27
1.5 Revisão bibliográfica.....	28
1.6 Delineamento deste texto	30
2 Fundamentação.....	32
2.1 Métodos iterativos básicos	32
2.2 Análise de convergência.....	33
2.3 Análise de erros de Fourier.....	37
2.4 Aplicação da análise de erros de Fourier na equação de Poisson 1D.....	39
2.5 Princípios fundamentais do método <i>multigrid</i>	44
2.6 Resumo do capítulo 2	46
3 Método <i>multigrid</i>	47
3.1 Algoritmos	47
3.2 Processo de restrição	53
3.3 Processo de prolongação	56
3.4 Problemas anisotrópicos.....	58
3.5 Resumo do capítulo 3	61
4 Problemas unidimensionais lineares: equações de difusão e de advecção-difusão	62
4.1 Modelos matemáticos	62
4.2 Modelos numéricos	64
4.2.1 Discretização	65
4.2.2 Esquema de correção (CS)	68
4.2.3 Esquema de aproximação completa (FAS)	70
4.2.4 Dados de implementação.....	70
4.3 Resultados numéricos com o esquema de correção (CS)	72
4.3.1 Iteração interna	74
4.3.2 Nível de malha.....	75
4.3.3 Razão de engrossamento	77
4.3.4 Tolerância e estimativa inicial.....	79
4.4 Resultados numéricos com o esquema de aproximação completa (FAS)	81
4.4.1 Iteração interna	82
4.4.2 Nível de malha.....	83
4.4.3 Razão de engrossamento	84
4.4.4 Tolerância e estimativa inicial.....	86
4.5 Esquema de correção (CS) <i>versus</i> esquema de aproximação completa (FAS)	87
4.6 Conclusão do capítulo 4	89
5 Problema unidimensional não-linear: equação de Burgers	90
5.1 Modelo matemático	90
5.2 Modelos numéricos	91
5.2.1 Discretização	91
5.3 Resultados numéricos com o esquema de correção (CS)	94
5.3.1 Iteração interna	94
5.3.2 Nível de malha.....	96

5.3.3	Razão de engrossamento	97
5.3.4	Tolerância e estimativa inicial.....	99
5.4	Resultados numéricos com o esquema de aproximação completa (FAS)	100
5.4.1	Iteração interna	101
5.4.2	Nível de malha.....	103
5.4.3	Razão de engrossamento	105
5.4.4	Tolerância e estimativa inicial.....	109
5.5	Esquema de correção (CS) <i>versus</i> esquema de aproximação completa (FAS)	111
5.6	Conclusão do capítulo 5	112
6	Problema bidimensional linear isotrópico: equação de Laplace	114
6.1	Modelo matemático	114
6.2	Modelo numérico.....	116
6.2.1	Discretização	116
6.3	Resultados numéricos com o esquema de correção (CS)	118
6.3.1	Iteração interna	119
6.3.2	Nível de malha.....	121
6.3.3	Razão de engrossamento	122
6.4	Resultados numéricos com o esquema de aproximação completa (FAS)	126
6.4.1	Iteração interna	127
6.4.2	Nível de malha.....	129
6.4.3	Razão de engrossamento	130
6.5	Esquema de correção (CS) <i>versus</i> esquema de aproximação completa (FAS)	134
6.6	Conclusão do capítulo 6	137
7	Problema bidimensional linear anisotrópico: equação de Laplace.....	139
7.1	Modelos matemático e numérico.....	140
7.2	Dados de implementação.....	141
7.3	Resultados numéricos	143
7.3.1	Iteração interna (<i>ITI</i>) e razão de aspecto (<i>RA</i>).....	144
7.3.2	Efeito do número de incógnitas (<i>N</i>) sobre a razão de aspecto (<i>RA</i>)	145
7.3.3	Algoritmo de semi-engrossamento parcial (SE-EP).....	147
7.4	Conclusão do capítulo 7	150
8	Conclusão	152
9	Referências bibliográficas	157
Apêndice A - Métodos iterativos básicos e outros <i>solvers</i>	163	
A.1	Exemplos de métodos iterativos básicos	163
A.2	Métodos para a resolução de sistemas de equações	165
Método TDMA	165	
Método Gauss-Seidel.....	167	
Método MSI.....	168	
Apêndice B - Soluções analíticas para os problemas uni e bidimensionais	171	
B.1	Solução da Equação de difusão 1D	171
B.2	Solução da Equação de advecção-difusão 1D	173
B.3	Solução da Equação de Burgers 1D	175
B.4	Solução da Equação de Laplace 2D	176
Apêndice C - Testes adicionais para critério de convergência.....	178	
Apêndice D - Artigos completos com resultados desta Tese publicados em anais de Eventos	180	
D.1	Efeito de Parâmetros do Método <i>Multigrid</i> sobre o Tempo de CPU para a Equação de Burgers Unidimensional	181

D.2	Effect of Parameters of <i>Multigrid</i> Method on the CPU Time for One-dimensional Problems	195
D.3	Efeito de Parâmetros do Método <i>Multigrid</i> CS e FAS sobre o Tempo de CPU em Problemas 1D Lineares e Não-lineares	204
D.4	Efeito da Razão de Aspecto de Malha sobre o Tempo de CPU para a Equação de Laplace 2D Resolvida com o Método <i>Multigrid</i>	223
D.5	Efeito de Parâmetros do Método <i>Multigrid</i> CS e FAS sobre o Tempo de CPU para a Equação de Laplace Bidimensional	238

Lista de figuras

Figura 1.1: Exemplos de malhas: (a) regular e (b e c) irregular	21
Figura 1.2: Exemplos de: a) malha fina e b) malha grossa	23
Figura 2.1: Modos de Fourier com $k = 1, 3, 6$ (BRIGGS et al., 2000)	38
Figura 2.2: (a) Jacobi ponderado com $\omega = 2/3$, (b) Gauss-Seidel regular aplicados ao problema unidimensional com $N = 64$, estimativas iniciais para $k = 1, 3, 6$ e para 100 iterações (BRIGGS et al., 2000)	42
Figura 2.3: Jacobi ponderado com $\omega = 2/3$ aplicado ao problema unidimensional com $N = 64$, estimativa inicial $(v_1+v_6+v_{32})/3$ e para 100 iterações (BRIGGS et al., 2000)	42
Figura 2.4: Jacobi ponderado com $\omega = 2/3$ aplicado ao problema unidimensional com $N = 64$, com estimativa inicial $(v_2+v_{16})/2$ antes (esquerda) e depois (direita) de 10 iterações (BRIGGS et al., 2000)	43
Figura 2.5: (a) Jacobi regular ($\omega = 1$) e (b) Jacobi ponderado com $\omega = 2/3$ aplicados ao problema unidimensional com $N = 64$, estimativas iniciais com $1 \leq k \leq 63$, com norma do erro reduzida pelo fator de 100 (BRIGGS et al., 2000)	44
Figura 2.6: Número de ondas $k = 4$ sobre uma malha fina com $N = 12$ e sobre uma malha grossa com $N = 6$ (BRIGGS et al., 2000)	45
Figura 3.1: Malhas grossas uni e bidimensional	48
Figura 3.2: Diagrama ciclo V	52
Figura 3.3: Operador de restrição (ponderação completa) da malha fina para a malha grossa	55
Figura 3.4: Operador de prolongação (interpolação linear) da malha grossa para a malha fina	57
Figura 3.5: (a) malha fina original e (b) malha grossa com semi-engrossamento na direção x	60
Figura 4.1: Domínio unidimensional de cálculo e condições de contorno	64
Figura 4.2: Malha 1D uniforme	65
Figura 4.3: Soluções analítica e numéricas para $N = 32$ para a equação de difusão com CDS	73
Figura 4.4: Soluções analítica e numéricas para $N = 32$ para a equação de advecção-difusão com UDS/CDS	73
Figura 4.5: Tempo de CPU versus ITI para a equação de advecção-difusão com o esquema CS, diversos r e com o esquema numérico UDS/CDS	74
Figura 4.6: Tempo de CPU versus L para a equação de advecção-difusão com o esquema CS, diversos r e com o esquema numérico UDS/CDS	76
Figura 4.7: Tempo de CPU versus N para a equação de advecção-difusão com o esquema CS, diversos r , com o esquema numérico UDS/CDS e diversos métodos	78
Figura 4.8: Tempo de CPU versus ITI para a equação de advecção-difusão com o esquema FAS, diversos r e com o esquema numérico UDS/CDS	82
Figura 4.9: Tempo de CPU versus L para a equação de advecção-difusão com o esquema FAS, diversos r e com o esquema numérico UDS/CDS	84
Figura 4.10: Tempo de CPU versus N para a equação de advecção-difusão com o esquema FAS, diversos r , com o esquema numérico UDS/CDS e diversos métodos	85
Figura 4.11: Detalhe da figura 4.10	85
Figura 4.12: Tempo de CPU versus N para a equação de advecção-difusão com os esquemas CS e FAS, $r = 2$ e $r = 3$, com o esquema UDS/CDS	87
Figura 5.1: Tempo de CPU versus ITI para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS	95

Figura 5.2: Tempo de CPU versus L para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS.....	97
Figura 5.3: Tempo de CPU versus N para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS.....	98
Figura 5.4: Soluções analítica e numéricas para $N = 32$ com esquema FAS e esquema numérico UDS/CDS para a equação de Burgers	100
Figura 5.5: Soluções analítica e numéricas para $N = 32$ com esquema FAS e esquema numérico CDS_CA/CDS para a equação de Burgers.....	101
Figura 5.6: Tempo de CPU versus ITI para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico UDS/CDS.....	102
Figura 5.7: Tempo de CPU versus ITI para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico CDS_CA/CDS	103
Figura 5.8: Tempo de CPU versus L para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico UDS/CDS.....	104
Figura 5.9: Tempo de CPU versus L para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico CDS_CA/CDS	105
Figura 5.10: Tempo de CPU versus N para a equação de Burgers com o esquema FAS, diversos r , com o esquema numérico UDS/CDS e para diversos métodos	107
Figura 5.11: Detalhe da figura 5.10.....	107
Figura 5.12: Tempo de CPU versus N para a equação de Burgers com o esquema FAS, $r = 2$ e $r = 3$, com o esquema numérico CDS_CA/CDS e para diversos métodos	108
Figura 5.13: Detalhe da figura 5.12	108
Figura 5.14: Tempo de CPU versus N para a equação de Burgers com os esquemas CS e FAS, $r = 2$ e $r = 3$ e com o esquema numérico UDS/CDS	111
Figura 6.1: Domínio bidimensional de cálculo para a equação de Laplace	115
Figura 6.2: Malha 2D uniforme	116
Figura 6.3: Tempo de CPU versus ITI para a equação de Laplace com o esquema CS, com o esquema numérico CDS/CDS, $r = 2$ e solver MSI	120
Figura 6.4: Tempo de CPU versus L para a equação de Laplace com o esquema CS, $r = 2$, com o esquema numérico CDS/CDS e diversos solvers.....	122
Figura 6.5: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, com o esquema numérico CDS/CDS e diversos métodos	123
Figura 6.6: Tempo de CPU versus N para a equação de Laplace com o esquema CS, diversos r , com o esquema numérico CDS/CDS e o solver MSI	125
Figura 6.7: Tempo de CPU versus ITI para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e o solver MSI	128
Figura 6.8: Tempo de CPU versus L para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e diversos solvers.....	130
Figura 6.9: Tempo de CPU versus N para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e com diversos solvers	131
Figura 6.10: Tempo de CPU versus N para a equação de Laplace com o esquema FAS, diversos r , com o esquema numérico CDS/CDS e solver MSI.....	133
Figura 6.11: Detalhe da figura 6.10	133
Figura 6.12: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 2$, com o esquema numérico CDS/CDS e solver MSI	135
Figura 6.13: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 3$, com o esquema numérico CDS/CDS e solver MSI	135
Figura 6.14: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 2$ e $r = 3$, com o esquema numérico CDS/CDS e solver MSI	136

Figura 7.1: Exemplos de malhas: a) isotrópica e b) anisotrópica com $RA = 4$	139
Figura 7.2: Engrossamentos: EP (1a-1c), EP-SE (2a-2e), SE (3a-3c) e SE-EP (4a-4e)	142
Figura 7.3: Tempo de CPU <i>versus</i> N para a equação de Laplace com esquema CS, $r = 2, RA = 2$, com o esquema numérico CDS/CDS e diversos algoritmos.....	145
Figura 7.4: Tempo de CPU <i>versus</i> N para a equação de Laplace com esquema CS, $r = 2, RA = 16$, com o esquema numérico CDS/CDS e diversos algoritmos	146
Figura 7.5: Tempo de CPU <i>versus</i> N para a equação de Laplace com esquema CS, $r = 2, RA = 128$, com o esquema numérico CDS/CDS e diversos algoritmos	147
Figura 7.6: Tempo de CPU <i>versus</i> N para a equação de Laplace com esquema CS, $r = 2$, diversos RA , com o esquema numérico CDS/CDS e algoritmo SE-EP	149

Lista de tabelas

Tabela 1.1: Comparaçao entre as trés técnicas de solução	19
Tabela 1.2: Razão de convergência assintótica para o método MSG em várias malhas	26
Tabela 3.1: Esquema FAS para duas malhas (adaptado de Briggs et al., 2000)	49
Tabela 3.2: Esquema CS para duas malhas (adaptado de Briggs et al., 2000)	50
Tabela 3.3: Esquema CS para várias malhas com ciclo V (adaptado de Briggs et al., 2000)	53
Tabela 4.1: Algumas malhas para cada uma das razões testadas	72
Tabela 4.2: Variação de parâmetros de acordo com a variação da tolerância no esquema CDS para a equação de difusão	80
Tabela 4.3: Variação de parâmetros de acordo com a variação da tolerância no esquema UDS/CDS para a equação de advecção-difusão	80
Tabela 4.4: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema CDS para a equação de difusão	81
Tabela 4.5: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema UDS/CDS para a equação de advecção-difusão	81
Tabela 5.1: Variação de parâmetros de acordo com a variação da tolerância no esquema UDS/CDS	110
Tabela 5.2: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema UDS/CDS	110
Tabela 6.1: Algumas malhas para cada uma das razões testadas para os esquemas CS e FAS	119
Tabela 6.2: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema CS	119
Tabela 6.3: Número ótimo de iterações internas ($ITI_{óptimo}$) para diversos solvers, esquema CS e $r = 2$	120
Tabela 6.4: Número ótimo de iterações internas ($ITI_{óptimo}$) para solver MSI, esquema CS e diversas razões de engrossamento	121
Tabela 6.5: Coeficiente (c) e ordem (p) dos métodos e solvers para o esquema CS	124
Tabela 6.6: Coeficiente (c) e ordem (p) das razões de engrossamento para o solver MSI e esquema CS	126
Tabela 6.7: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema FAS	127
Tabela 6.8: Número ótimo de iterações internas ($ITI_{óptimo}$) para diversos solvers, esquema FAS e $r = 2$	128
Tabela 6.9: Número ótimo de iterações internas ($ITI_{óptimo}$) para solver MSI, esquema FAS e diversas razões de engrossamento	129
Tabela 6.10: Coeficiente (c) e ordem (p) dos métodos e solvers para o esquema FAS	132
Tabela 6.11: Coeficiente (c) e ordem (p) das razões de engrossamento para o solver MSI e esquema FAS	134
Tabela 7.1: Norma infinito do erro para as malhas mais finas para cada uma das razões de aspectos (RA) testadas com o solver GS e o esquema CS	143
Tabela 7.2: Malhas utilizadas para determinar $ITI_{óptimo}$	144
Tabela 7.3: Valor do $ITI_{óptimo}$	145
Tabela 7.4: Malhas adicionais usadas para obter $ITI_{óptimo}$ com o algoritmo SE-EP	148
Tabela 7.5: $ITI_{óptimo}$ para o algoritmo SE-EP	148

Tabela 7.6: Valor do coeficiente (c) e ordem (p) da equação 6.10 para o algoritmo SE-EP	148
---	-----

Lista de símbolos

a_{ij}	Coeficientes da matriz A
A	Matriz dos coeficientes
A_i	i -ésima constante para a solução da EDO
A^h	Operador de malha fina
A^{2h}	Operador de malha grossa
A^H	Operador de malha grossa genérica
c	Coeficiente de ajuste de curvas
c_p	Calor específico do fluido
C	Comprimento do domínio 1D
CFD	<i>Computational Fluid Dynamic</i>
C_x	Comprimento do domínio 2D: direção x
C_y	Comprimento do domínio 2D: direção y
$\text{cond}(A)$	Número de condicionamento da matriz A
e	Erro de iteração
f	Vetor independente no sistema linear
h_x	Tamanho da malha na direção x
h_y	Tamanho da malha na direção y
I_h^{2h}	Operador de restrição
I_{2h}^h	Operador de prolongação
i	contador na direção coordenada x
\vec{i}	vetor canônico na direção coordenada x
ITI	Número de iterações internas
$\text{ITI}_{\text{ótimo}}$	Número ótimo de iterações internas
j	contador na direção coordenada y
\vec{j}	vetor canônico na direção coordenada y
k	Número do modo de Fourier
k'	Modo de Fourier complementar
\vec{k}	vetor canônico na direção coordenada z

K_r	Coeficiente para restrição
K_p	Coeficiente para prolongação
N	Número de incógnitas ou subintervalos do problema
N_x	Número de incógnitas na direção x
N_y	Número de incógnitas na direção y
L	Número de níveis
$L_{máximo}$	Número máximo de níveis
$L_{ótimo}$	Número ótimo de níveis
$O(\cdot)$	Ordem de convergência
p	Ordem do <i>solver</i>
Pe	Número de Peclet
\dot{q}	Geração de calor
R	Resíduo
R	Domínio real unidimensional
R^2	Domínio real bidimensional
r	Razão de engrossamento
RA	Razão de aspecto
Re	Número de Reynolds
S	Matriz de iteração
S_G	Matriz de iteração (Gauss-Seidel)
S_J	Matriz de iteração (Jacobi)
S_Φ	Termo fonte de cada Φ
S_ω	Matriz de iteração (Jacobi ponderado)
T	Temperatura
t_{CPU}	Tempo de CPU
u	Solução exata do sistema
u_x	Derivada primeira de u
u_{xx}	Derivada segunda de u
\vec{u}	Vetor velocidade
v	Solução aproximada do sistema

v_i	Aproximação da solução 1D
$v_{i,j}$	Aproximação da solução 2D
\mathbf{v}_k	k -ésimo modo de Fourier
x, y	Variáveis do sistema coordenado cartesiano 2D
$w_{k,j}$	j -ésima componente do k -ésimo modo de Fourier
Z	Conjunto dos números inteiros

Letras gregas

β	Constante para o método CDS_CA/CDS
ε	Tolerância
Γ	Coeficiente de difusão
φ	Informação a ser transferida entre malhas
Φ	Propriedade a ser transportada
η_1	Número de pré-suavização
η_2	Número de pós-suavização
κ	Condutividade Térmica
$\lambda(A)$	Autovalores de A
Λ	Seqüência de malhas
μ	Número de ciclos internos
ρ	Massa específica
$\rho(A)$	Raio espectral de A
σ	Constante para solução de EDO
ω	Fator de amortecimento
Ω^h	Malha fina
Ω^{2h}	Malha grossa com engrossamento padrão
Ω^H	Malha grossa genérica
Ω^{Kh}	Malha mais grossa

Símbolos

\approx	Aproximadamente igual
$\vec{\nabla}$	Operador <i>nabla</i>
$<<$	Muito menor
$\ \cdot \ _2$	Norma Euclidiana (ou norma-2)
$\ \cdot \ _\infty$	Norma do máximo (ou norma-infinito)

1 Introdução

Neste capítulo é feito um estudo das generalidades sobre dinâmica dos fluidos e sobre o método *multigrid*. Uma motivação para o tema desta proposta de Tese e os objetivos são apresentados. Em seguida é detalhada uma revisão bibliográfica sobre *multigrid* de uma forma geral, sobre razões de engrossamento e sobre problemas anisotrópicos.

1.1 Generalidades em dinâmica dos fluidos computacional

Modelos matemáticos na dinâmica dos fluidos computacional surgem em fenômenos físicos que envolvem fluidos em movimento, com ou sem troca de calor (Fortuna, 2000; Maliska, 2004). Estes modelos matemáticos, em geral, não têm soluções analíticas conhecidas. Buscam-se então soluções numéricas transformando-se o modelo contínuo em um modelo discreto.

O uso de métodos numéricos, juntamente com as técnicas experimentais e a análise teórica, complementam-se durante um projeto que envolva escoamento de fluidos e o estudo de modelos teóricos. A tabela 1.1, extraída de Tannehill et al. (1997), compara as três estratégias.

Tabela 1.1: Comparação entre as três técnicas de solução

Técnica	Vantagens	Desvantagens
Experimental	- mais realista	- equipamento exigido - problemas de escala - dificuldades de medição - custo operacional
Teórica	- mais geral - fórmula fechada	- restrita a geometrias e processos físicos simples - geralmente restrita a problemas lineares
Numérica	- não há restrição quanto à linearidade - geometrias e processos complicados - evolução temporal do processo	- erros de truncamento - prescrição das condições de contorno apropriadas - custo computacional

Um método de discretização muito usado é o método de diferenças finitas (GOLUB e ORTEGA, 1992; TANNEHILL et al., 1997), onde, em problemas unidimensionais, o domínio

$\{x \in R : 0 \leq x \leq 1\}$ é particionado em N subintervalos (elementos), introduzindo uma malha com os pontos

$$x_i = (i-1)h, \quad i = 1, \dots, N+1 \text{ e } h = \frac{1}{N},$$

onde h é o comprimento de cada subintervalo. Isto estabelece uma malha com subintervalos (intervalo entre dois pontos consecutivos) de tamanho h que se denota por Ω^h . Neste trabalho, por facilidade de compreensão, adota-se o termo “elemento” para designar este subintervalo. Este termo é usado no mesmo sentido que na literatura referente ao método dos elementos finitos. A cada um dos $N-1$ pontos interiores da malha, uma equação diferencial pode ser substituída por aproximações de diferenças finitas (TANNEHILL et al., 1997; FERZIGER e PERIC, 1999). Em problemas bidimensionais, o domínio $\{(x, y) \in R^2 : 0 \leq x, y \leq 1\}$ é particionado em subconjuntos através de um número de incógnitas ou número de pontos (nós), dado por $N = N_x N_y$, onde N_x e N_y são os números de incógnitas nas direções coordenadas x e y , respectivamente, introduzindo uma malha com os pontos

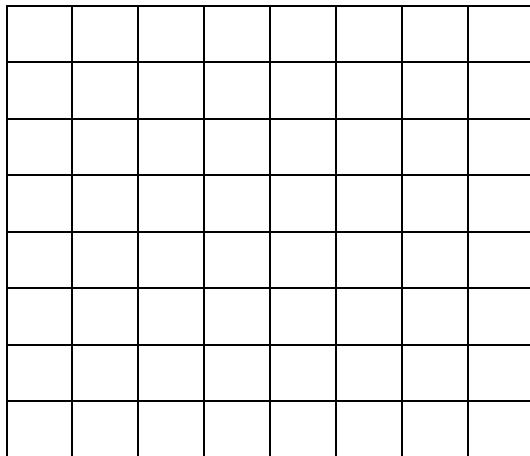
$$(x_i, y_j) = ((i-1)h_x, (j-1)h_y), \text{ com } h_x = \frac{1}{N_x - 1} \text{ e } h_y = \frac{1}{N_y - 1},$$

onde $i = 1, \dots, N_x$, $j = 1, \dots, N_y$ e h_x e h_y os comprimentos de cada elemento nas direções coordenadas x e y , respectivamente. Isto estabelece uma malha com elementos de tamanho h_x por h_y que se denota por Ω^h , com razão de aspecto (RA) definida por

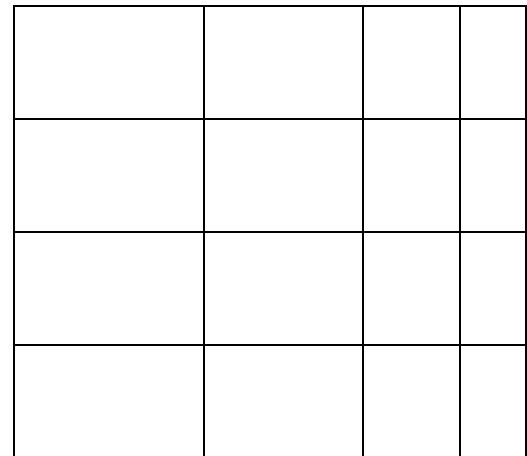
$$RA = \frac{h_x}{h_y}.$$

Se $RA = 1$, o problema chama-se isotrópico; caso contrário, anisotrópico (BRIGGS et al., 2000; DENDY et al., 1989). Neste caso tem-se anisotropia de malha ou anisotropia geométrica, termo que será definido com mais detalhes na seção 3.4.

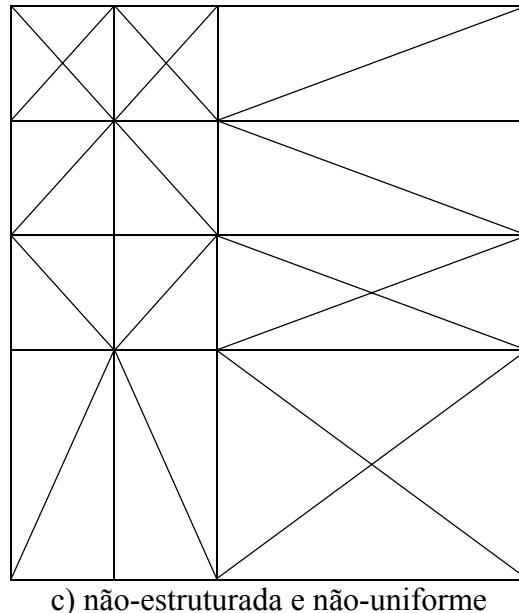
Dependendo da distribuição dos pontos discretos no domínio, a malha pode ser classificada em uniforme ou não-uniforme e estruturada ou não-estruturada. A figura 1.1 ilustra alguns exemplos de malhas regulares (uniforme e/ou estruturada) e malhas irregulares (não-uniforme e/ou não-estruturada).



a) estruturada e uniforme



b) estruturada e não-uniforme



c) não-estruturada e não-uniforme

Figura 1.1: Exemplos de malhas: (a) regular e (b e c) irregular

A discretização dos modelos matemáticos de dinâmica dos fluidos conduz a grandes sistemas de equações algébricas do tipo

$$Au = f, \quad (1.1)$$

onde A é uma matriz quadrada, f é o vetor independente e u é o vetor de incógnitas. A estrutura da matriz A depende do método e das aproximações numéricas usados para discretizar o modelo matemático.

Várias técnicas numéricas têm sido estudadas para resolver o sistema dado pela equação 1.1 com o menor custo computacional e a solução a mais próxima possível da exata (sem erros de iteração, FERZIGER e PERIC, 1999).

A resolução por métodos diretos não é recomendável, visto que na prática, a matriz dos coeficientes é muito grande e o custo da inversão da matriz é alto (GOLUB e VAN LOAN, 1989). Os métodos diretos são utilizados neste trabalho apenas com o intuito de comparação com alguns métodos iterativos, mesmo porque, estes métodos geralmente têm um custo computacional da ordem de $O(N^3)$, onde N é o número de incógnitas. Para problemas de grande porte os métodos iterativos são mais adequados (BURDEN e FAIRES, 1997). Neste caso, os métodos iterativos clássicos são da ordem de $O(N^2/2)$ (FERZIGER e PERIC, 1999). O método do gradiente conjugado (BURDEN e FAIRES, 1997; GOLUB e ORTEGA, 1992), introduzido por Hestenes e Stiefeld (1952), usa técnicas que são mais específicas para geometrias simples e é um método sensível ao condicionamento da matriz.

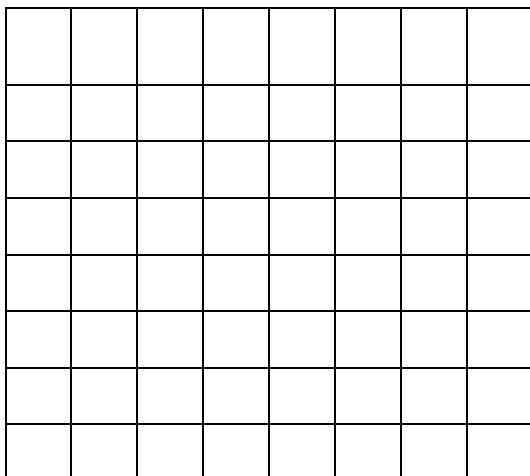
1.2 Generalidades do método *multigrid*

O método *multigrid*, proposto originalmente por Fedorenko (1964), é atualmente um método muito usado para resolver iterativamente sistemas de equações do tipo da equação 1.1.

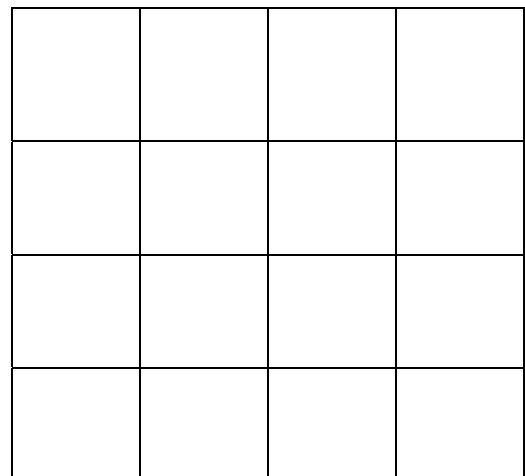
A idéia básica é usar um conjunto de malhas e executar, ora iterações em cada nível de malha, ora soluções aproximadas desta equação em malhas mais grossas (BRIGGS et al., 2000). São usados operadores para transferir informações (resíduo ou solução) da malha fina para a malha imediatamente mais grossa (processo chamado de restrição) e da malha grossa para a malha imediatamente mais fina (processo de prolongação). Veja na figura 1.2 exemplos de malhas fina e grossa.

Com o método *multigrid* podem ser usados alguns tipos de esquemas que se caracterizam por suas diferentes informações e a forma como as mesmas são transportadas (BRIGGS et al., 2000; TROTTERBERG et al., 2001; WESSELING, 1992): o esquema de correção (*Correction Scheme*, CS) e o esquema de aproximação completa (*Full Approximation Scheme*, FAS). No esquema CS, a equação 1.1 é resolvida apenas na malha

mais fina; nas malhas mais grossas, resolve-se a equação residual. Já no caso do esquema FAS, a equação 1.1 é resolvida em todas as malhas. O esquema CS é geralmente utilizado em problemas lineares e o esquema FAS, em problemas não-lineares (BRANDT, 1977). Outro esquema usa soluções nas malhas grossas para gerar estimativas iniciais melhoradas para as malhas mais finas. O esquema que usa esta estratégia chama-se *multigrid* completo (*Full Multigrid*, FMG).



a) malha fina



b) malha grossa

Figura 1.2: Exemplos de: a) malha fina e b) malha grossa

A razão de engrossamento, para o caso unidimensional, considerando malhas uniformes (os elementos possuem o mesmo tamanho h), é definida como

$$r = \frac{h_{\Omega^H}}{h_{\Omega^h}},$$

onde h_{Ω^h} representa o tamanho dos elementos da malha fina Ω^h e h_{Ω^H} o tamanho dos elementos da malha imediatamente mais grossa Ω^H .

Para o caso bidimensional algumas simplificações deverão ser consideradas neste trabalho:

- Será admitido engrossamento padrão (engrossamento cuja razão de engrossamento é única em ambas as direções coordenadas, ou seja, $r = r_x = r_y$);
- Será admitido semi-engrossamento (engrossamento em apenas uma das direções coordenadas e na outra não).

Considerando-se o caso bidimensional para malhas uniformes por direção (malhas uniformes em cada uma das direções coordenadas), tem-se

$$r = \frac{h_{x\Omega^H}}{h_{x\Omega^h}} = \frac{h_{y\Omega^H}}{h_{y\Omega^h}}. \quad (1.2)$$

Em particular, para o caso isotrópico, onde $RA = 1$, ou seja, $h = h_x = h_y$, a razão de engrossamento para o caso bidimensional é definida como no caso unidimensional.

Em cada malha o sistema de equações é resolvido com um método iterativo com propriedades de reduzir rapidamente os erros oscilatórios (propriedades de suavização). Com este conceito, vários trabalhos (BRANDT, 1977; MONTERO et al., 2001; NISHIDA e SATOFUKA, 1992; SATHYAMURTHY e PATANKAR, 1994; STÜBEN, 1999; WESSELING e OOSTERLEE, 2001) apresentaram bons resultados numéricos para problemas de dinâmica dos fluidos. Os resultados de Fedorenko (1964) mostram que a taxa de velocidade de convergência com o uso da técnica *multigrid* é muito melhor que a dos métodos iterativos puros, ou seja, métodos iterativos sem o uso da técnica multigrid, também conhecido como métodos *singlegrid* (SG). O objetivo do método *multigrid* é acelerar a convergência a fim de reduzir o tempo de CPU necessário para resolver a equação 1.1. Os melhores desempenhos do método *multigrid* são obtidos em problemas totalmente elípticos (WESSELING, 1992), ou seja, problemas dominados pela difusão; e os menores em problemas dominados pela advecção (FERZIGER e PERIC, 1999).

Diferentes problemas e malhas no contexto do método *multigrid* nos conduz ao estudo de duas técnicas conhecidas por *multigrid* geométrico e *multigrid* algébrico. No *multigrid* geométrico (WESSELING e OOSTERLEE, 2001), as variáveis u_i são definidas sob uma posição geométrica (pontos ou nós) na malha mais fina. Então, seleciona-se um subconjunto dessas posições geométricas como sendo a malha imediatamente mais grossa. Portanto, um subconjunto das variáveis u_i é usado para representar a solução na malha grossa. Este tipo de técnica é mais adequado quando se tem malha estruturada, onde se conhece as posições geométricas dos pontos. No *multigrid* algébrico (STÜBEN, 2001), identifica-se as variáveis u_i apenas com os índices das incógnitas $\{1, 2, \dots, N\}$. Então, seleciona-se um subconjunto de desses índices para representar as incógnitas da malha grossa. Este tipo de técnica é mais adequado quando se tem malha não-estruturada ou método sem malha.

Em Briggs et al. (2000), o uso do *multigrid* geométrico e algébrico para a equação de Poisson bidimensional em uma malha cartesiana com condições de contorno de Dirichlet é comparado. Nesta referência, é recomendado o uso do *multigrid* algébrico apenas em problemas onde o geométrico não possa ser utilizado, pois o *multigrid* algébrico é mais caro computacionalmente, em termos de memória, computação e tempo de CPU do que o *multigrid* geométrico.

A ferramenta básica para estudo de erros no método *multigrid* é a análise de Fourier. Ela é baseada na análise do erro após várias aplicações do método iterativo com um tipo especial de estimativa inicial, também conhecida como modo de Fourier. Tal análise produz uma boa aproximação do comportamento dos modos de erros oscilatórios (erros de alta-freqüência), que são os modos significativos na avaliação do método *multigrid*.

1.3 Motivação

A solução de problemas de mecânica dos fluidos e transferência de calor através de métodos numéricos exige um alto custo computacional, pois, procurando-se diminuir drasticamente os erros de discretização, aumenta-se o número de elementos na malha. Estes problemas são da ordem de milhões de variáveis. Isto faz com que a matriz dos coeficientes seja bastante grande e esparsa, necessitando assim, de técnicas sofisticadas para tratar essas particularidades.

A eficiência do método *multigrid* não tem sido totalmente alcançada em aplicações realísticas da engenharia na área de Dinâmica dos Fluidos Computacional (DFC)¹. Uma importante razão para isto é que DFC, em geral, resolve problemas com perturbações singulares, fortes anisotropias, alta razão de aspecto, equações governantes que podem exibir um certo comportamento em uma parte do domínio e outro comportamento em outra parte do domínio, etc.

Quanto mais distinta a razão de aspecto (RA) está da unidade, ou seja, $0 < RA \ll 1$ ou $RA \gg 1$, mais se deteriora a razão de convergência do método *multigrid* (WESSELING e OOSTERLEE, 2001), podendo até mesmo ocorrer divergência (LARSSON et al., 2005). Este tipo de anisotropia é muito comum em problemas práticos de engenharia, como dito anteriormente, em problemas de camada limite, onde a razão de aspecto (RA) pode ser da

¹ Computational Fluid Dynamic (CFD).

ordem de 10^3 , 10^4 ou mais (WESSELING e OOSTERLEE, 2001); por isto a importância de se estudar algoritmos eficientes para razões de aspecto distintas da unidade.

Para poder entender melhor a degeneração do método *multigrid*, vejamos como o método MSG (Método de semi-engrossamento múltiplo), definido em Mulder (1989), se deteriora perante estas dificuldades. Veja tabela 1.2, que é uma adaptação da tabela 1 de Naik e van Rosendale (1993). Aqui $RA = \alpha/\gamma$ é a razão de aspecto da equação $\alpha u_{xx} + \gamma u_{yy} = f$ daquele artigo usando-se o *solver* MSG. Se $RA = 1$, o problema chama-se isotrópico; caso contrário, anisotrópico. Neste caso tem-se anisotropia de coeficientes, ou seja, anisotropia física, termo que será definido com mais detalhes na seção 3.4.

Tabela 1.2: Razão de convergência assintótica para o método MSG em várias malhas

	8X8	16X16	32X32
$\alpha/\gamma = 1$	0,07	0,09	0,10
$\alpha/\gamma = 10$	0,13	0,15	0,15
$\alpha/\gamma = 100$	0,16	0,19	0,19
$\alpha/\gamma = 1000$	0,16	0,19	0,21

Nota-se que quanto maior é a razão de aspecto da equação, maior será a razão de convergência assintótica. Como poderá ser visto no próximo capítulo, isto implica em um método iterativo mais lento.

Em Brandt (1998) e Wesseling e Oosterlee (2001) são descritas diversas deficiências e apontados vários desafios para o método *multigrid* geométrico, como: os sistemas de Equações Diferenciais Parciais (EDP) de Navier-Stokes e Euler, suavizadores (métodos de resolução de sistemas de equações algébricas com boas propriedades de suavização) para sistemas especiais (discretização h -elíptica, Euler e Navier-Stokes compressível não-conservativo, etc), operadores não-elípticos, ondas de choque, problemas com perturbações singulares, paralelização de algoritmos, baixa taxa de convergência induzida pela malha (grande razão de aspecto, expansão de malhas) ou mesmo problemas de camada limite que são problemas com as malhas fortemente alongadas.

Isto motiva a estudar-se as propriedades do método *multigrid* geométrico, tanto em sua fundamentação como em aplicações, para que se possam elaborar algoritmos *multigrid* mais eficientes para classes de problemas mais abrangentes. Para tanto, inicia-se o estudo com problemas unidimensionais (dois lineares e um não-linear) modelados por equações bastante

simples para se compreender o efeito que alguns parâmetros do método *multigrid* exercem sobre o tempo de CPU. Em seguida, continuam-se os estudos com um problema bidimensional linear modelado também por uma equação bastante simples no intuito de se verificar se o efeito de tais parâmetros estudados no caso unidimensional continua sendo válido nesta etapa. Finalmente estuda-se um problema bidimensional linear em um caso especial com malhas alongadas. O que se espera é que as propriedades encontradas nos estudos anteriores sejam estendidas para equações mais gerais.

1.4 Objetivos

Neste trabalho os seguintes parâmetros são investigados: razões de engrossamento, número de incógnitas, número de iterações internas (número de iterações do método numérico a fim de suavizar as componentes de erro), o número de níveis (número de malhas percorridas), os métodos de resolução de sistemas de equações algébricas (aqui chamados de *solvers*) Gauss-Seidel, MSI (SCHNEIDER e ZEDAN, 1981) e ADI (TANNEHILL et al., 1997) e os esquemas CS e FAS. São estudados também quatro algoritmos baseados no método *multigrid* para problemas anisotrópicos para diversas razões de aspecto: engrossamento padrão, engrossamento padrão seguido de semi-engrossamento, semi-engrossamento e semi-engrossamento seguido de engrossamento padrão.

Os objetivos desta tese são: verificar o efeito dos diversos parâmetros citados acima sobre o tempo de CPU para o *multigrid* geométrico e realizar comparações entre o esquema de correção (CS) e o esquema de aproximação completa (FAS) com ciclo V proposto em Wesseling (1992) e também comparações entre os diversos algoritmos para problemas anisotrópicos com a finalidade de se melhorar o desempenho do método *multigrid*, no que diz respeito à minimização do tempo de CPU, para problemas iguais ou similares aos estudados. Os resultados são comparados com os obtidos na bibliografia. São usados os operadores de restrição por injeção e prolongação por interpolação linear (no caso unidimensional) e bilinear (no caso bidimensional) encontrados em Briggs et al., 2000. Os modelos matemáticos considerados neste trabalho envolvem: as equações de difusão, de advecção-difusão (lineares) e Burgers (não-linear) unidimensionais e a equação de Laplace (linear) bidimensional. Elas representam problemas de transferência de calor e escoamento de fluido.

1.5 Revisão bibliográfica

Brandt (1977) trabalhou com o método *multigrid* geométrico em diversos problemas de transferência de calor e escoamento de fluidos, unidimensionais e bidimensionais, lineares e não-lineares. Em seu trabalho fez comparações com a razões de engrossamento $r = 2, 3$ e $3/2$. Para os problemas testados, Brandt (1977) concluiu que a razão $r = 2$ é a recomendável. Neste trabalho, Brandt (1977) também mostrou preferência pelo esquema CS em relação ao esquema FAS no caso de problemas lineares. Segundo Brandt, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS devido à transferência de informações a respeito do resíduo e da solução para as malhas mais grossas. Stüben (1999) desenvolveu um estudo com razões $r = 2$ e 4 em malhas não-estruturadas, para diversos problemas de transferência de calor, escoamento e eletromagnetismo, bidimensionais e tridimensionais, lineares e não-lineares. Em seu trabalho, a razão $r = 4$ se mostrou eficiente para problemas anisotrópicos (anisotropia dos coeficientes e anisotropia devido à malhas fortemente alongadas). Briggs et al. (2000) afirmaram que a razão $r = 2$ é uma prática universal e que $r \neq 2$ não traz vantagens. Moro (2004) trabalhou com as razões $r = 2$ e 4 em malhas estruturadas para problemas de difusão com termo fonte. Em seu trabalho a razão $r = 4$ apresentou tempo de CPU menor que a razão $r = 2$.

Em Ghia et al. (1982) foi desenvolvido um método *multigrid* implícito para problemas fortemente acoplados aplicado às equações de Navier-Stokes para escoamentos incompressíveis com alto número de Reynolds. Em Hutchinson e Raithby (1986) foi criado um método *multigrid* baseado no método de correção aditiva de Settari e Aziz (1973) e foi mostrado como este método pode ser usado para melhorar a razão de convergência em problemas bi e tridimensionais quando os coeficientes das equações algébricas tornam-se fortemente anisotrópicos (anisotropia física). Já o método de correções por blocos foi discutido em Kelker (1990) e Sathyamurthy e Patankar (1994) e em Karki et al. (1996) com algumas alterações para o caso de problemas de acoplamento, gerando o método *multigrid* com correções por bloco (*Block Corrections Multigrid Method*, BCMM). Uma extensão do método *multigrid* para malhas co-localizadas foi apresentada em Hortmann et al. (1990), e para malhas curvilíneas em Smith et al. (1993).

Outros desenvolvimentos e aplicações para o método *multigrid* foram realizados, como o acoplamento entre a técnica *multigrid* e a técnica de Decomposição de Domínio para geometrias complexas. Esta técnica foi apresentada em Perng e Street (1991) e em Lai e Przekwas (1996). Em Nishida e Satofuka (1992) foi desenvolvido um método *multigrid*

específico para as equações de Navier-Stokes na formulação vorticidade/corrente, usando uma variação do método de Runge-Kutta, conhecido como Runge-Kutta racional (Rational Runge-Kutta, RRK). Aceleração do método *multigrid* em problemas de turbulência foi tratada em Rubini et al. (1992). Em Lilek et al. (1997) o método *multigrid* foi usado para acelerar o método *Semi Implicit Linked Equations* (SIMPLE) em escoamentos incompressíveis. Uma variante do *Full-Multigrid/Full Approximation Scheme* (FMG-FAS) foi proposta em Yan e Thiele (1998) e Mesquita e de-Lemos (2004), que é usado para o caso não-linear, onde somente o resíduo é restrito à malha grossa, contrariamente ao FAS, que restringe tanto o resíduo como a solução.

Como a razão de convergência do método *multigrid*, baseado em suavizadores do tipo relaxação por pontos, falha ou se degenera nos casos anisotrópicos, então, outros suavizadores, como Jacobi por linhas, Gauss-Seidel por linhas, Jacobi por linhas com direções alternadas (ADLJ), etc, geralmente são necessários para obter-se boa razão de convergência. Em Sivaloganathan (1991) os métodos Gauss-Seidel simétrico (*Symmetric Coupled Gauss-Seidel*, SCGS) e correção de pressão pelo método SIMPLE (*SIMPLE Pressure Correction*, SPC) são incorporados e comparados como suavizadores no método *multigrid* com bloco implícito (*Block Implicit Multigrid Method*, BLIMM). Versões linha de SCGS são descritas e aplicadas em Oosterlee e Wesseling (1993). O SIMPLE é comparado com uma versão linha do SCGS em Sockol (1993) e mostrou-se superior na presença de malhas alongadas. Uma análise de suavização na presença de engrossamento direcional e o uso de pré-condicionadores foi feita em Pierce e Alonso (1998).

Estudos recentes para problemas anisotrópicos (anisotropia geométrica) têm focado para os algoritmos de semi-engrossamento (engrossamento em apenas uma das direções coordenadas). Em Mulder (1989) e em Naik e van Rosendale (1993) trocam-se o engrossamento padrão pelo semi-engrossamento múltiplo, que mostrou ser mais preciso. O semi-engrossamento direcional combinado com relaxação por linhas foi desenvolvido em Dendy et al. (1989). Em Radespiel e Swanson (1995) é apresentada uma variante de semi-engrossamento para as equações de Navier-Stokes. Um comparativo entre dois métodos *multigrid* robustos para as equações de Navier-Stokes compressível 3D (suavizadores alternados de planos combinado com engrossamento padrão e suavizadores implícitos de planos combinado com semi-engrossamento) foi feito em Montero et al. (2001). Em Larsson et al. (2005) foi desenvolvido um semi-engrossamento condicional para a equação de Poisson em malhas anisotrópicas, resultando em um algoritmo *multigrid* eficiente e simples de implementar.

Em Zhang (2002), são abordados problemas anisotrópicos acoplando-se a técnica de semi-engrossamento seguida de engrossamento padrão (engrossamento em ambas as direções) intitulado como “*partial semicoarsening*”. Comparações foram feitas para a equação de Poisson bidimensional com a técnica que se chama de engrossamento completo (*full coarsening*), conhecido também como engrossamento padrão. Zhang (2002) comparou estas técnicas para as razões de aspecto $RA = 2, 4, 8$ e 16 (razões de aspecto modestas). Zhang constatou que o algoritmo *partial semicoarsening* com os suavizadores *red-black* e *four-color* Gauss-Seidel são eficientes para as anisotropias estudadas.

Refinamento e engrossamento são empregados simultaneamente para fins adaptativos em malhas anisotrópicas por Ham et al. (2002) e em Ferm e Lötstedt (2003).

1.6 Delineamento deste texto

Esta tese está organizada da seguinte forma:

- No capítulo 2 é apresentada uma visão geral dos métodos iterativos básicos, uma análise de erros de Fourier e os princípios básicos do método *multigrid*;
- No capítulo 3 é vista a filosofia do método *multigrid*, algoritmos dos esquemas CS e FAS, os operadores de transferência entre malhas, juntamente com os casos dos problemas anisotrópicos (anisotropia física e geométrica);
- No capítulo 4 são apresentados os modelos matemáticos e numéricos, resultados com o esquema CS e FAS e comparações para os problemas unidimensionais lineares: equação de difusão e de advecção-difusão;
- No capítulo 5 é apresentado o modelo matemático e numérico, resultados com o esquema CS e FAS e comparações para o problema unidimensional não-linear: equação de Burgers;
- No capítulo 6 é apresentado o modelo matemático e numérico, resultados com o esquema CS e FAS e comparações para o problema isotrópico bidimensional linear: equação de Laplace;
- No capítulo 7 são apresentados resultados com o esquema CS e comparações entre algoritmos para o problema anisotrópico (anisotropia geométrica) bidimensional linear: equação de Laplace;
- No capítulo 8 é apresentada a conclusão, contribuições, recomendações e extrapolações deste trabalho de Tese e propostas de trabalhos futuros;

- No apêndice A estão alguns exemplos de métodos iterativos básicos e outros *solvers* utilizados neste trabalho;
- No apêndice B estão as soluções analíticas para os problemas unidimensionais, assim como para o problema bidimensional;
- No apêndice C estão os testes adicionais para o critério de convergência;
- No apêndice D estão os artigos com resultados deste trabalho publicados em anais de Congressos Nacionais e Internacionais.

2 Fundamentação

Neste capítulo é apresentada uma fundamentação teórica para o desenvolvimento do método *multigrid* que será abordado no capítulo 3. Estuda-se o princípio fundamental dos métodos iterativos básicos e faz-se uma análise de erros e de convergência. Finalmente, faz-se um estudo sobre os princípios básicos do método *multigrid*.

2.1 Métodos iterativos básicos

Esta seção introduz as idéias básicas sobre os métodos iterativos que serão utilizados no decorrer deste trabalho.

Suponha que a discretização das equações diferenciais (ordinárias ou parciais) a serem resolvidas conduz ao seguinte sistema de equações algébricas

$$Ax = b, \quad (2.1)$$

onde A é uma matriz $N \times N$, x e $b \in R^N$, com x sendo o vetor incógnita e b o vetor independente.

Caso o sistema de equações algébricas seja não-linear, adota-se a seguinte notação

$$A(x) = b. \quad (2.1a)$$

Em ambos os casos, seja a matriz A dividida na forma

$$A = M - N', \quad (2.2)$$

com M não-singular.

Para a solução da equação 2.1, tem-se o seguinte método que é chamado de método iterativo básico

$$Mx^{(m+1)} = N'x^{(m)} + b, \quad (2.3)$$

onde o superescrito de x é a representação dos níveis iterativos.

Considera-se o método da seguinte forma

$$x^{(m+1)} = Sx^{(m)} + M^{-1}b, \quad (2.4)$$

onde, $S = M^{-1}N'$, $N' = M - A$, e a matriz S é chamada de matriz de iteração do método dado pela equação 2.4.

Os métodos iterativos básicos podem ser amortecidos por uma pequena modificação da equação 2.4, como segue

$$x^* = Sx^{(m)} + M^{-1}b,$$

e

$$x^{(m+1)} = \omega x^* + (1 - \omega)x^{(m)}. \quad (2.5)$$

Substituindo x^* na equação 2.5, tem-se

$$x^{(m+1)} = S^*x^{(m)} + \omega M^{-1}b, \quad (2.6)$$

onde, $S^* = \omega S + (1 - \omega)I$, com I sendo a matriz identidade e $\omega \in R$.

Os autovalores da matriz de iteração não-amortecida S e os da matriz de iteração amortecida S^* são relacionados por

$$\lambda(S^*) = \omega\lambda(S) + 1 - \omega. \quad (2.7)$$

Exemplos de métodos iterativos básicos e outros métodos para a resolução se sistemas de equações (*solvers*) que serão tratados neste trabalho, podem ser vistos no apêndice A desta Tese.

2.2 Análise de convergência

Deve-se procurar dentre os possíveis métodos iterativos básicos, aqueles que são convergentes, noção que será vista nesta seção.

Seja o sistema $Ax = b$ e o método iterativo básico $Mx = N'x + b$. Somente nesta seção denota-se x como sendo a solução exata do sistema e \bar{x} a solução aproximada, sendo que a solução aproximada pode ainda ser representada em função de suas iterações, ou seja, $x^{(m)}$. Desta forma, o erro é dado por $e = x - \bar{x}$, ou por $e^{(m)} = x - x^{(m)}$, e sua magnitude pode ser medida pelas normas vetoriais padrões (BURDEN e FAIRES, 1997), como a norma euclidiana, também chamada de norma-2

$$\|e\|_2 = \left[\sum_{j=1}^N e_j^2 \right]^{\frac{1}{2}},$$

ou a norma do máximo (norma-infinito)

$$\|e\|_\infty = \max_{1 \leq j \leq N} |e_j|,$$

onde e_j é a j -ésima coordenada do vetor e .

O erro, definido em função de suas iterações, $e^{(m)} = x - x^{(m)}$, satisfaz

$$e^{(m+1)} = Se^{(m)}. \quad (2.8)$$

Nota-se que o erro é tão inacessível quanto à solução exata. Entretanto, uma medida da proximidade de x e \bar{x} (ou $x^{(m)}$, em função das iterações) é o resíduo, definido por

$$R = b - A\bar{x}, \quad (2.9)$$

ou ainda,

$$R^{(m)} = b - Ax^{(m)}. \quad (2.10)$$

No caso não-linear (veja equação 2.1a) tem-se o resíduo dado por

$$R^{(m)} = b - A(x^{(m)}). \quad (2.10a)$$

Uma importante relação entre erro e resíduo é dada pela expressão (BRIGGS et al., 2000)

$$\frac{1}{\text{cond}(A)} \frac{\|R^{(m)}\|}{\|b\|} \leq \frac{\|e^{(m)}\|}{\|x^{(m)}\|} \leq \text{cond}(A) \frac{\|R^{(m)}\|}{\|b\|}, \quad (2.11)$$

com $\text{cond}(A) = \|A\| \|A^{-1}\|$ sendo o número de condicionamento da matriz A . Por definição $\text{cond}(A) \geq 1$ (BURDEN e FAIRES, 1997).

Nota-se ainda que pela equação 2.11 $R^{(m)} \approx 0$ implica que $e^{(m)} \approx 0$ somente se $\text{cond}(A)$ é pequeno, ou seja, próximo da unidade, $\text{cond}(A) \approx 1$. Neste caso diz-se que a matriz A é bem-condicionada.

Do sistema $Ax = b$ e das definições de resíduo $R^{(m)}$ e erro $e^{(m)}$, pode-se obter a seguinte relação que é chamada de “equação residual”

$$Ae^{(m)} = R^{(m)}. \quad (2.12)$$

Dado o sistema de equações algébricas $Ax = b$, com solução exata x e uma solução aproximada na m -ésima iteração $x^{(m)}$, cujo resíduo é dado por $R^{(m)} = b - Ax^{(m)}$, pode-se melhorar a aproximação $x^{(m)}$ resolvendo-se a equação residual, dada pela equação 2.12, para $e^{(m)}$ e então calcular a nova aproximação usando a definição de erro $x = x^{(m)} + e^{(m)}$. A este processo dá-se o nome de “correção residual” ou “refinamento iterativo”.

No caso não-linear (veja a equação 2.1a juntamente com a equação 2.10a) a equação residual é dada por

$$A(x) - A(x^{(m)}) = R^{(m)}. \quad (2.12a)$$

Nota-se que $A(e^{(m)}) \neq R^{(m)}$, contrariamente ao que se viu no caso linear, onde $Ae^{(m)} = R^{(m)}$. Em geral, no caso não-linear $A(e^{(m)}) = A(x - x^{(m)}) \neq A(x) - A(x^{(m)}) = R^{(m)}$.

Da expressão dada pela equação 2.8 tem-se que $e^{(m)} = S^m e^{(0)}$, onde o superescrito de e é uma representação do nível iterativo e o superescrito de S é uma representação da potência matricial, onde vale

$$\|e^{(m)}\| \leq \|S^m\| \|e^{(0)}\| = \|S\|^m \|e^{(0)}\| \quad (2.13)$$

para qualquer norma $\|\cdot\|$, tal que

$$\|S\| = \sup_{x \neq 0} \frac{\|Sx\|}{\|x\|} \quad (2.14)$$

é a norma matricial induzida pela norma vetorial. Neste caso $\|S\|$ é chamada de “número de contração” do método iterativo $x^{(m+1)} = Sx^{(m)} + M^{-1}b$.

Definição 2.1: O método iterativo $x^{(m+1)} = Sx^{(m)} + M^{-1}b$ é chamado de convergente se $\lim_{m \rightarrow \infty} \|S\|^m = 0$, com $S = M^{-1}N$.

Definição 2.2: O raio espectral da matriz A é dado por $\rho(A) = \max |\lambda(A)|$, onde $\lambda(A)$ são os autovalores de A .

Teorema 2.1: O método iterativo $x^{(m+1)} = Sx^{(m)} + M^{-1}b$ é convergente, se e somente se, $\rho(S) < 1$.

Prova: Veja em Burden e Faires (1997) nas páginas 441 a 451. \square

O raio espectral $\rho(S)$ serve também para fornecer aproximadamente quantas iterações são necessárias para reduzir o erro pelo fator de 10^{-d} , onde d é o número de casas decimais considerados nos cálculos.

Seja m (número de iterações) o menor inteiro que satisfaz

$$\frac{\|e^{(m)}\|}{\|e^{(0)}\|} \leq 10^{-d}. \quad (2.15)$$

Prova-se em Briggs et al. (2000) que esta condição será aproximadamente satisfeita se

$$[\rho(S)]^m \leq 10^{-d}, \quad (2.16)$$

onde $\rho(S)$ é o raio espectral da matriz de iteração S .

Resolvendo-se a desigualdade dada pela equação 2.16 para m , tem-se

$$m \geq -\frac{d}{\log_{10}[\rho(S)]}. \quad (2.17)$$

Pode-se ver que

$$\rho(S) \approx 1 \Rightarrow \log_{10} \rho(S) \approx 0 \Rightarrow m \approx \infty.$$

Neste caso a razão de convergência decresce. Pode-se ver também que

$$0 < \rho(S) \approx 0 \Rightarrow \log_{10} \rho(S) \approx -\infty \Rightarrow m \approx 0.$$

Neste caso tem-se uma alta razão de convergência.

2.3 Análise de erros de Fourier

É importante conhecer o desempenho dos métodos iterativos básicos estudados nas seções anteriores. Por isto estuda-se a análise de erros.

Seja $Au = f$ um sistema de equações algébricas a ser resolvido. Daqui em diante usa-se u para a solução exata e v para uma solução aproximada.

Da definição 2.1, nota-se que o estudo de convergência de um método iterativo se reduz ao estudo da matriz de iteração $S = M^{-1}N'$. Pode-se notar que esta matriz depende apenas da forma como ocorreu o desmembramento da matriz A . Portanto, ao estudar-se erros do sistema $Au = f$, é suficiente trabalhar com o sistema linear homogêneo $Au = 0$ e usar estimativas iniciais arbitrárias para começar o esquema iterativo, aqui também chamado de esquema de relaxação. Neste caso tem-se a vantagem da solução exata $u = 0$ ser conhecida e o erro da aproximação v ser simplesmente $-v$.

Aplicam-se várias iterações a este sistema linear de equações com um tipo especial de estimativa inicial consistindo de vetores do tipo

$$\mathbf{v}_k = \left[\operatorname{sen}\left(\frac{jk\pi}{N}\right) \right]_{j=1,\dots,N}, \quad 0 \leq j \leq N, 1 \leq k \leq N-1. \quad (2.18)$$

Estes vetores são chamados de modos de Fourier. Por este motivo este tipo de análise de erros é conhecida como análise de erros de Fourier. Para cada k fixo, o índice j denota a j -ésima componente do vetor \mathbf{v}_k .

Definição 2.3: O inteiro k é chamado de número de ondas ou freqüência. Ele indica o número de “meios senos” que constituem o vetor \mathbf{v}_k no domínio do problema.

A figura 2.1, extraída de Briggs et al. (2000), ilustra algumas estimativas iniciais, como \mathbf{v}_1 , \mathbf{v}_3 e \mathbf{v}_6 . Nota-se que valores pequenos de k correspondem a ondas longas e suaves, enquanto que valores grandes de k correspondem a ondas curtas e oscilatórias.

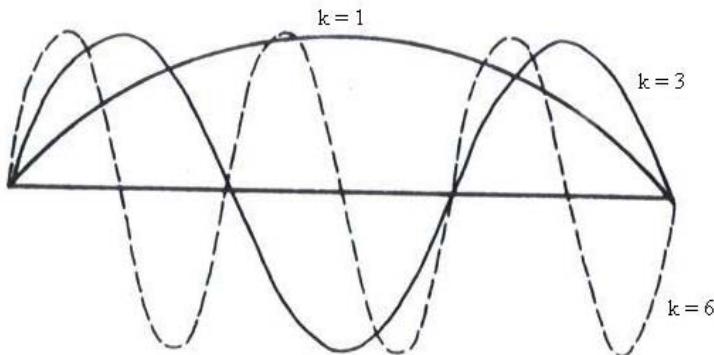


Figura 2.1: Modos de Fourier com $k = 1, 3, 6$ (BRIGGS et al., 2000)

Seja $e^{(0)}$ o erro na estimativa inicial usada em um método numérico. Então é possível representar $e^{(0)}$ usando os autovetores de A (que poder-se-á constatar pela equação 2.28 adiante, ser igual aos autovetores S_ω) na forma

$$e^{(0)} = \sum_{k=1}^{N-1} c_k w_k , \quad (2.19)$$

onde $c_k \in R$. Após m passos de relaxação e usando a expansão dada pela equação 2.19, tem-se

$$e^{(m)} = S_\omega^m e^{(0)} = \sum_{k=1}^{N-1} c_k S_\omega^m w_k = \sum_{k=1}^{N-1} c_k \lambda_\omega^m w_k . \quad (2.20)$$

Pode-se concluir da equação 2.20 que tal método quando aplicado a um modo de Fourier, a iteração muda a amplitude daquele modo, mas não o converte em um outro modo.

Nota-se que o k -ésimo modo consiste de $k/2$ ondas de senos completos que tem o comprimento de onda de $l = 2Nh/k = 2/k$ (considerando que o intervalo tem comprimento $L = 1$ e $h = L/N$). O modo $k = N/2$ tem o comprimento de onda $l = 4h$ e o modo $k = N - 1$ tem o comprimento próximo de $l = 2h$ (BRIGGS et al., 2000). Ondas com número de onda maior do que N (com comprimento de onda menor do que $2h$) não podem ser representadas na malha.

Definição 2.4: Os modos de Fourier localizados na metade inferior do espectro, com $1 \leq k < N/2$, são chamados de modos de Fourier de baixa freqüência ou modos suaves. Os modos de Fourier localizados na metade superior do espectro, com $N/2 \leq k \leq N - 1$, são chamados de modos de Fourier de alta freqüência ou modos oscilatórios.

2.4 Aplicação da análise de erros de Fourier na equação de Poisson 1D

Obtém-se uma boa percepção no estudo da análise de erros de Fourier, se é tomado primeiramente o caso unidimensional linear, como é o caso da equação dada por

$$\begin{cases} -u_{xx} = f, & 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases}, \quad (2.21)$$

onde u_{xx} representa a segunda deriva da função u em relação à variável espacial x .

Esta equação é conhecida como equação de Poisson unidimensional. Sua forma discretizada pelo método das diferenças finitas, usando diferenças centradas (CDS), será dada por

$$\begin{cases} -v_{j-1} + 2v_j - v_{j+1} = h^2 f_j, & 1 \leq j \leq N - 1 \\ v_0 = v_N = 0 \end{cases} \quad (2.22)$$

e o caso homogêneo dado por

$$\begin{cases} -v_{j-1} + 2v_j - v_{j+1} = 0, & 1 \leq j \leq N - 1 \\ v_0 = v_N = 0 \end{cases}, \quad (2.23)$$

onde N , nesta seção, é o número de incógnitas (elementos) e v_j é uma aproximação (solução numérica) da solução exata $u(x_j)$.

Considere o método de Jacobi ponderado, com fator de ponderação ω , aplicado ao problema dado pela equação 2.23. Temos $S_\omega = (1 - \omega)I + \omega S_J$ com S_J sendo a matriz de iteração do método de Jacobi (veja apêndice A). Assim, com um pouco de algebrismo, tem-se:

$$S_\omega = I - \frac{\omega}{2}A, \quad (2.24)$$

onde

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}_{(N-1) \times (N-1)}. \quad (2.25)$$

Os autovalores de S_ω e A são relacionados pela expressão

$$\lambda(S_\omega) = 1 - \left(\frac{\omega}{2}\right)\lambda(A). \quad (2.26)$$

Procurando-se os autovalores da matriz original A , encontra-se (BRIGGS et al., 2000)

$$\lambda_k(A) = 4 \operatorname{sen}^2\left(\frac{k\pi}{2N}\right), \quad 1 \leq k \leq N-1. \quad (2.27)$$

Os autovetores de A são dados por

$$w_{k,j} = \operatorname{sen}\left(\frac{jk\pi}{N}\right), \quad 1 \leq k \leq N-1, \quad 0 \leq j \leq N. \quad (2.28)$$

Pode-se perceber que os autovetores de A são simplesmente os modos de Fourier dados pela equação 2.18. Dessa forma, pode-se encontrar os autovalores de S_ω

$$\lambda_k(S_\omega) = 1 - 2\omega \operatorname{sen}^2\left(\frac{k\pi}{2N}\right), \quad 1 \leq k \leq N-1, \quad (2.29)$$

enquanto que os autovetores de S_ω são os mesmos autovetores de A .

Tem-se que $0 < \omega \leq 1 \Rightarrow |\lambda_k(S_\omega)| < 1$ e, neste caso, o método de Jacobi ponderado é convergente. Para $0 < \omega \leq 1$ (que garante convergência) e $k=1$, tem-se uma situação específica para a equação 2.29 que será descrita a seguir

$$\lambda_1 = 1 - 2\omega \operatorname{sen}^2\left(\frac{\pi}{2N}\right) = 1 - 2\omega \operatorname{sen}^2\left(\frac{\pi h}{2}\right) \approx 1 - \frac{\omega \pi^2 h^2}{2}. \quad (2.30)$$

Pode-se perceber que λ_1 é o autovalor associado ao modo de Fourier mais suave $k=1$ e que ele estará sempre próximo da unidade. Quanto menor o espaçamento h da malha, mais próximo λ_1 estará da unidade. Qualquer tentativa para melhorar a solução (decréscimo no espaçamento h , por exemplo), irá piorar a convergência dos modos suaves do erro. Nenhum valor de ω amortece as componentes de erros suaves de uma maneira satisfatória.

A figura 2.2 a seguir, foi adaptada de Briggs et al. (2000). Ela mostra uma aplicação de 100 iterações do método de Jacobi ponderado (a) para $\omega = 2/3$ (fator de ponderação ótimo, BRIGGS et al., 2000), e o método de Gauss-Seidel (b), ao problema modelo dado pela equação 2.23 numa malha com $N = 64$ elementos e com as estimativas iniciais para \mathbf{v}_1 , \mathbf{v}_3 e \mathbf{v}_6 , que correspondem aos modos de Fourier para $k = 1, 3$ e 6 , respectivamente. Esta figura mostra a norma infinito ($\|\cdot\|_\infty$) do erro versus o número de iterações.

A figura 2.3, extraída de Briggs et al. (2000), mostra o método de Jacobi ponderado com $\omega = 2/3$ aplicado ao problema dado pela equação 2.23 com 100 iterações numa malha com $N = 64$ e com a estimativa inicial $(\mathbf{v}_1 + \mathbf{v}_6 + \mathbf{v}_{32})/3$, ou seja, uma estimativa inicial composta de modos de Fourier de baixa, média e alta freqüências. A figura 2.3 mostra o erro na norma $\|\cdot\|_\infty$. Nota-se que o erro decresce rapidamente com as primeiras iterações, depois o degréscimo é mais lento.

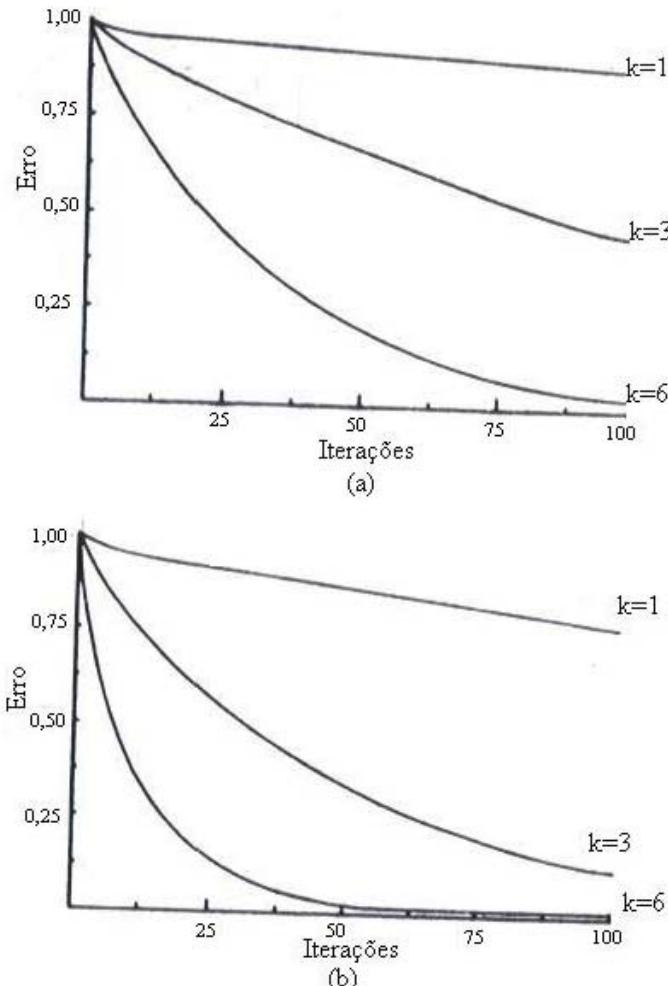


Figura 2.2: (a) Jacobi ponderado com $\omega = 2/3$, (b) Gauss-Seidel regular aplicados ao problema unidimensional com $N = 64$, estimativas iniciais para $k = 1, 3, 6$ e para 100 iterações (BRIGGS et al., 2000)

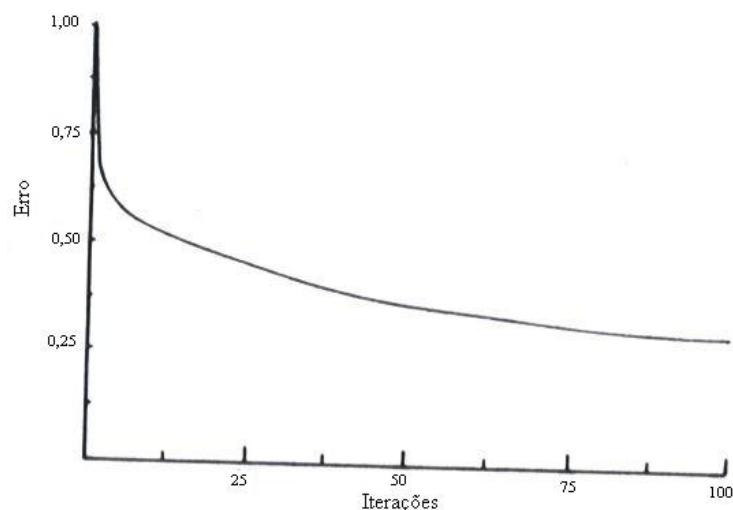


Figura 2.3: Jacobi ponderado com $\omega = 2/3$ aplicado ao problema unidimensional com $N = 64$, estimativa inicial $(v_1+v_6+v_{32})/3$ e para 100 iterações (BRIGGS et al., 2000)

Na figura 2.3, o decréscimo inicial é correspondente à rápida eliminação dos modos de erros de Fourier de alta freqüência, ou modos oscilatórios. O lento decréscimo que se sucede é devido à presença de modos de erros de Fourier de baixa freqüência, ou modos suaves (BRIGGS et al., 2000).

A figura 2.4, adaptada de Briggs et al. (2000), mostra a seletividade da propriedade de amortecimento. Usa-se a estimativa inicial $(v_2 + v_{16})/2$, ou seja, uma estimativa inicial composta de um modo suave e o outro oscilatório. Após alguns passos de relaxação (10 passos) pelo método de Jacobi ponderado com $\omega = 2/3$, os modos de alta freqüência são rapidamente eliminados. Entretanto, os modos de baixa freqüência persistem.

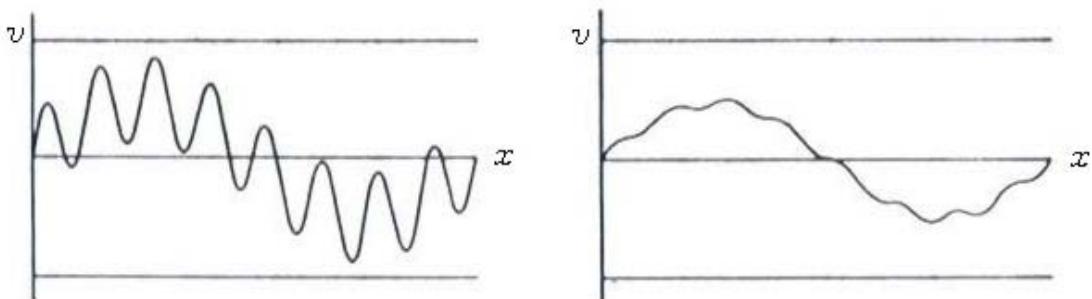


Figura 2.4: Jacobi ponderado com $\omega = 2/3$ aplicado ao problema unidimensional com $N = 64$, com estimativa inicial $(v_2+v_{16})/2$ antes (esquerda) e depois (direita) de 10 iterações (BRIGGS et al., 2000)

Definição 2.5: A propriedade de eliminar os modos oscilatórios e deixar apenas modos suaves chama-se propriedade de suavização.

A seguir procura-se verificar se todos os métodos iterativos possuem a propriedade de suavização dada pela definição 2.5.

A figura 2.5, extraída de Briggs et al. (2000), mostra o desempenho dos métodos (Jacobi e Jacobi ponderado) conforme a variação do número de ondas k . O gráfico mostra o número de ondas *versus* o número de iterações exigido para reduzir a norma do erro inicial por um fator de 10^2 . Na figura 2.5(a) tem-se o método de Jacobi ponderado com $\omega = 1$, ou seja, o método de Jacobi regular e na figura 2.5(b) tem-se o método de Jacobi ponderado com $\omega = 2/3$, ou seja, o método de Jacobi com o fator de ponderação ótimo (BRIGGS et al., 2000).

Como pôde-se constatar, o método de Jacobi regular não possui a propriedade de suavização. Outro método clássico que não possui tal propriedade é o método *Successive Over Relaxation* (SOR) (HACKBUSCH, 1985; página 52).

Uma análise similar ao que foi realizada nesta seção, fazendo-se uso do método de Gauss-Seidel para o mesmo modelo dado pela equação 2.21, pode ser encontrada Briggs et al. (2000).

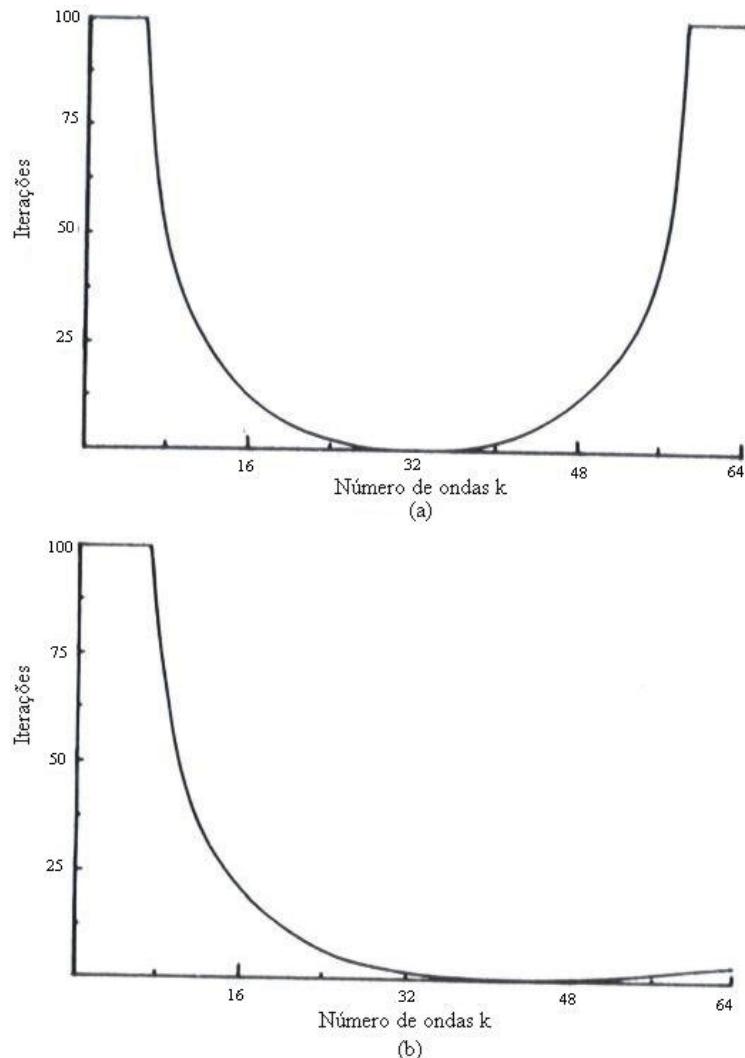


Figura 2.5: (a) Jacobi regular ($\omega = 1$) e (b) Jacobi ponderado com $\omega = 2/3$ aplicados ao problema unidimensional com $N = 64$, estimativas iniciais com $1 \leq k \leq 63$, com norma do erro reduzida pelo fator de 100 (BRIGGS et al., 2000)

2.5 Princípios fundamentais do método *multigrid*

Nesta seção assume-se que um método de relaxação tenha sido aplicado nas componentes de erros em uma dada malha fina até ficarem somente componentes de erro suave. Esta malha fina será denotada por Ω^h , o que significa que cada elemento dessa malha possui tamanho h ; e a malha imediatamente mais grossa (malha com um número menor de

elementos) será denotada por Ω^h , ou seja, cada elemento possui tamanho H , com $H > h$. Nesta seção será utilizada a razão de engrossamento padrão $r = 2$, ou seja, $H = 2h$.

Na figura 2.6, obtida de Briggs et al. (2000), observa-se como as componentes de erro suave apresentam-se em uma malha fina Ω^h e em uma malha mais grossa Ω^{2h} . Na parte superior da figura 2.6 tem-se o modo de Fourier na malha fina Ω^h e na inferior, o mesmo modo de Fourier projetado na malha imediatamente mais grossa Ω^{2h} .

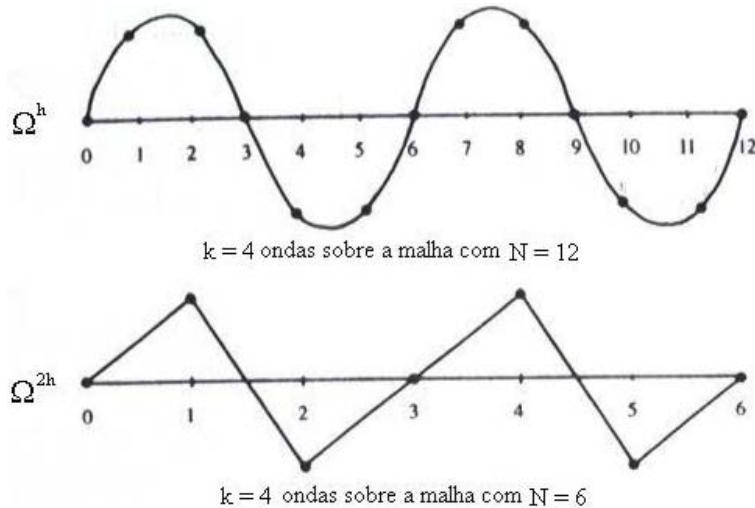


Figura 2.6: Número de ondas $k = 4$ sobre uma malha fina com $N = 12$ e sobre uma malha grossa com $N = 6$
(BRIGGS et al., 2000)

O modo de Fourier antes de ser projetado da malha Ω^h para a malha Ω^{2h} , tem $N = 12$ e $k = 4$, portanto, pela definição 2.4, tem-se um modo suave. Por outro lado, o modo de Fourier após ser projetado da malha Ω^h para a malha Ω^{2h} , tem $N = 6$ para o mesmo $k = 4$, portanto, pela definição 2.4, tem-se um modo oscilatório. Pode-se notar que modos suaves em Ω^h tornam-se mais oscilatórios em Ω^{2h} .

Seja v_k um modo de Fourier com o número de ondas k , com $1 \leq k < N - 1$ (veja definição 2.3). Deseja-se verificar se o número de ondas k do modo v_k continua sendo o mesmo após sua projeção da malha fina Ω^h para a malha grossa Ω^{2h} .

Para responder a esta questão, considere-se agora o k -ésimo modo na malha fina avaliado nos pontos pares desta malha. As componentes dos modos suaves (em relação à malha), ou seja, $1 \leq k < N/2$, podem ser escritas como

$$w_{k,2j}^h = \operatorname{sen}\left(\frac{2jk\pi}{N}\right) = \operatorname{sen}\left(\frac{jk\pi}{n/2}\right) = w_{k,j}^{2h}, \quad 1 \leq k < \frac{N}{2}. \quad (2.31)$$

Pode-se observar que o k -ésimo modo em Ω^h torna-se o k -ésimo modo em Ω^{2h} , ou seja, o número de ondas k em Ω^{2h} continua sendo o mesmo k de Ω^h , mas ao passar da malha fina Ω^h para a malha imediatamente mais grossa Ω^{2h} , o modo tornou-se mais oscilatório (em relação à malha). Isto sugere que quando um esquema de relaxação começa a tornar-se lento, sinalizando a predominância de modos suaves de erro (veja a figura 2.3), é recomendável transferir o problema de relaxação para a malha grossa, pois lá os modos de erros suaves se apresentarão mais oscilatórios, e o processo de relaxação será mais eficiente (WESSELING, 1992 e TROTENBERG et al. 2001).

Pode-se mostrar com um pouco de algebrismo que para $k = N/2$ o modo em Ω^h torna-se nulo em Ω^{2h} e para os modos oscilatórios, ou seja, $N/2 < k < N - 1$, o k -ésimo modo em Ω^h torna-se o $(N - k)$ -ésimo modo em Ω^{2h} . Portanto, os modos oscilatórios de Ω^h são os modos suaves de Ω^{2h} (BRIGGS et al., 2000).

No próximo capítulo detalha-se mais o método *multigrid*, com seus algoritmos, operadores de transferência entre malhas, etc.

2.6 Resumo do capítulo 2

Neste capítulo apresentou-se uma fundamentação teórica para o método *multigrid*. Nesta fundamentação revisou-se a teoria básica dos métodos iterativos básicos com uma breve abordagem da análise de convergência. Nesta análise detectou-se, por exemplo, a importância do raio espectral para a determinação, não só da convergência, mas também da razão de convergência. Em seguida foi feita uma análise de erros de Fourier, onde foram definidos os modos de Fourier e a ação dos suavizadores sobre estes modos. Nesta análise detectou-se que os modos oscilatórios são mais sensíveis às iterações dos suavizadores do que os modos suaves. Uma aplicação bastante simples dessa análise foi feita no caso da equação de Poisson unidimensional. Finalmente, estudaram-se também os princípios fundamentais do método *multigrid*, onde notou-se que os modos de Fourier suaves nas malhas mais finas tornam-se modos mais oscilatórios nas malhas mais grossas.

3 Método *multigrid*

A solução de problemas de mecânica dos fluidos e transferência de calor através de métodos numéricos são problemas da ordem de milhões de variáveis necessitando de técnicas sofisticadas para tratar esses problemas.

Neste capítulo estuda-se a técnica conhecida como método *multigrid*, que é uma técnica que leva informações do problema para diversas malhas, desde as mais finas até as mais grossas. Dessa forma, o método *multigrid* passa a ser eficiente para todas as componentes de erro (modos suaves ou de baixa freqüência nas malhas mais grossas e modos oscilatórios ou de alta freqüência nas malhas mais finas) acelerando a solução de sistemas de equações envolvidos nos problemas.

Usa-se inicialmente uma razão de engrossamento padrão $r = 2$, ou seja, o caso onde a malha grossa tem o dobro do espaçamento da malha fina, mas veremos mais adiante que outros tipos de engrossamentos também são possíveis, ou seja, $r = 3$, $r = 4$ e $r = 5$.

Esta seção está dividida da seguinte forma: primeiramente estudam-se os principais algoritmos do método *multigrid*. Em seguida caracterizam-se os processos de restrição e prolongação. Finalmente estuda-se o caso especial de *multigrid* para problemas anisotrópicos (problemas onde o acoplamento entre os pontos vizinhos torna-se muito forte em alguma das direções ou quando a discretização é baseada em malhas com razão de aspecto bem distinta da unidade), caso onde aumenta-se a necessidade da utilização do método *multigrid*.

3.1 Algoritmos

Nesta seção estuda-se um algoritmo básico para duas malhas que englobe os casos linear e não-linear, quaisquer razões de engrossamento, várias malhas, assim como os diversos tipos de ciclos (ordem na qual as malhas são visitadas), como o ciclo V ou o ciclo W.

Seja dada a seqüência $\Lambda = \{\Omega^L : L = 1, 2, \dots, K\}$ de malhas crescentemente mais grossas, com $\Omega^L = \Omega^{2^{(L-1)}h}$ para o caso de $r = 2$. Ou seja, $\Omega^1 = \Omega^h$ é a malha mais fina, $\Omega^2 = \Omega^{2h}$ é a malha imediatamente mais grossa, $\Omega^3 = \Omega^{4h}$ é a outra malha imediatamente mais grossa, até $\Omega^K = \Omega^{2^{(K-1)}h}$, que é a malha mais grossa possível. O número L , tal que $1 \leq L \leq K$ é chamado de número de níveis de malha.

Neste trabalho a malha mais grossa para problemas unidimensionais será sempre composta por dois elementos, ou seja, um nó interno e dois pontos de fronteira. Para problemas bidimensionais será sempre composta por quatro elementos, ou seja, dois elementos em cada direção coordenada, portanto, um nó interno e três pontos de fronteira para cada face (norte, sul, leste e oeste), conforme figura 3.1.

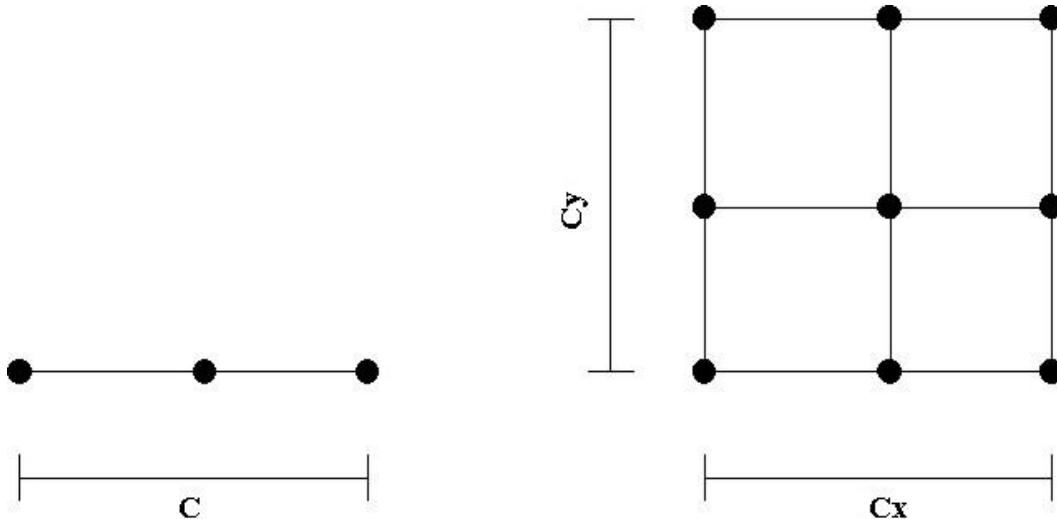


Figura 3.1: Malhas grossas uni e bidimensional

Seja U^k , com $U^k : \Omega^k \rightarrow R$, o conjunto de funções de malha fina, ou seja, o conjunto das funções com valores reais definidos nos pontos da malha Ω^k com $u^k \in U^k$. De forma análoga define-se $U^{k+1} : \Omega^{k+1} \rightarrow R$ o conjunto de funções de malha grossa.

Pode-se provar (WESSELING, 1992) a existência dos operadores $I_k^{k+1} : U^k \rightarrow U^{k+1}$ (operador de restrição) e $I_{k+1}^k : U^{k+1} \rightarrow U^k$ (operador de prolongação). Tais operadores transferem informações da malha fina para a malha imediatamente mais grossa e vice-versa, respectivamente.

Seja o problema a ser resolvido em Ω^k , denotado por

$$A^k(u^k) = f^k, \quad (3.1)$$

onde o operador A^k pode ser linear ou não. Neste caso A^k representa o operador A no k -ésimo nível de malha, ou seja, na malha Ω^k .

Por questões didáticas, assume-se: $r = 2$ e duas malhas Ω^h e Ω^{2h} . Seja o problema $A^h(u^h) = f^h$ a ser resolvido na malha fina Ω^h . Suponha que A^h seja um operador não-linear.

A tabela 3.1 a seguir nos fornece o algoritmo não-linear que é denominado de esquema de aproximação completa (FAS).

Tabela 3.1: Esquema FAS para duas malhas (adaptado de Briggs et al., 2000)

Esquema de aproximação completa para duas malhas	
	DG(v_0, v, f, h)
	Início
1.	Iterar $A^h(u^h) = f^h$ η_1 vezes com estimativa inicial v_0^h ;
2.	Calcular $R^h = f^h - A^h(v^h)$;
3.	Escolher $v_0^{2h} = I_h^{2h}v^h$;
4.	Obter $R^{2h} = I_h^{2h}R^h$;
5.	Definir $f^{2h} = A^{2h}(v_0^{2h}) + R^{2h}$;
6.	Resolver $A^{2h}(u^{2h}) = f^{2h}$ com estimativa inicial v_0^{2h} ;
7.	Calcular $e^{2h} = v^{2h} - v_0^{2h}$;
8.	Obter $e^h = I_{2h}^h e^{2h}$;
9.	Corrigir $v^h \leftarrow v^h + e^h$;
10.	Iterar $A^h(u^h) = f^h$ η_2 vezes com estimativa inicial v^h .
	Fim de DG(v_0, v, f, h)

O algoritmo descrito na tabela 3.1 será detalhado a seguir. O passo (1) representa η_1 iterações de suavização (pré-suavização) na malha fina Ω^h com uma estimativa inicial v_0^h a fim de se obter uma aproximação v^h . No passo (2) o resíduo R^h é calculado na malha fina Ω^h . Em (3) é escolhida uma estimativa inicial v_0^{2h} para a malha grossa Ω^{2h} . Neste algoritmo, v_0^{2h} é a restrição da solução aproximada v^h da malha fina Ω^h para a malha grossa Ω^{2h} , ou seja, v_0^{2h} é o vetor resultante da aplicação do operador de restrição I_h^{2h} sobre o vetor v^h (solução na malha Ω^h). O passo (4) representa a obtenção do resíduo na malha grossa Ω^{2h} pelo processo de restrição. No passo (5) é definido o novo termo fonte (vetor independente) f^{2h} na malha grossa Ω^{2h} . Este termo fonte é extraído da equação residual dada por 2.12a. O passo (6) representa a resolução (exatamente ou aproximadamente) da equação residual na malha grossa Ω^{2h} com uma estimativa inicial v_0^{2h} a fim de se obter uma aproximação v^{2h} . Em (7) é calculada a correção “ e ” na malha grossa Ω^{2h} . No passo (8) é obtida a correção na

malha fina Ω^h através do processo de prolongação. No passo (9) a aproximação da solução na malha fina Ω^h é corrigida. O passo (10) representa η_2 vezes iterações de suavização (pós-suavização) na malha fina Ω^h com a estimativa inicial v^h , dada pelo passo (9), a fim de se obter uma melhor aproximação v^h . Na prática, η_1 e η_2 são tomados 1, 2 ou 3 (BRIGGS et al., 2000).

Uma chamada de **DG** desenvolve uma única iteração de duas malhas. O esquema a seguir desenvolve diversas chamadas de **DG** até se atingir um certo critério de convergência ou até se atingir um número ite_{\max} pré-fixado.

Chamada do algoritmo DG (adaptado de Wesseling, 1992).

Escolher v_0^h e ite_{\max}

$i = 1$

Enquanto (não atingir o critério de convergência) ou ($i \leq ite_{\max}$), faça

DG(v_0, v, f, h)

$v_0 = v$

Fim enquanto

É de grande valia verificar quais diferenças ocorrem quando tem-se A^k sendo um operador linear. Neste caso é razoável assumir que A^{k+1} também seja linear (WESSELING, 1992). Seja agora o problema $A^h u^h = f^h$ a ser resolvido na malha fina Ω^h . A tabela 3.2 a seguir nos fornece o algoritmo linear que é denominado de esquema de correção (CS).

Tabela 3.2: Esquema CS para duas malhas (adaptado de Briggs et al., 2000)

Esquema de correção para duas malhas

LDG(v_0, v, f, h)

Início

1. Iterar $A^h u^h = f^h$ η_1 vezes com estimativa inicial v_0^h ;
2. Calcular $R^h = f^h - A^h v^h$;
3. Escolher $v_0^{2h} = 0$;
4. Obter $R^{2h} = I_h^{2h} R^h$;
5. Definir $f^{2h} = R^{2h}$
6. Resolver $A^{2h} u^{2h} = f^{2h}$ com estimativa inicial v_0^{2h} ;
7. Calcular $e^{2h} = v^{2h}$;
8. Obter $e^h = I_{2h}^h e^{2h}$;
9. Corrigir $v^h \leftarrow v^h + e^h$;
10. Iterar $A^h u^h = f^h$ η_2 vezes com estimativa inicial v^h .

Fim de **LDG**(v_0, v, f, h)

A diferença básica entre os esquemas CS e FAS está no passo 3 do esquema acima, onde $v_0^{2h} = 0$ no caso do esquema CS. Note que esta estimativa inicial é razoável, pois neste esquema (CS) quer se resolver a equação residual $A^{2h}v^{2h} = R^{2h}$ na malha grossa Ω^{2h} , onde v^{2h} representa a correção e da equação 2.12.

Algumas questões aparecem nestes dois esquemas anteriores (CS e FAS):

1. Como transferir da malha fina Ω^h para a malha grossa Ω^{2h} e vice-versa?
2. Como obter o operador A^{2h} ?

A primeira questão sugere a necessidade de alguns mecanismos, como os operadores de transferência entre malhas, ou seja, os operadores I_h^{2h} (restrição) e I_{2h}^h (prolongação), que serão definidos nas seções seguintes, no decorrer deste capítulo.

A segunda questão sugere mecanismos de geração dos operadores A^{2h} a partir de A^h , I_h^{2h} e I_{2h}^h . Existem basicamente duas formas para se obter A^{2h} :

- i) Aproximação de malha grossa por discretização: como A^h , A^{2h} também é obtida pela discretização da equação diferencial no domínio;
- ii) Aproximação de malha grossa por Galerkin²: $A^{2h} = I_h^{2h} A^h I_{2h}^h$.

Neste trabalho opta-se pela primeira forma, ou seja, pela rediscritização do operador A^h na obtenção de A^{2h} , pois a aproximação de Galerkin demanda muito mais memória no armazenamento dos operadores de restrição e prolongação, além de efetuar operações entre matrizes esparsas de grande porte (I_h^{2h} , A^h e I_{2h}^h). Com isto, este tipo de aproximação é usado somente para fins teóricos (WESSELING, 1992).

Nota-se que o problema na malha grossa é idêntico ao problema original. Portanto, pode-se repetir este processo sucessivamente para outras malhas mais grossas. A vantagem deste processo é que se pode chegar a um nível de malha grossa que seja possível calcular a solução exata (ou com grande precisão) da equação ou da equação residual. Então, basta ter eficientes processos de restrição e prolongação para poder transferir dados de uma malha para outra sem perda na qualidade dessas informações.

A figura 3.2, adaptada de Briggs et al. (2000), mostra um ciclo V com $L = 5$. O ponto representa uma operação de suavização e a linha que os une são os operadores de transferência entre malhas. Pode-se referir também a este ciclo V como ciclo- $V(\eta_1, \eta_2)$ devido aos números de passos de pré e pós-suavização η_1 e η_2 .

² O nome desta técnica nada tem a ver com o nome do método variacional conhecido como Método de Galerkin, utilizado na literatura do Método dos Elementos Finitos.

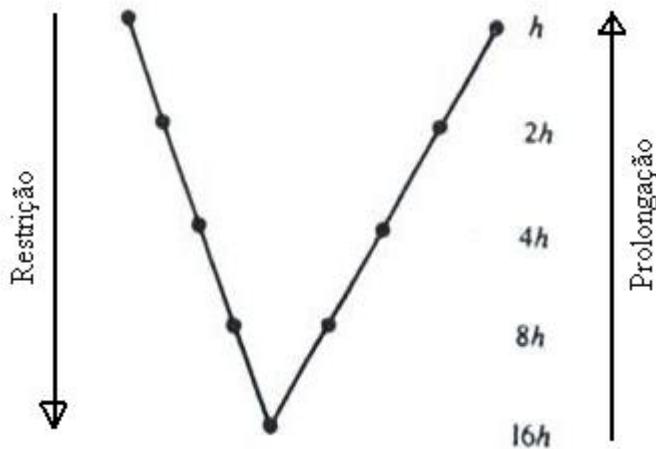


Figura 3.2: Diagrama ciclo V

Nota-se que este ciclo recebe o nome de ciclo V devido ao seu formato. Assume-se então a existência de $L > 1$ malhas, ou níveis, com a razão de engrossamento padrão, portanto, malhas com tamanhos de elementos dados por $h, 2h, 4h, \dots, Kh = 2^{L-1}h$, onde h é o tamanho do elemento da malha mais fina e Kh da malha mais grossa, no caso de $r = 2$.

Por questões didáticas o esquema de correção (e não o esquema de aproximação completa) toma-se como exemplo de algoritmo para várias malhas conforme tabela 3.3. Neste caso, faz-se uso do ciclo V.

A ordem na qual as malhas são visitadas é chamada de ciclo *multigrid*. Existe uma generalização do ciclo V conhecida como ciclo μ (WESSELING, 1992). Desta generalização pode-se obter outros tipos de ciclos, que devido aos seus formatos, são chamados de ciclo W, ciclo F, ciclo-dente-de-serra, etc. Por facilidade na programação, neste trabalho faz-se uso apenas do ciclo V tanto para o esquema de correção (CS) como para o esquema de aproximação completa (FAS).

Considere-se um ciclo V aplicado a um problema d -dimensional com N^d incógnitas. Pode-se verificar em Briggs et al. (2000) que o custo de cada ciclo V é de $O(N^d)$ e com isto, o custo para a convergência do erro de iteração até o nível do erro de discretização é de $O(N^d \log N)$.

Tabela 3.3: Esquema CS para várias malhas com ciclo V (adaptado de Briggs et al., 2000)

Esquema de correção ciclo V	
LMG(v_0, v, f, h)	
Início	
<ul style="list-style-type: none"> • Iterar $A^h u^h = f^h$ η_1 vezes com estimativa inicial v_0^h; • Calcular $R^h = f^h - A^h v^h$; • Obter $f^{2h} = I_h^{2h} R^h$; <ul style="list-style-type: none"> ◦ Iterar $A^{2h} u^{2h} = f^{2h}$ η_1 vezes com estimativa inicial $v_0^{2h} = 0$; ◦ Calcular $R^{2h} = f^{2h} - A^{2h} v^{2h}$; ◦ Obter $f^{4h} = I_{2h}^{4h} R^{2h}$; <ul style="list-style-type: none"> ▪ Iterar $A^{4h} u^{4h} = f^{4h}$ η_1 vezes com estimativa inicial $v_0^{4h} = 0$; ▪ Calcular $R^{4h} = f^{4h} - A^{4h} v^{4h}$; ▪ Obter $f^{8h} = I_{4h}^{8h} R^{4h}$; <ul style="list-style-type: none"> ▪ Corrigir $v^{4h} \leftarrow v^{4h} + I_{8h}^{4h} v^{8h}$; ▪ Iterar $A^{4h} u^{4h} = f^{4h}$ η_2 vezes com estimativa inicial v^{4h}; 	
.	
.	
.	
Resolver $A^{Kh} u^{Kh} = f^{Kh}$;	
.	
.	
<ul style="list-style-type: none"> ◦ Corrigir $v^{4h} \leftarrow v^{4h} + I_{8h}^{4h} v^{8h}$; ◦ Iterar $A^{4h} u^{4h} = f^{4h}$ η_2 vezes com estimativa inicial v^{4h}; ◦ Corrigir $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$; ◦ Iterar $A^{2h} u^{2h} = f^{2h}$ η_2 vezes com estimativa inicial v^{2h}; 	
<ul style="list-style-type: none"> • Corrigir $v^h \leftarrow v^h + I_{2h}^h v^{2h}$; • Iterar $A^h u^h = f^h$ η_2 vezes com estimativa inicial v^h. 	
Fim de LMG(v_0, v, f, h)	

3.2 Processo de restrição

Os operadores que transferem informações da malha fina Ω^h para a malha grossa Ω^{2h} são chamados de operadores de restrição e são denotados genericamente por I_h^{2h} .

Entre os operadores de restrição conhecidos (WESSELING, 1992; BRIGGS et al., 2000; TROTTERBERG et al. 2001), o mais comum é o operador de restrição por injeção. Para a razão de engrossamento $r = 2$ ele é definido por $[I_{inj}]_h^{2h} v^h = v^{2h}$.

Para o caso unidimensional tem-se

$$v_j^{2h} = v_{2j}^h, \quad 1 \leq j \leq \frac{N}{2} - 1, \quad (3.2)$$

onde $N + 1$ é o número de pontos (nós) da malha fina. E para o caso bidimensional tem-se

$$v_{i,j}^{2h} = v_{2i,2j}^h, \quad 1 \leq j \leq \frac{N_x}{2} - 1, \quad 1 \leq j \leq \frac{N_y}{2} - 1, \quad (3.2a)$$

onde $N_x + 1$ e $N_y + 1$ são os números do nós da malha fina nas direções coordenadas x e y , respectivamente.

Outro operador de restrição bastante conhecido (WESSELING, 1992; BRIGGS et al., 2000; TROTTERBERG et al. 2001) é o operador de restrição com ponderação completa, definido por $[I_{pond}]_h^{2h} v^h = v^{2h}$.

Para o caso unidimensional tem-se

$$v_j^{2h} = \frac{1}{4} (v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h), \quad 1 \leq j \leq \frac{N}{2} - 1, \quad (3.3)$$

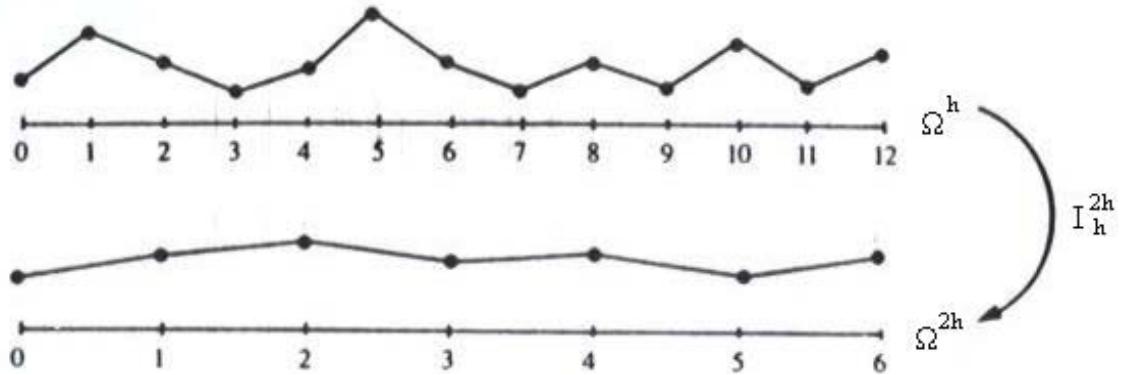
onde $N + 1$ é o número do nós da malha fina. E para o caso bidimensional tem-se

$$\begin{aligned} v_{i,j}^{2h} = & \frac{1}{16} [v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h \\ & + 2(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h) + 4v_{2i,2j}^h], \end{aligned} \quad (3.3a)$$

com $1 \leq i \leq N_x/2 - 1$, $1 \leq j \leq N_y/2 - 1$, onde $N_x + 1$ e $N_y + 1$ são os números dos pontos (nós) da malha fina nas direções coordenadas x e y , respectivamente.

Usa-se a simbologia geral I_h^{2h} para o operador de restrição, independente de qual seja seu tipo, pois isto ficará claro no contexto. Neste trabalho será utilizada uma generalização destes operadores de restrição para qualquer razão de engrossamento para o caso unidimensional. Esta generalização será dada no capítulo 4, na seção que trata do detalhamento dos esquemas CS e FAS utilizado neste trabalho.

A figura 3.3, obtida de Briggs et al. (2000), mostra a ação do operador de restrição por ponderação completa com razão de engrossamento $r = 2$ para o caso unidimensional.



**Figura 3.3: Operador de restrição (ponderação completa) da malha fina para a malha grossa
(BRIGGS et al., 2000)**

São estabelecidas agora as propriedades espectrais do operador com ponderação completa, ou seja, estabelece-se como esse operador de restrição age sobre os modos de Fourier do operador original A^h .

Sejam os modos de Fourier de A^h dados por

$$w_{k,j}^h = \operatorname{sen}\left(\frac{jk\pi}{N}\right), \quad 1 \leq k \leq N-1, \quad 0 \leq j \leq N. \quad (3.4)$$

Pode-se mostrar (BRIGGS et al., 2000) que

$$I_h^{2h} w_k^h = \cos^2\left(\frac{k\pi}{2N}\right) w_k^{2h}, \quad 1 \leq k \leq \frac{N}{2}. \quad (3.5)$$

Da expressão acima conclui-se que o operador I_h^{2h} age nos k -ésimos modos suaves de A^h produzindo os k -ésimos modos de A^{2h} , ou seja, o operador de restrição com ponderação completa não altera o número de ondas (ou freqüência) após sua ação sobre os modos de Fourier suaves. Como o número de pontos após o processo de restrição será sempre menor que o número de pontos da malha anterior e o número de ondas é o mesmo, pode-se concluir que o modo de Fourier tornou-se mais oscilatório nesta malha após a ação deste operador; propriedade desejável no uso do método *multigrid*. Esta propriedade também é verificada para o operador de restrição por injeção.

3.3 Processo de prolongação

Os operadores que transferem informações da malha grossa Ω^{2h} para a malha fina Ω^h são chamados de operadores de prolongação, também conhecidos como interpolação, e são denotados genericamente por I_{2h}^h . Entre os conhecidos (WESSELING, 1992; BRIGGS et al., 2000; TROTTERBERG et al. 2001), o mais comum e bastante eficiente é o operador de interpolação linear, definido por $[I_{lin}]_{2h}^h v^{2h} = v^h$.

Para o caso unidimensional tem-se

$$\begin{cases} v_{2j}^h = v_j^{2h} \\ v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \quad 0 \leq j \leq \frac{N}{2} - 1 \end{cases}, \quad (3.6)$$

onde $N + 1$ é o número do nós da malha fina. E para o caso bidimensional tem-se

$$\begin{cases} v_{2i,2j}^h = v_{i,j}^{2h} \\ v_{2i+1,2j}^h = \frac{1}{2}(v_{i,j}^{2h} + v_{i+1,j}^{2h}) \\ v_{2i,2j+1}^h = \frac{1}{2}(v_{i,j}^{2h} + v_{i,j+1}^{2h}) \\ v_{2i+1,2j+1}^h = \frac{1}{4}(v_{i,j}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}) \end{cases}, \quad (3.6a)$$

com $1 \leq i \leq N_x/2 - 1$, $1 \leq j \leq N_y/2 - 1$, onde $N_x + 1$ e $N_y + 1$ são os números de nós da malha fina nas direções coordenadas x e y , respectivamente.

Outros operadores de prolongação podem ser vistos em Wesseling (1992), como o operador por interpolação quadrática ou o operador de prolongação de De Zeeuw. Usa-se a simbologia geral I_{2h}^h para o operador de prolongação, independente de qual seja seu tipo, pois isto ficará claro no contexto.

A figura 3.4, obtida de Briggs et al. (2000), mostra a ação do operador de prolongação (interpolação linear) com razão de engrossamento $r = 2$ para o caso unidimensional.

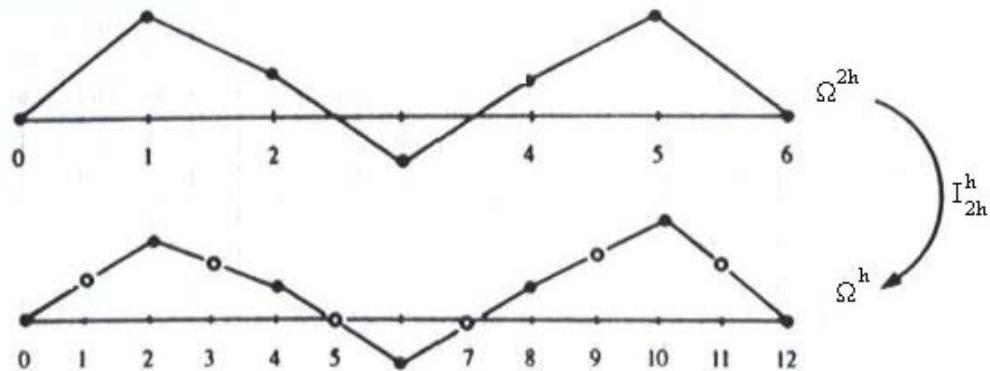


Figura 3.4: Operador de prolongação (interpolação linear) da malha grossa para a malha fina

(BRIGGS et al., 2000)

São estabelecidas agora as propriedades espetrais do operador de interpolação linear, ou seja, estabelece-se como esse operador age sobre os modos de Fourier do operador original A^{2h} .

Sejam os modos de Fourier de A^{2h} dados por

$$w_{k,j}^{2h} = \operatorname{sen}\left(\frac{jk\pi}{N/2}\right), \quad 1 \leq k \leq \frac{N}{2}, \quad 0 \leq j \leq \frac{N}{2}. \quad (3.7)$$

Pode-se mostrar (BRIGGS et al., 2000) que

$$I_{2h}^h w_k^{2h} = \left[1 - O\left(\frac{k^2}{N^2}\right) \right] w_k^h + O\left(\frac{k^2}{N^2}\right) w_{k'}^h. \quad (3.8)$$

Da expressão acima conclui-se que os modos suaves em Ω^{2h} recebem forte influência dos modos suaves em Ω^h e pouca influência dos modos complementares oscilatórios (definido como $k' = N - k$). Inversamente ocorre para os modos oscilatórios em Ω^{2h} . Portanto, o operador de prolongação por interpolação linear altera o número de ondas (ou freqüência) após sua ação sobre os modos de Fourier suaves e oscilatórios. Na hipótese de estarmos prolongando modos suaves (e geralmente é isto que se espera após um processo de suavização na malha grossa) da malha grossa para a malha fina, após o processo de prolongação, este modo irá receber maior influência de modos suaves do que dos modos

oscilatórios, sinalizando a existência de modos suaves na malha fina novamente; propriedade desejável para concluir-se que a interpolação foi de qualidade.

3.4 Problemas anisotrópicos

Nas seções anteriores nada se falou sobre a natureza dos operadores ou o tipo de malha disponível. Pretende-se nesta seção aplicar o método *multigrid* para um tipo de problema mais específico, para tanto esta seção propõe-se a tratar de problemas anisotrópicos.

Considerem-se problemas bidimensionais, tais que:

- **Situação 1:** A equação diferencial tenha coeficientes constantes para as derivadas parciais, porém distintos, nas direções coordenadas. Neste caso define-se como sendo anisotropia física ou anisotropia de coeficientes.
- **Situação 2:** A discretização da malha tenha tamanhos constantes, porém distintos, nas direções coordenadas. Neste caso define-se como sendo anisotropia geométrica ou anisotropia de malha.

Um modelo que exemplifica estas duas situações pode ser dado por:

$$-[c^2 + \varepsilon s^2]u_{xx} + 2(1-\varepsilon)csu_{xy} - [s^2 + \varepsilon c^2]u_{yy} = f, \quad (3.9)$$

onde $c = \cos(\alpha)$, $s = \sin(\alpha)$, $0 \leq \alpha \leq \pi/2$ e $0 < \varepsilon \ll 1$ ou $\varepsilon \gg 1$, ou seja, ε bem distinto da unidade. Aqui ε representa a relação entre os coeficientes ou os tamanhos de malha distintos nas direções coordenadas.

No caso particular onde $\alpha = 0$, a expressão dada pela equação 3.9 torna-se

$$-u_{xx} - \varepsilon u_{yy} = f, \quad (3.10)$$

e diz-se que a anisotropia está alinhada com o eixo coordenado x .

E no caso particular onde $\alpha = \frac{\pi}{2}$, a equação 3.9 torna-se

$$-\varepsilon u_{xx} - u_{yy} = f, \quad (3.11)$$

e diz-se que a anisotropia está alinhada com o eixo coordenado y .

Por questões didáticas, toma-se como problema modelo o caso onde $\alpha = 0$, ou seja, anisotropia alinhada na direção x , dado pela equação 3.10. Neste caso tem-se:

- **Exemplo da situação 1** (anisotropia física): a equação diferencial dada pela equação 3.10 discretizada em uma malha de tamanho h em ambas as direções ($h_x = h_y = h$), com ε bem distinto da unidade. Neste caso a razão de aspecto (RA), conforme definida na seção 1.1, é redefinida em função da equação 3.9 (e obviamente de 3.10 e 3.11) por

$$RA = \sqrt{\varepsilon} . \quad (3.12)$$

- **Exemplo da situação 2** (anisotropia geométrica): a equação de Poisson $-u_{xx} - u_{yy} = f$, discretizada em uma malha com tamanho constante h_x na direção x e $h_y = h_x / \sqrt{\varepsilon}$ na direção y , com ε bem distinto da unidade. Neste caso a razão de aspecto (RA) é redefinida por

$$RA = \frac{h_x}{h_y} = \sqrt{\varepsilon} . \quad (3.12a)$$

Ambas as situações são definidas como “problemas anisotrópicos”, ou seja, problemas onde o acoplamento entre os pontos vizinhos torna-se muito forte em alguma das direções coordenadas (anisotropia física) ou quando a discretização é baseada em malhas com razão entre os tamanhos de malha bem distinto nas direções coordenadas (anisotropia geométrica).

Pode-se ter duas situações, a saber: $0 < \varepsilon \ll 1$ ou $\varepsilon \gg 1$. Quando tem-se $0 < \varepsilon \ll 1$, o forte acoplamento se dá na direção x e quando tem-se $\varepsilon \gg 1$, na direção y . Vale lembrar que para $\varepsilon = 1$ tem-se o caso isotrópico.

Ao aplicar-se o método *multigrid* padrão³, o fator de convergência se deteriora quando se tem $\varepsilon \approx 0$ ou $\varepsilon \gg 1$. Para que se possa entender essa deterioração no caso específico onde se tem $\varepsilon \approx 0$, toma-se o caso extremo, onde $\varepsilon = 0$ na equação 3.10. Com isto tem-se o problema unidimensional

$$-u_{xx} = f , \quad (3.13)$$

³ Quando a razão de engrossamento $r = 2$ em ambas as direções.

ou seja, tem-se a equação de Poisson sem conexão nenhuma na direção y . Portanto, os métodos de relaxação, como Jacobi ponderado ou Gauss-Seidel, suavizarão as componentes de erro apenas na direção x , o que significa que o controle do erro na direção y será totalmente aleatório.

Como em Briggs et al. (2000), aqui se estabelece uma estratégia para tratar de problemas que envolvam este tipo de anisotropia:

- **Estratégia:** Como a boa convergência do método *multigrid* é esperada para o caso unidimensional na direção x , que é a direção de forte acoplamento, deve-se executar o método *multigrid* empregando-se relaxação por pontos, mas com um engrossamento apenas na direção x . A este processo dá-se o nome de “semi-engrossamento”.

Neste caso, a restrição dá-se por um tipo especial de engrossamento onde é tomado duas vezes o espaçamento na direção x (caso seja adotada a razão de engrossamento padrão $r = 2$) e com o espaçamento original na direção y .

A figura 3.5 mostra um processo de semi-engrossamento na direção x .

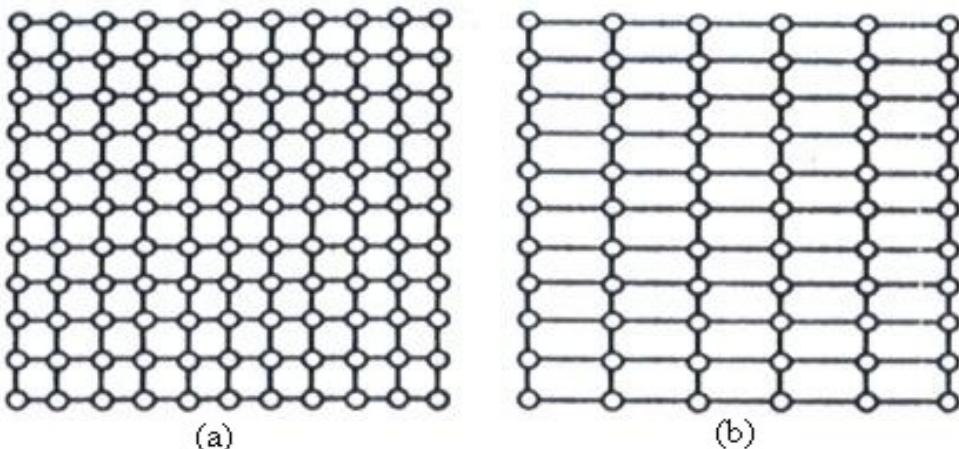


Figura 3.5: (a) malha fina original e (b) malha grossa com semi-engrossamento na direção x

Pelo fato do engrossamento ser unidimensional, o ciclo V mantém sua complexidade da ordem $O(N)$ no caso de semi-engrossamento. Para detalhes sobre outras estratégias para tratar de problemas anisotrópicos, como:

- engrossamento padrão com relaxação por linhas e
- semi-engrossamento com relaxação por linhas,

veja em Briggs et al. (2000), em Wesseling (1992) e outras referências citadas na revisão bibliográfica.

3.5 Resumo do capítulo 3

Neste capítulo apresentaram-se os principais aspectos do método *multigrid*. Mostrou-se um algoritmo bastante simples que desfruta a propriedade de suavização de certos métodos iterativos. Este algoritmo é o esquema de aproximação completa (que trata das não-linearidades) e que tem como caso particular o esquema de correção (que trata das linearidades). Os diversos ciclos no método *multigrid*, com alguns dos principais algoritmos para duas malhas e diversas malhas foram comentados, incluindo o que será utilizado neste trabalho que é o ciclo V. Em seguida detalharam-se os processos de restrição e prolongação indicando seus operadores mais utilizados e suas propriedades espectrais. Finalmente foi descrito um caso especial do método *multigrid*, que é o caso dos problemas anisotrópicos com opções para solução.

4 Problemas unidimensionais lineares: equações de difusão e de advecção-difusão

Neste capítulo descreve-se a primeira classe de problemas tratados neste trabalho, que são os problemas unidimensionais lineares. Neste caso estudam-se as equações de difusão e de advecção-difusão (VERSTEEG e MALALASEKERA, 1995).

Este capítulo está dividido da seguinte forma: primeiramente estudam-se os modelos matemáticos com suas definições e soluções analíticas. Em seguida estudam-se os modelos numéricos com suas discretizações e os esquemas *multigrid* usados: esquema de correção (CS) e esquema de aproximação completa (FAS) e dados de implementação. Finalmente detalham-se os principais resultados numéricos com estes esquemas, como: iterações internas, nível de malha, razão de engrossamento, tolerância e estimativa inicial.

4.1 Modelos matemáticos

Tem-se a seguir um modelo matemático geral para problemas em regime permanente de advecção-difusão com termo fonte (VERSTEEG e MALALASEKERA, 1995)

$$\vec{\nabla} \cdot (\rho \vec{u} \Phi) = \vec{\nabla} \cdot (\Gamma \vec{\nabla} \Phi) + S_\Phi, \quad (4.1)$$

onde $\vec{\nabla} = \partial(\cdot)/\partial x \vec{i} + \partial(\cdot)/\partial y \vec{j} + \partial(\cdot)/\partial z \vec{k}$ é o operador *nabla*, ρ é a massa específica, Γ é o coeficiente de difusão, $\vec{u} = u_1 \vec{i} + u_2 \vec{j} + u_3 \vec{k}$ é o vetor velocidade para os quais u_1 , u_2 e u_3 são as componentes nas direções coordenadas cartesianas x , y e z , respectivamente, S_Φ é o termo fonte de cada propriedade Φ , $\vec{\nabla} \Phi$ é o vetor gradiente do escalar Φ e $\vec{\nabla} \cdot \vec{u}$ é o escalar divergente do vetor \vec{u} .

Considerando-se $\Phi = T$ (temperatura), $S_\Phi = \dot{q}$ (geração de calor) e $\Gamma = \kappa/c_p$ na equação 4.1, obtém-se a equação da conservação da energia

$$\vec{\nabla} \cdot (\rho c_p \vec{u} T) = \vec{\nabla} \cdot (\kappa \vec{\nabla} T) + \dot{q}, \quad (4.2)$$

onde κ é a condutividade térmica do fluido e c_p é o calor específico do fluido.

Para o caso unidimensional, com propriedades constantes e não havendo escoamento, ou seja, velocidade constante $u_1 = 0$, tem-se a seguinte simplificação da equação 4.2

$$\frac{d^2T}{dx^2} + \frac{\dot{q}}{\kappa} = 0,$$

ou ainda,

$$\frac{d^2T}{dx^2} + f(x) = 0, \quad (4.3)$$

onde x é a coordenada cartesiana espacial e $f(x)$ é o termo fonte. Esta equação recebe o nome de equação de difusão e é conhecida também como equação de Poisson (TANNEHILL et al., 1997).

Para o caso unidimensional, velocidade $u_1 \neq 0$ constante, com propriedades constantes (portanto, fluido incompressível) e sem geração de calor, ou seja, $\dot{q} = 0$, tem-se a seguinte simplificação da equação 4.2

$$Pe \frac{dT}{dx} = \frac{d^2T}{dx^2}, \quad (4.3a)$$

onde $Pe = (u_1 c_p \rho C)/\kappa$ é o número adimensional de Peclet (TANNEHILL et al., 1997) e C é o comprimento do domínio. Esta equação recebe o nome de equação de advecção-difusão (TANNEHILL et al., 1997).

Neste capítulo serão consideradas as condições de contorno de Dirichlet num domínio unidimensional unitário, $\{x \in R : 0 \leq x \leq 1\}$, conforme a figura 4.1, portanto os problemas a serem resolvidos serão

$$\begin{cases} \frac{d^2T}{dx^2} + f(x) = 0, & 0 < x < 1 \\ T(0) = 0, \quad T(1) = 1 \end{cases} \quad (4.4)$$

e

$$\begin{cases} Pe \frac{dT}{dx} = \frac{d^2T}{dx^2}, & 0 < x < 1 \\ T(0) = 0, \quad T(1) = 1 \end{cases}. \quad (4.5)$$

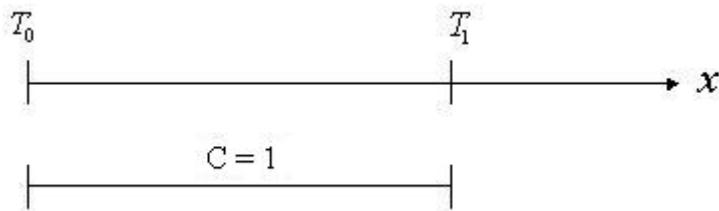


Figura 4.1: Domínio unidimensional de cálculo e condições de contorno

Para a equação 4.4, toma-se $f(x) = 1 + 3x + 26x^2$ e para a equação 4.5, o número de Peclet, $Pe = 20$, será um dado fixo, ou seja, um problema de convecção dominante. O interesse aqui é estudar o efeito da advecção e não da difusão, como já realizado no problema dado pela equação 4.4.

Com a finalidade de buscar-se uma solução analítica para os problemas 4.4 e 4.5, use-se a técnica de solução de equações diferenciais ordinárias lineares de segunda ordem com coeficientes constantes (GREENBERG, 1998). Estas soluções analíticas serão descritas no apêndice B desta Tese.

4.2 Modelos numéricos

Tem-se a seguir um detalhamento dos esquemas numéricos a serem adotados neste capítulo para as equações unidimensionais lineares estudadas (equação de difusão e equação de advecção-difusão), além de suas soluções numéricas. Tem-se também uma breve descrição dos esquemas *multigrid* utilizados: esquema de correção (CS) e esquema de aproximação completa (FAS).

4.2.1 Discretização

O método numérico a ser utilizado nos problemas descritos na seção anterior será o método de diferenças finitas (BURDEN e FAIRES, 1997; TANNEHILL et al., 1997). A discretização do domínio será desenvolvida, fazendo-se uso de malhas estruturadas e uniformes, onde o domínio $\{x \in R : 0 \leq x \leq 1\}$ é partitionado em um número N de subintervalos (elementos). Isto introduz uma malha dada por $x_i = (i - 1)h$, onde $i = 1, \dots, N + 1$ e $h = 1/N$. Neste caso, estabelece-se uma malha com elementos de tamanho h que se denota por Ω^h . Cada ponto de referência será denotado por P , seu vizinho a oeste por W e o seu vizinho a leste por E , conforme figura 4.2.

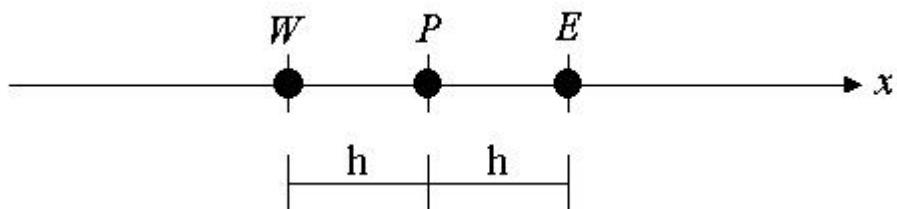


Figura 4.2: Malha 1D uniforme

Primeiramente vamos detalhar o esquema numérico para o problema 4.4, ou seja, a equação de difusão.

Neste esquema numérico é utilizada a diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema CDS pode ser visto em Tannehill et al. (1997). Para cada um dos $N - 1$ pontos interiores da malha, a equação resultante é

$$\left\{ \begin{array}{l} \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + f_i = 0, \quad 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{array} \right. , \quad (4.6)$$

onde v_i é uma aproximação (solução numérica) para a solução exata $T(x_i)$.

Rearranjando os termos da equação 4.6, obtém-se a forma geral

$$a_p v_P = a_w v_W + a_e v_E + b_p, \quad (4.7)$$

onde os coeficientes são dados por

$$a_p = \frac{2}{h^2}, \quad (4.8 \text{ a})$$

$$a_e = a_w = \frac{1}{h^2}, \quad (4.8 \text{ b})$$

$$b_p = f_i. \quad (4.9)$$

Em seguida vamos detalhar os esquemas numéricos para o problema 4.5, ou seja, a equação de advecção-difusão. Neste problema são utilizados dois esquemas distintos:

- 1) No primeiro esquema numérico são utilizadas a diferença atrasada (UDS) para a derivada de primeira ordem (termo advectivo), também conhecido como esquema *Upwind*, e a diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema UDS/CDS pode ser visto em Tannehill et al. (1997). Para cada um dos $N - 1$ pontos interiores da malha, a equação resultante é

$$\left\{ \begin{array}{l} Pe \frac{v_i - v_{i-1}}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2}, \quad 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{array} \right. . \quad (4.10)$$

Rearranjando os termos da equação 4.10, obtém-se a forma geral dada pela equação 4.7, onde os coeficientes são dados por

$$a_p = \frac{2}{h^2} + \frac{Pe}{h}, \quad (4.11 \text{ a})$$

$$a_w = \frac{1}{h^2} + \frac{Pe}{h}, \quad (4.11 \text{ b})$$

$$a_e = \frac{1}{h^2}, \quad (4.11 \text{ c})$$

$$b_p = 0. \quad (4.12)$$

- 2) No segundo esquema numérico são utilizadas a diferença centrada com correção adiada (CDS_CA) para a derivada de primeira ordem (termo advectivo) e diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema CDS_CA/CDS pode ser visto em Lilek et al. (1997). Para cada um dos $N - 1$ pontos interiores da malha, a equação resultante é

$$\left\{ \begin{array}{l} Pe \left[\frac{v_i - v_{i-1}}{h} + \beta \left(\frac{v_{i-1} - 2v_i + v_{i+1}}{2h} \right)^* \right] = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2}, \quad 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{array} \right. , \quad (4.13)$$

onde $0 \leq \beta \leq 1$ e o índice * denota valores que são obtidos da iteração anterior.

Rearranjando os termos da equação 4.13, obtém-se a forma geral dada pela equação 4.7, onde os coeficientes são dados pelas equações 4.11 a, b e c e

$$b_p = -\beta Pe \left(\frac{v_{i-1} - 2v_i + v_{i+1}}{2h} \right)^*. \quad (4.14)$$

Note que para $\beta = 0$ tem-se o esquema UDS/CDS e para $\beta = 1$ tem-se o CDS_CA/CDS.

Se v e f são denotados por $v = (v_1, \dots, v_{N+1})^t$ e $f = (f_1, \dots, f_{N+1})^t$, respectivamente, então os sistemas dados pelas equações 4.6, 4.10 e 4.13, podem ser representados por um sistema de equações algébricas, do tipo da equação 1.1, dada por

$$Av = f, \quad (4.15)$$

onde A é uma matriz tridiagonal $(N+1) \times (N+1)$, definida positiva (Briggs et al., 2000; Burden e Faires, 1997), f é o vetor independente e v é o vetor de incógnitas.

A equação 4.15 é resolvida com o método direto TDMA (Ferziger e Peric, 1999), para obter-se informações sobre a performance do método neste tipo de problema. Resolve-se também este problema, apenas na malha mais fina, com o método *singlegrid* de Gauss-Seidel. No caso do método *multigrid*, os sistemas de equações $Av = b$, onde b representa o termo fonte (resíduo na restrição e correção na prolongação) a cada nível de malha, são resolvidos com o método de Gauss-Seidel a cada nível de malha. Estes métodos de solução das equações empregados foram descritos no apêndice A.

4.2.2 Esquema de correção (CS)

Considerando-se o esquema de correção para duas malhas introduzido pela tabela 3.2 da seção 3.1, pretende-se fazê-lo mais geral, incluindo diversas razões de engrossamento.

Neste caso, como visto na seção 1.2, a razão de engrossamento é definida por

$$r = \frac{h_{\Omega^H}}{h_{\Omega^h}}, \quad (4.16)$$

onde Ω^h é a notação utilizada para representar uma malha fina e Ω^H a malha imediatamente mais grossa. Como pode-se notar, H representa-se também por $H = rh$, por exemplo, $H = 2h$, $H = 3h$, $H = 4h$, etc.

Uma forma alternativa para reescrever a equação 4.16 é

$$r = \frac{p}{q}, \quad p, q \in Z_+^*, \quad p > q. \quad (4.16 \text{ a})$$

Neste caso, a razão de engrossamento r é chamada razão pura, ou inteira, se $q = 1$.

Seja o problema $A^h u^h = f^h$ a ser resolvido e suponha que A^h seja um operador linear. Neste caso é razoável assumir o operador A^H também linear (WESSELING, 1992).

De acordo com o algoritmo dado pela tabela 3.2 (esquema da correção para duas malhas), basta substituir $2h$ por H dos passos 3 ao 8 daquele algoritmo. Nas simulações a serem executadas com este algoritmo já alterado (substituindo-se $2h$ por H nos passos do algoritmo dado pela tabela 3.2), considera-se $\eta_1 = \eta_2 = ITI$ para os passos 1 e 10, ou seja, o número de pré-suavizações igual ao número de pós-suavizações. Neste caso, ITI será

chamado de número de iterações internas. Cada repetição dos passos de 1 a 10 será chamada de iteração externa, também denotada por *ITE*.

Os operadores de transferência da malha fina Ω^h para a malha grossa Ω^H , que são chamados de operadores de restrição, são denotados genericamente por $I_h^H \varphi^h = \varphi^H$, onde φ assume o valor do resíduo R . Vale lembrar que na seção 3.2 foi definido o operador de restrição I_h^H para o caso particular onde $H = 2h$. Neste trabalho um operador de injeção com sua forma generalizada para qualquer razão de engrossamento $r \in (1, \infty)$ foi desenvolvido para problemas unidimensionais. Ele é dado por

$$\varphi_i^H = K_r \varphi_{cr}^h + (1 - K_r) \varphi_{cr+1}^h; \quad 2 \leq i \leq N^H, \quad (4.17)$$

com

$$cr = \text{ceiling}\left(\frac{p}{q}(i-1)\right), \quad K_r = cr - \frac{p}{q}(i-1), \quad N^H = N^h \frac{q}{p} \quad (4.18)$$

e N^h sendo o número de intervalos (elementos) da malha imediatamente mais fina. A função *ceiling* é definida por

$$\text{ceiling} : R \rightarrow Z, \text{ com } x \mapsto \text{ceiling}(x) = \min\{z \in Z / z \geq x\}, \quad (4.19)$$

onde Z é o conjunto dos números inteiros.

A equação 4.17 não é calculada para $i = 1$ e $i = N^H + 1$, visto que neste capítulo adota-se as condições de contorno de Dirichlet. Portanto, $R = 0$ (resíduo nulo) nos pontos de fronteira.

Os operadores de transferência da malha grossa Ω^H para a malha fina Ω^h são chamados de operadores de prolongação, ou interpolação, e são denotados genericamente por $I_H^h \varphi^H = \varphi^h$, onde φ assume a aproximação do erro na equação residual, ou seja, a correção. Vale lembrar que na seção 3.3 foi definido o operador de prolongação I_H^h para o caso particular onde $H = 2h$. Como no caso do operador de restrição, neste capítulo um operador

de interpolação linear com sua forma generalizada para qualquer $r \in (1, \infty)$ foi desenvolvido. Ele é dado por

$$\varphi_i^h = K_p \varphi_{cp}^H + (1 - K_p) \varphi_{cp+1}^H; \quad 2 \leq i \leq N^h, \quad (4.20)$$

com

$$cp = \text{ceiling}\left(\frac{q}{p}(i-1)\right), \quad K_p = cp - \frac{q}{p}(i-1). \quad (4.21)$$

4.2.3 Esquema de aproximação completa (FAS)

Considerando-se o esquema de aproximação completa para duas malhas introduzido pela tabela 3.1 da seção 3.1, pretende-se fazê-lo mais geral, incluindo diversas razões de engrossamento.

Seja o problema $A^h(u^h) = f^h$ a ser resolvido e suponha que A^h seja um operador não-linear.

De acordo com o algoritmo dado pela tabela 3.1 (esquema de aproximação completa para duas malhas), basta substituir $2h$ por H dos passos 3 ao 8 daquele algoritmo.

Novamente, nas simulações a serem executadas, considera-se $\eta_1 = \eta_2 = ITI$, para os passos 1 e 10. Neste caso, como no esquema CS, ITI será chamado de número de iterações internas. Cada repetição dos passos de 1 a 10 será chamada de iteração externa, também denotada por ITE .

Os operadores de restrição e de prolongação são os mesmos dados no esquema de correção, portanto são os operadores dados pelas equações 4.17 e 4.20, respectivamente.

4.2.4 Dados de implementação

Nas simulações a seguir considere-se o número de níveis L , tal que $1 \leq L \leq K$, onde $\Lambda = \{\Omega^L : L = 1, 2, \dots, K\}$ é uma seqüência de malhas sucessivamente mais grossas, com Ω^1 sendo a malha mais fina, Ω^2 a malha imediatamente mais grossa, até Ω^K , que é a malha mais grossa possível.

O critério de parada para as iterações externas *ITE* (número necessário de ciclos para suavizar as componentes de erro) é dado pela razão entre a norma L_1 do resíduo (Ferziger e Peric, 1999) e a norma do resíduo com base na estimativa inicial, onde o resíduo de cada nó é calculado através da equação 2.9. Ou seja, o critério de parada é

$$\frac{\|R(k)\|_1}{\|R(0)\|_1} < \varepsilon, \quad (4.22)$$

onde, o símbolo $R(0)$ representa o resíduo baseado na estimativa inicial e $R(k)$ representa o resíduo na k -ésima iteração. Neste trabalho adota-se como referência $\varepsilon = 10^{-7}$ e $v = (0,0,\dots,1)$ para o critério de parada e estimativa inicial, respectivamente, mas outros valores de ε , como $\varepsilon = 10^{-4}$ e $\varepsilon = 10^{-10}$, e outros valores de v , como $v = (0,1/2,1/2,\dots,1/2,1)$ e $v = (0,1,1,\dots,1)$ também serão adotados. A menos que se diga o contrário, esses valores serão adotados nos demais capítulos.

Todos os algoritmos deste capítulo e dos capítulos seguintes foram implementados na linguagem FORTRAN/95 com o uso do Compaq Visual FORTRAN 6.6, projeto tipo Console e em versão release.

Os testes foram realizados em um microcomputador com Processador Intel Pentium 4 com 2,66 GHz e 1 GB RAM usando aritmética de dupla precisão. Os resultados completos dos testes podem ser vistos no site do Grupo de Dinâmicas dos Fluidos Computacional da UFPR, cito: ftp://ftp.demec.ufpr.br/CFD/Monografias/2006_Marcio_Villela_doutorado.

O foco de estudo deste trabalho é a minimização do tempo de CPU. Entende-se por tempo de CPU, o tempo gasto para realizar a geração de malhas, atribuição da estimativa inicial, cálculo dos coeficientes e resolução do sistema linear 4.15 até atingir o critério de convergência (critério de parada). Este tempo é medido em segundos (s) e usando-se a função TIMEF da biblioteca PORTLIB do FORTRAN/95. Através de testes realizados verificou-se que a incerteza desta função é de aproximadamente $\pm 0,05$ s.

4.3 Resultados numéricos com o esquema de correção (CS)

De acordo com a definição dada pela equação 4.16a, pode-se definir os seguintes engrossamentos que foram usados nesta seção: engrossamento padrão, ou seja, $r = 2$ e alguns engrossamentos fortes, ou seja, $r \in (2, \infty)$, usando-se razões puras da forma $r = p/q$, onde $q = 1$, tais como: $r = 3$, $r = 4$ e $r = 5$. Um pouco mais de 300 simulações foram realizadas com outras variantes, como: número de incógnitas (elementos) dos problemas (N), número de iterações internas (ITI), número de níveis de malhas (L), tolerâncias (ε) e estimativas iniciais (v). Somente alguns destes testes são apresentados nesta seção, pois são os testes mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

A tabela 4.1 mostra em sua primeira linha as razões de engrossamento utilizadas em testes numéricos para as equações de difusão e de advecção-difusão. Na segunda e terceira linhas têm-se os números mínimos e máximos de elementos da malha mais fina (N) nos diversos testes. Na quarta linha têm-se alguns N utilizados nos testes representativos escolhidos. Nota-se que para cada razão de engrossamento foram utilizados N distintos. Entretanto, são valores de N que possibilitam mostrar o efeito qualitativo das propriedades que serão avaliadas nas seções seguintes.

Tabela 4.1: Algumas malhas para cada uma das razões testadas

r	2	3	4	5
N mínimo	2	2	2	2
N máximo	8.388.608	9.565.938	8.388.608	3.906.250
N	1.048.576	1.062.882	2.097.152	781.250

Na figura 4.3 tem-se a solução analítica e as soluções numéricas para a equação de difusão dada pela equação 4.4 com $f(x) = 1 + 3x + 26x^2$, usando-se o esquema numérico CDS dado pela equação 4.6. A solução analítica é comparada com as soluções numéricas para três métodos distintos: *multigrid* usando a razão de engrossamento $r = 2$, o método de Gauss-Seidel e o método direto TDMA.

Neste caso, estipulou-se que as soluções numéricas deveriam atingir erro de máquina, eliminando-se assim o erro de iteração e diminuindo-se o erro numérico (soma dos efeitos dos erros dos erros de iteração, arredondamento e discretização). Para este problema, com $N = 32$ tem-se uma malha que reduz razoavelmente o erro de discretização, por isto é que as soluções numéricas são visualmente coincidentes. Estipulou-se ainda o uso do método de discretização

CDS, reduzindo-se assim o erro de discretização, pois como se sabe, é um método de segunda ordem (VERSTEEG e MALALASEKERA, 1995). Pode-se perceber visualmente com isto, que a solução analítica e as soluções numéricas são coincidentes.

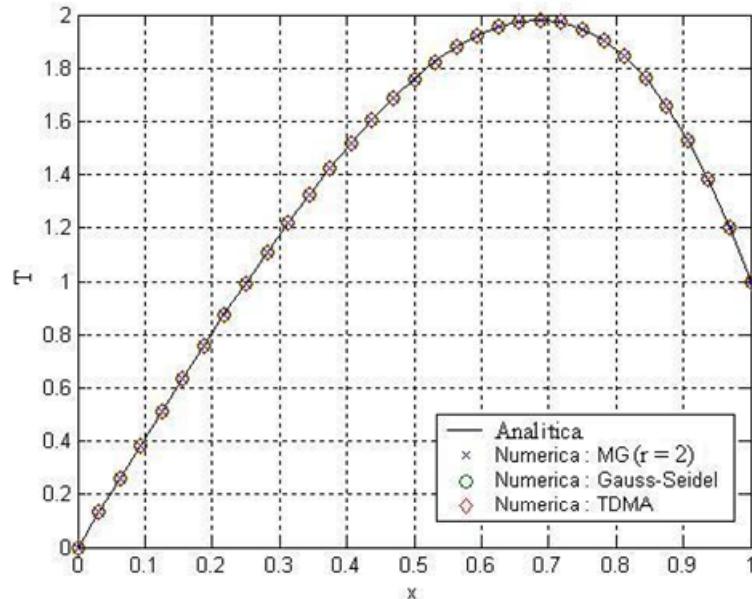


Figura 4.3: Soluções analítica e numéricas para $N = 32$ para a equação de difusão com CDS

Na figura 4.4 tem-se as soluções analítica e numéricas para a equação de advecção-difusão dada pela equação 4.5 com $Pe = 20$, usando-se o esquema numérico UDS/CDS dado pela equação 4.10.

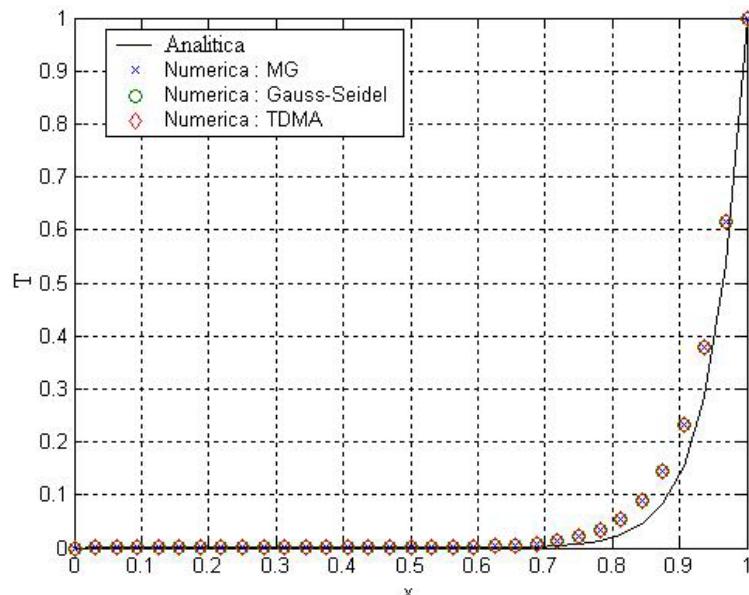


Figura 4.4: Soluções analítica e numéricas para $N = 32$ para a equação de advecção-difusão com UDS/CDS

Na figura 4.4, a solução analítica é comparada com as soluções numéricas para os mesmos métodos que a equação de difusão.

Como anteriormente, estipulou-se que as soluções numéricas deveriam atingir erro de máquina. Estipulou-se ainda o uso do método de discretização UDS/CDS, que é um método de primeira ordem (VERSTEEG e MALALASEKERA, 1995). Pode-se perceber visualmente com isto, que a solução analítica e as soluções numéricas não são coincidentes devido ao erro de discretização.

4.3.1 Iteração interna

Deve-se lembrar que, como foi dito na seção 4.2.2, número de iterações internas (ITI) é definido como sendo $\eta_1 = \eta_2 = ITI$ nos passos 1 e 10 (número de pré-suavizações igual ao número de pós-suavizações) do algoritmo dado pela tabela 3.2 (esquema de correção).

A figura 4.5 mostra a influência do número de iterações internas (ITI) sobre o tempo de CPU para a equação de advecção-difusão. Esta figura mostra também o número ótimo de iterações internas ($ITI_{óptimo}$, que é o ITI que resulta no menor tempo de CPU) para as diversas razões de engrossamento (r) para um número fixo de elementos da malha mais fina (N) dos problemas. Estes números são dados pela última linha da tabela 4.1.

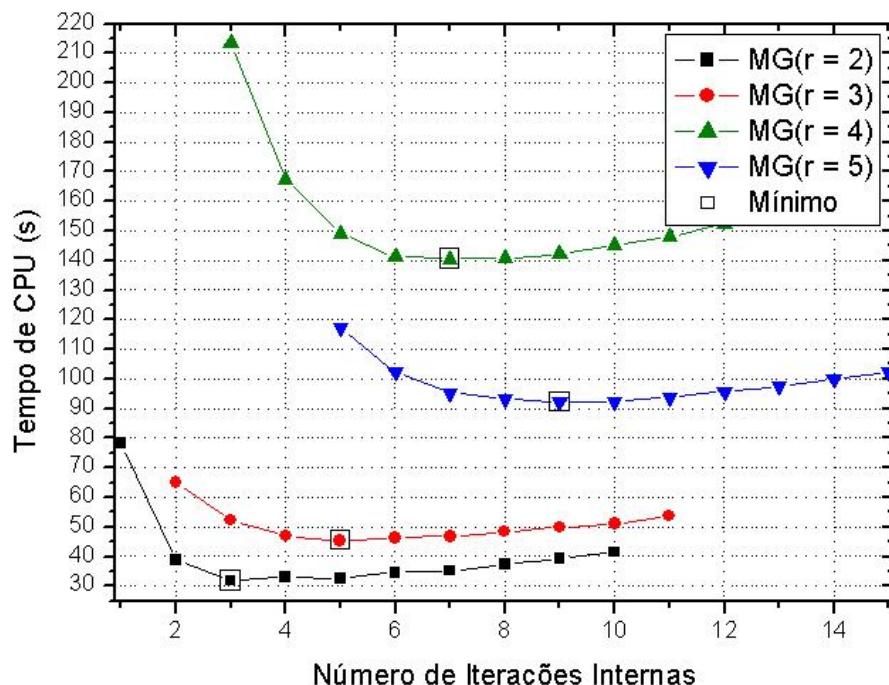


Figura 4.5: Tempo de CPU versus ITI para a equação de advecção-difusão com o esquema CS, diversos r e com o esquema numérico UDS/CDS

Pode-se notar que os valores de N desta tabela, e que são utilizados por estes gráficos, são distintos entre si. Esta figura serve apenas para mostrar o efeito qualitativo da variação do ITI . Nota-se ainda que, o número de iterações internas influencia o tempo de CPU com o uso do esquema CS e que $ITI_{\text{ótimo}} \approx 2r - 1$ para a maioria das razões estudadas, onde r é a razão de engrossamento. Este resultado confirma o resultado de Tannehill et al. (1997) para $r = 2$, no caso da equação de Laplace bidimensional, ou seja, $ITI = 3$ ou $ITI = 4$.

Verificou-se que, para todas as razões estudadas e para ambos os problemas, uma pequena diminuição do número de iterações internas, relativamente ao ponto de mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, o aumento sensível do número de iterações internas, relativamente ao ponto de mínimo, aumenta pouco o tempo de CPU. Portanto recomenda-se utilizar $ITI = 2r$ para as razões puras; por exemplo, para $r = 2$, recomenda-se $ITI = 4$.

Seja o problema de minimização do tempo de CPU, em função do número de iterações internas, para as diversas razões de engrossamento com o uso do esquema CS. Para este problema, verificou-se que o número ótimo de iterações internas tem uma forte influência do número de vezes que o método de Gauss-Seidel necessita para atualizar o resíduo (R) em todo o domínio unidimensional.

4.3.2 Nível de malha

A figura 4.6 mostra a influência do número de níveis (L) sobre o tempo de CPU e o número ótimo de níveis ($L_{\text{ótimo}}$, que é o L que resulta no menor tempo de CPU) para as diversas razões de engrossamento (r) e número de elementos (N) de problemas dados pela tabela 4.1 para a equação de advecção-difusão. Pode-se notar que os valores de N desta tabela, e que são utilizados por estes gráficos, são distintos entre si. Esta figura serve apenas para mostrar o efeito qualitativo da variação do número de níveis.

Nota-se ainda que, o número de níveis pode afetar significativamente o tempo de CPU com o uso do esquema CS e que $L_{\text{ótimo}} \approx L_{\text{máximo}}$ para as diversas razões estudadas para ambos os problemas.

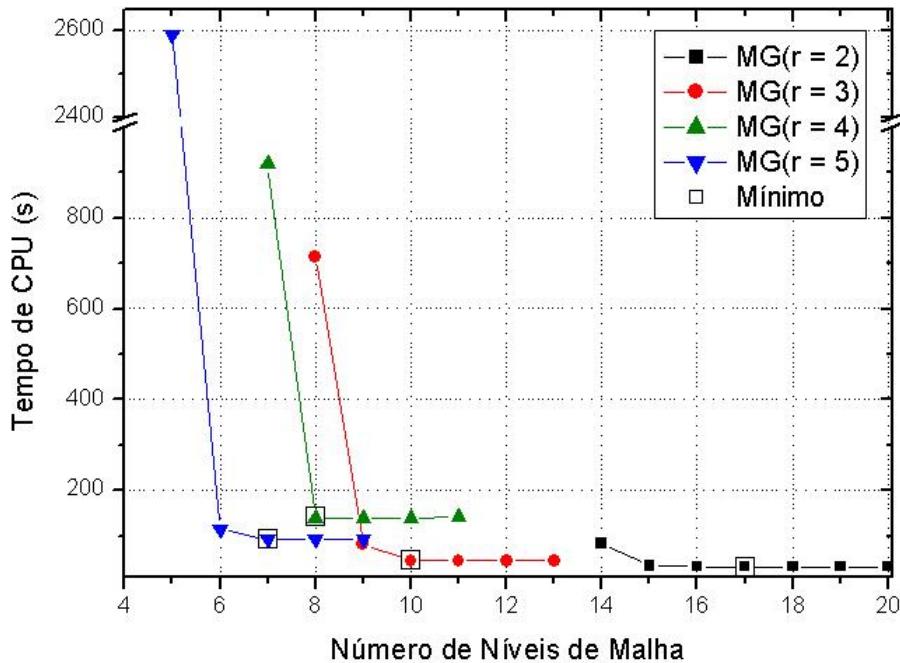


Figura 4.6: Tempo de CPU versus L para a equação de advecção-difusão com o esquema CS, diversos r e com o esquema numérico UDS/CDS

Aqui $L_{máximo}$ representa o número máximo possível de níveis. Por exemplo, uma malha com $N = 1.024$ elementos e usando $r = 2$, possui $L_{máximo} = 10$, isto é, usando-se a razão de engrossamento $r = 2$, para resolver o problema na malha mais fina de 1.024 elementos (que é o nível 1, o nível da malha mais fina, ou seja, Ω^h), pode-se usar no máximo mais 9 níveis adicionais, além de $N = 1.024$, que são: 512 (que é o nível 2, o nível da malha imediatamente mais grossa, ou seja, Ω^{2h}), 256 (nível 3, ou seja, Ω^{4h}), 128 (nível 4, ou seja, Ω^{8h}), 64 (nível 5, ou seja, Ω^{16h}), 32 (nível 6, Ω^{32h}), 16 (nível 7, Ω^{64h}), 8 (nível 8, Ω^{128h}), 4 (nível 9, Ω^{256h}) e 2 (que é o nível 10, o nível da malha mais grossa que são dois elementos com um ponto interno, ou seja, Ω^{512h}). Outro exemplo, uma malha com $N = 1.458$ elementos e usando $r = 3$, possui $L_{máximo} = 7$, isto é, usando-se a razão de engrossamento $r = 3$, para resolver o problema na malha mais fina de 1.458 elementos (que é o nível 1, o nível da malha mais fina, ou seja, Ω^h), pode-se usar no máximo mais 6 níveis adicionais, além de $N = 1.458$, que são: 486 (que é o nível 2, o nível da malha imediatamente mais grossa, ou seja, Ω^{3h}), 162 (nível 3, ou seja, Ω^{9h}), 54 (nível 4, ou seja, Ω^{27h}), 18 (nível 5, ou seja, Ω^{81h}), 6 (nível 6, Ω^{243h}) e 2 (que é o nível 7, o nível da malha mais grossa que são dois elementos com um ponto interno, ou seja, Ω^{729h}).

Para ambos os problemas e para todas as razões testadas, verificou-se que, em geral o tempo de CPU tem uma taxa de decréscimo bastante acentuada até atingir os seus mínimos ($L_{óptimo}$) e, após estes pontos, a taxa de crescimento é bem menor até atingir $L_{máximo}$. Portanto, recomenda-se utilizar $L = L_{máximo}$ para as razões puras (razões da forma $r = p/q$ onde $p \in Z$ e $q = 1$); por exemplo, para a equação de advecção-difusão, para $r = 2$ e $N = 2^{20} = 1.048.576$ elementos, recomenda-se usar $L = 20$, que é exatamente o $L_{máximo}$. Este resultado é similar aos resultados de Tannehill et al. (1997), no caso da equação de Laplace bidimensional (caso linear); e Mesquita e De-Lemos (2004), no caso da equação de Navier-Stokes compressível bidimensional (caso não-linear), ambos para o caso onde $r = 2$.

Seja o problema de minimização do tempo de CPU, em função do número de níveis, para as diversas razões de engrossamento com o uso do esquema CS. Para este problema, verificou-se que o número ótimo de níveis tem uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas, ou seja, malhas com uma quantidade menor de pontos.

4.3.3 Razão de engrossamento

A figura 4.7 mostra o tempo de CPU para as diversas razões de engrossamento para o método *multigrid*, e também uma comparação entre os métodos *singlegrid* Gauss-Seidel e TDMA, em função de diversos N para a equação de advecção-difusão com o uso do esquema CS. Neste trabalho, os métodos diretos são utilizados apenas para efeito de comparação entre os métodos *singlegrid* (SG) e *multigrid* (MG). Pode-se notar que o método TDMA é o mais eficiente de todos os métodos testados para o problema. Vale lembrar que este método é da ordem $O(N)$, enquanto que outros métodos diretos (por exemplo, Eliminação de Gauss), em geral são da ordem $O(N^3)$ (BURDEN e FAIRES, 1997). Este método é seguido pelos métodos *multigrid* e finalmente pelo método de Gauss-Seidel. Métodos iterativos básicos, em geral são da ordem de $O(N^2/2)$ (FERZIGER e PERIC, 1999). Portanto, para os problemas unidimensionais testados, recomenda-se o uso do método direto TDMA. Entretanto o objetivo nesta seção (esquema CS) e na próxima seção (esquema FAS) é verificar o efeito dos parâmetros *multigrid* no tempo de CPU para o *multigrid* geométrico com o uso dos diferentes esquemas (CS e FAS) em problemas unidimensionais lineares a fim de se obter informações para outros tipos de problemas que serão tratados nos capítulos seguintes.

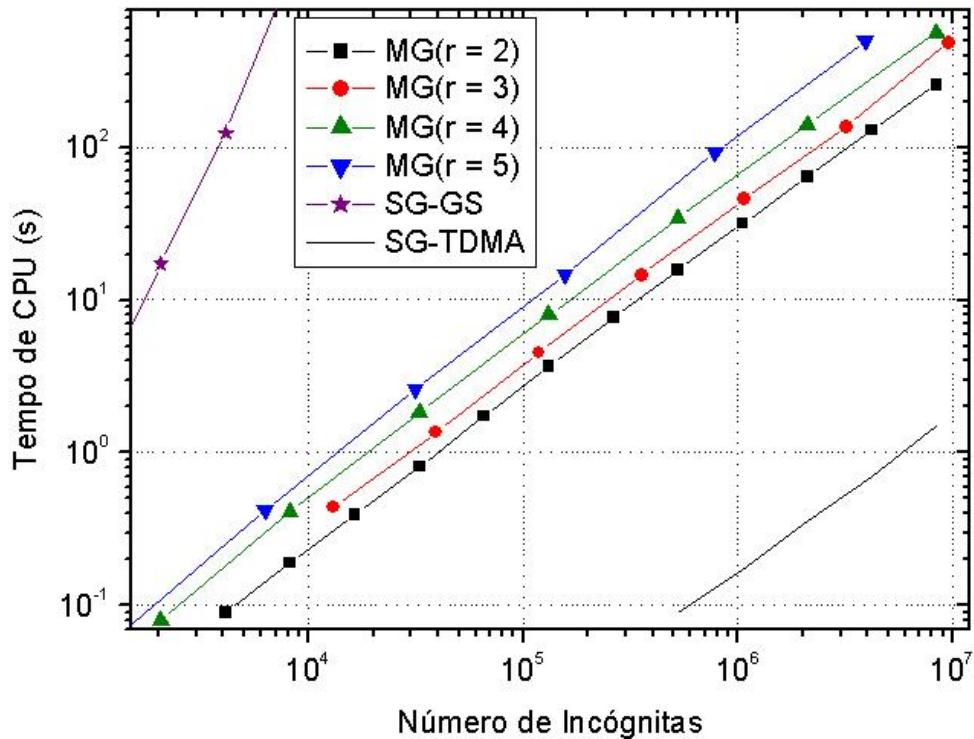


Figura 4.7: Tempo de CPU versus N para a equação de advecção-difusão com o esquema CS, diversos r , com o esquema numérico UDS/CDS e diversos métodos

Para os mesmos ε , v e N , com as razões de engrossamento testadas e o método *multigrid* esquema CS, pode-se perceber que para os problemas testados (equação de difusão e equação de advecção-difusão), tem-se

$$t_{CPU}(r=2) < t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=5). \quad (4.23)$$

Portanto, no caso do método *multigrid* com o uso do esquema CS, recomenda-se o uso da razão $r = 2$ para as equações em questão.

Este estudo confirma as análises de Brandt (1977). Em seu trabalho, Brandt fez análises teóricas e experimentais (numéricas) entre as razões de engrossamento $r = 2$, $r = 3$ e $r = 3/2$ para diversos problemas, mas não tratou estes problemas. Em seus estudos constatou que a razão $r = 2$ é a recomendável, por estar mais próxima da razão de engrossamento ótima e ser mais conveniente e econômica para o processo de interpolação, além de ser de mais fácil implementação. Briggs et al. (2000) afirmam que $r \neq 2$, em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas isto é verdade.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do esquema CS. Para este problema, verificou-se que a razão ótima de engrossamento (r que resulta no menor tempo de CPU) tem uma forte influência de dois principais aspectos: a razão de queda do número de incógnitas (elementos) de uma malha para a outra malha imediatamente mais grossa e a qualidade das informações passadas pela restrição e pela prolongação.

4.3.4 Tolerância e estimativa inicial

Nesta seção estudam-se os casos com variações, tanto da tolerância ε como da estimativa inicial v .

Os seguintes engrossamentos foram usados: engrossamento padrão, ou seja, $r = 2$ e um engrossamento forte usando-se uma razão pura, que no caso foi $r = 3$. Estes engrossamentos foram utilizados por terem apresentado boas características no que diz respeito à minimização do tempo de CPU, apresentadas nas seções anteriores.

Para a variação da tolerância, estudam-se os casos onde $\varepsilon = 10^{-4}$ e $\varepsilon = 10^{-10}$, além de $\varepsilon = 10^{-7}$ usado nos resultados anteriores. Os resultados dos estudos do número ótimo de iterações internas sofreram pouca variação sobre o tempo de CPU: cerca de 2,40% ao se adotar $ITI = 2r$ para a equação de difusão e cerca de 3,40% para a equação de advecção-difusão.

Para a variação da estimativa inicial, estudam-se os casos onde $v = (0,1/2,\dots,1/2,1)$ e $v = (0,1,\dots,1)$, além de $v = (0,0,\dots,1)$ usado nos resultados anteriores. Os resultados dos estudos do número ótimo de iterações internas sofreram uma variação sobre o tempo de CPU um pouco mais substancial: cerca de 28,70% ao se adotar $ITI = 2r$ para a equação de difusão e cerca de 22,00% para a equação de advecção-difusão.

Em relação ao número de níveis, ao se adotar $L = L_{máximo}$, a maior variação encontrada, foi para a equação de difusão, e esta variação foi de 0,90%, independentemente de se estar variando a tolerância ou a estimativa inicial.

As tabelas 4.2 e 4.3, para as equações de difusão e de advecção-difusão, respectivamente, mostram em suas colunas, o estudo referente ao $ITI_{ótimo}$, $L_{ótimo}$ e t_{CPU} para as razões de engrossamento $r = 2$ (com $N = 1.048.576$ e $L_{máximo} = 20$) e $r = 3$ (com $N = 1.062.822$ e $L_{máximo} = 13$), para problemas variando a tolerância ε e considerando

primeiramente a estimativa inicial fixa. Neste caso, adotou-se a estimativa inicial padrão $v = (0,0,\dots,1)$ para os dois tipos de problemas.

Nestas tabelas e nas demais, o símbolo – denota que as simulações não foram realizadas.

Tabela 4.2: Variação de parâmetros de acordo com a variação da tolerância no esquema CDS para a equação de difusão

ϵ	$ITI_{\text{ótimo}}$		t_{CPU} de $ITI_{\text{ótimo}}$		$L_{\text{ótimo}}$		t_{CPU} de $L_{\text{ótimo}}$	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
10^{-4}	3	6	18,39	25,45	18	-	18,36	-
10^{-7}	3	5	32,89	46,72	19	-	32,62	-
10^{-10}	3	5	47,03	68,28	18	-	46,91	-

Tabela 4.3: Variação de parâmetros de acordo com a variação da tolerância no esquema UDS/CDS para a equação de advecção-difusão

ϵ	$ITI_{\text{ótimo}}$		t_{CPU} de $ITI_{\text{ótimo}}$		$L_{\text{ótimo}}$		t_{CPU} de $L_{\text{ótimo}}$	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
10^{-4}	3	6	17,88	24,78	13	9	17,84	24,72
10^{-7}	3	5	31,83	45,38	17	10	31,81	45,12
10^{-10}	-	-	-	-	-	-	-	-

As tabelas 4.4 e 4.5, para as equações de difusão e de advecção-difusão, respectivamente, mostram em suas colunas, o estudo referente ao $ITI_{\text{ótimo}}$, $L_{\text{ótimo}}$ e t_{CPU} para as razões de engrossamento $r = 2$ (com $N = 1.048.576$ e $L_{\text{máximo}} = 20$) e $r = 3$ (com $N = 1.062.822$ e $L_{\text{máximo}} = 13$), para problemas variando as estimativas iniciais e considerando agora a tolerância fixa. Neste caso, adotou-se a tolerância padrão $\epsilon = 10^{-7}$.

Isto mostra que o $ITI_{\text{ótimo}}$ é fracamente dependente da tolerância ϵ (grande variação da tolerância implica em pequena variação do $ITI_{\text{ótimo}}$) e fortemente dependente da estimativa inicial v (grande variação da estimativa inicial implica em grande variação do $ITI_{\text{ótimo}}$). E o $L_{\text{ótimo}}$ é fracamente dependente da tolerância e estimativa inicial. Conclui-se que se pode utilizar $ITI_{\text{ótimo}} = 2r$ e $L_{\text{ótimo}} = L_{\text{máximo}}$ para as equações de difusão e de advecção-difusão com o uso do esquema CS.

Tabela 4.4: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema CDS para a equação de difusão

v	$ITI_{\text{ótimo}}$		t_{CPU} de $ITI_{\text{ótimo}}$		$L_{\text{ótimo}}$		t_{CPU} de $L_{\text{ótimo}}$	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
$v = (0,0,\dots,1)$	3	5	32,89	46,72	19	-	32,62	-
$v = (0,1/2,\dots,1/2,1)$	4	5	31,05	44,52	18	-	30,94	-
$v = (0,1,\dots,1)$	2	3	20,55	24,27	19	-	20,47	-

Tabela 4.5: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema UDS/CDS para a equação de advecção-difusão

v	$ITI_{\text{ótimo}}$		t_{CPU} de $ITI_{\text{ótimo}}$		$L_{\text{ótimo}}$		t_{CPU} de $L_{\text{ótimo}}$	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
$v = (0,0,\dots,1)$	3	5	31,83	45,38	17	10	31,81	45,12
$v = (0,1/2,\dots,1/2,1)$	-	-	-	-	-	-	-	-
$v = (0,1,\dots,1)$	2	3	19,98	23,58	20	13	19,95	23,55

4.4 Resultados numéricos com o esquema de aproximação completa (FAS)

Os seguintes engrossamentos foram usados nesta seção: engrossamento padrão, ou seja, $r = 2$ e alguns engrossamentos fortes usando-se razão pura. Cada esquema numérico usa as seguintes razões de engrossamento: UDS/CDS ($r = 2$, $r = 3$, $r = 4$ e $r = 5$) e CDS_CA/CDS ($r = 2$ e $r = 3$). Estes engrossamentos foram utilizados por terem apresentado boas características no que diz respeito à minimização do tempo de CPU apresentadas na seção anterior onde se utilizou o algoritmo CS. Cerca de 140 simulações foram realizadas com outras variantes, como: número de incógnitas (elementos) dos problemas (N), número de iterações internas (ITI), número de níveis de malhas (L), tolerâncias (ε) e estimativas iniciais (v). São apresentados nesta seção os resultados mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

Nesta seção não se apresentam testes com relação à equação de difusão. Pôde-se perceber que os resultados com o esquema FAS não seriam muito distintos para as duas equações (difusão e advecção-difusão), aos resultados alcançados com o esquema CS, já que a equação de advecção-difusão apresenta esquemas numéricos de discretização um pouco mais complexo ao esquema numérico de discretização da equação de difusão, ambas lineares.

Os números de incógnitas (elementos) usados nesta seção são dados pela tabela 4.1.

4.4.1 Iteração interna

Através da figura 4.5 concluiu-se que o número de iterações internas influencia o tempo de CPU para o esquema CS. A figura 4.8 mostra, para o esquema numérico UDS/CDS, a influência do número de iterações internas (*ITI*) sobre o tempo de CPU para alguns valores de N e razões de engrossamento com o uso do esquema FAS.

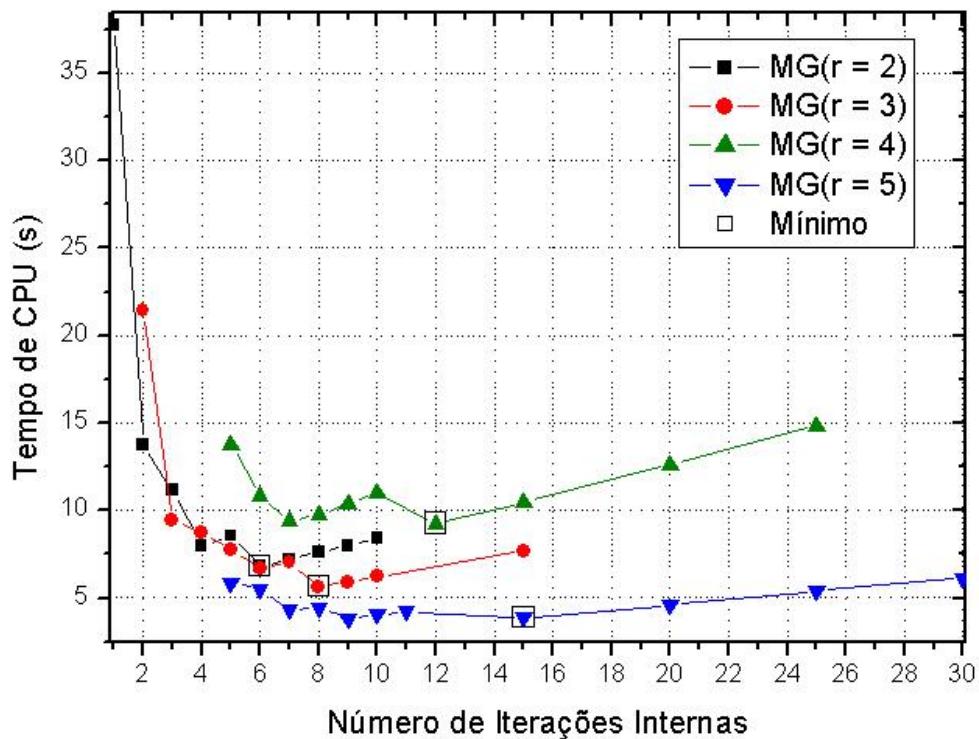


Figura 4.8: Tempo de CPU versus *ITI* para a equação de advecção-difusão com o esquema FAS, diversos r e com o esquema numérico UDS/CDS

Esta figura mostra também o $ITI_{óptimo}$ para as razões de engrossamento consideradas. Pode-se notar que os valores de N deste gráfico são distintos entre si. Esta figura serve apenas para mostrar o efeito qualitativo da variação do número de iterações internas. Nota-se ainda que o número de iterações internas tem influência o tempo de CPU para o esquema FAS e que $ITI_{óptimo} \approx 3r - 1$ para ambas as razões.

Para as razões estudadas, verificou-se que uma diminuição sensível do número de iterações internas, relativamente ao ponto mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, o aumento pequeno do número de iterações internas, relativamente ao ponto mínimo, aumenta pouco o tempo de CPU. Portanto

recomenda-se utilizar $ITI = 3r$ para as razões puras; por exemplo, para $r = 2$, recomenda-se $ITI = 6$ para ambos os problemas (equação de difusão e de advecção-difusão) e para ambos os esquemas numéricos.

Na seção anterior, verificou-se que $ITI_{\text{ótimo}} \approx 2r$ para o esquema CS. Aqui para o mesmo problema, verificou-se que $ITI_{\text{ótimo}} \approx 3r$ para o esquema FAS. Portanto, o esquema utilizado (CS ou FAS) influencia o $ITI_{\text{ótimo}}$ no método *multigrid*, pois no esquema FAS há a necessidade de atualização do resíduo e solução, enquanto que no esquema CS, apenas do resíduo.

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema FAS. Para este problema, verificou-se que o número ótimo de iterações internas tem uma forte influência do número de vezes que o método de Gauss-Seidel necessita para atualizar o resíduo e a solução em todo o domínio unidimensional.

4.4.2 Nível de malha

Através da figura 4.6 concluiu-se que o número de níveis (L) influencia o tempo de CPU para o esquema CS. A figura 4.9 mostra a influência de L sobre o tempo de CPU para alguns valores de N com o uso do esquema FAS.

Esta figura mostra também $L_{\text{ótimo}}$ para as razões de engrossamento consideradas para a equação de advecção-difusão usando o esquema numérico UDS/CDS. Nota-se que o número de níveis pode afetar显著mente o tempo de CPU e que $L_{\text{ótimo}} \approx L_{\text{máximo}}$ para as razões estudadas.

Nota-se que, em geral o tempo de CPU tem uma taxa de decréscimo bastante acentuada até atingir os seus mínimos e, após estes pontos, a taxa de crescimento é bem menor até atingirem $L_{\text{máximo}}$. Portanto recomenda-se utilizar $L = L_{\text{máximo}}$ para as razões puras.

Para os problemas lineares unidimensionais estudados na seção anterior, verificou-se $L_{\text{ótimo}} \approx L_{\text{máximo}}$ para o esquema CS. Aqui, para o problema linear unidimensional estudado, verificou-se $L_{\text{ótimo}} \approx L_{\text{máximo}}$ para o esquema FAS. Portanto, o esquema utilizado (CS ou FAS) não influencia o número ótimo de níveis para o método *multigrid*.

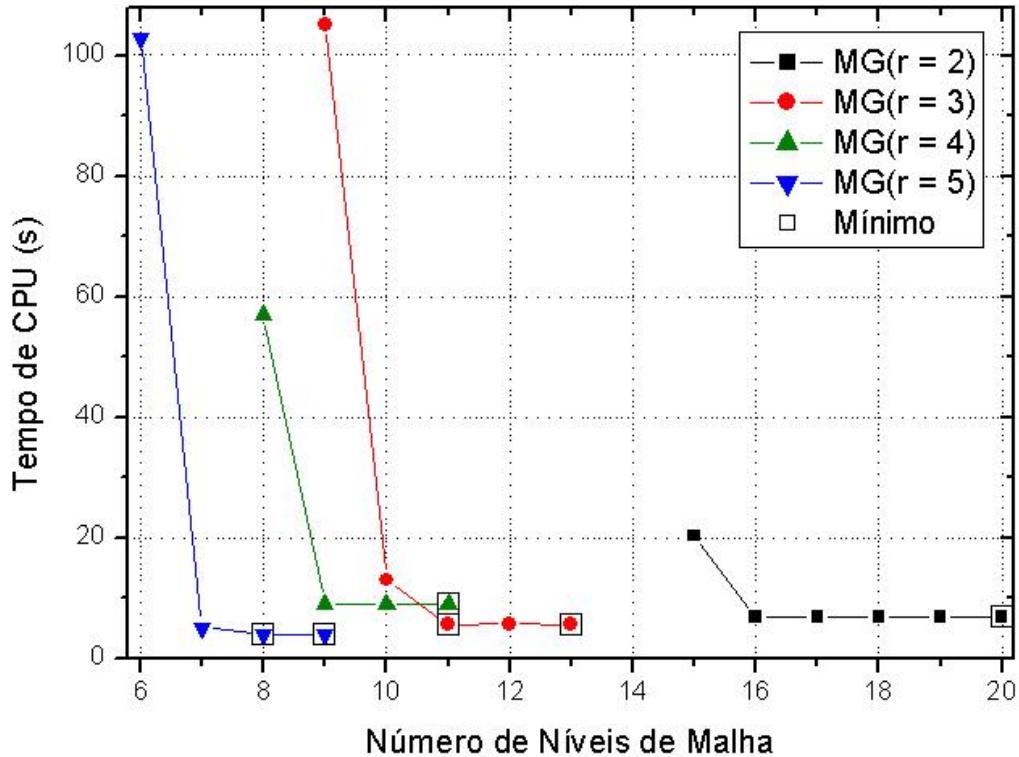


Figura 4.9: Tempo de CPU versus L para a equação de advecção-difusão com o esquema FAS, diversos r e com o esquema numérico UDS/CDS

4.4.3 Razão de engrossamento

A figura 4.10 mostra, para o esquema numérico UDS/CDS, o tempo de CPU para as razões de engrossamento consideradas para o método *multigrid* e também uma comparação entre métodos *singlegrid* Gauss-Seidel e TDMA em função de diversos N para a equação de advecção-difusão. A figura 4.11 mostra um detalhe da figura 4.10.

Pode-se notar novamente que o método TDMA é o mais eficiente dos métodos testados. Ele é seguido pelos métodos *multigrid* e finalmente pelo método de Gauss-Seidel.

Para os mesmos ε , v e N , com as razões de engrossamento testadas e o método *multigrid* com o esquema FAS, pode-se perceber que para a equação de advecção-difusão, para a maioria dos pontos do domínio em estudo, tem-se

$$t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=2) < t_{CPU}(r=5). \quad (4.24)$$

Portanto, no caso do método *multigrid* com o uso do esquema FAS, recomenda-se o uso da razão $r = 3$ para a equação em estudo. Esta constatação é inédita e inesperada, além de

mostrar que a razão $r = 2$ perde a hegemonia perante outras razões, pois com o uso do esquema FAS, há a atualização do resíduo e da solução. Por este motivo, pode-se fazer uso de um engrossamento mais agressivo ($r = 3$, por exemplo) de tal forma que os processos de restrição e de prolongação não percam qualidade.

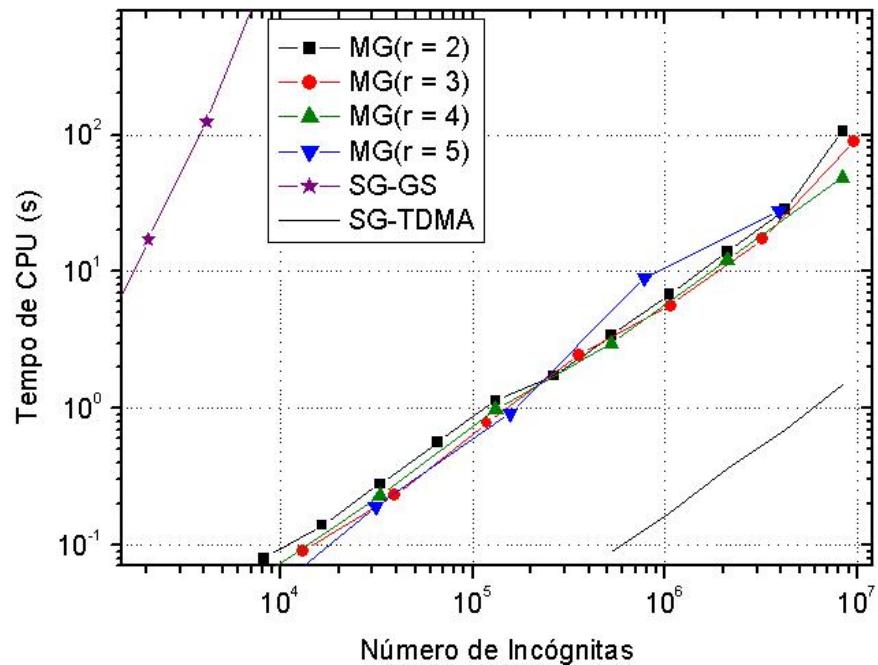


Figura 4.10: Tempo de CPU versus N para a equação de advecção-difusão com o esquema FAS, diversos r , com o esquema numérico UDS/CDS e diversos métodos

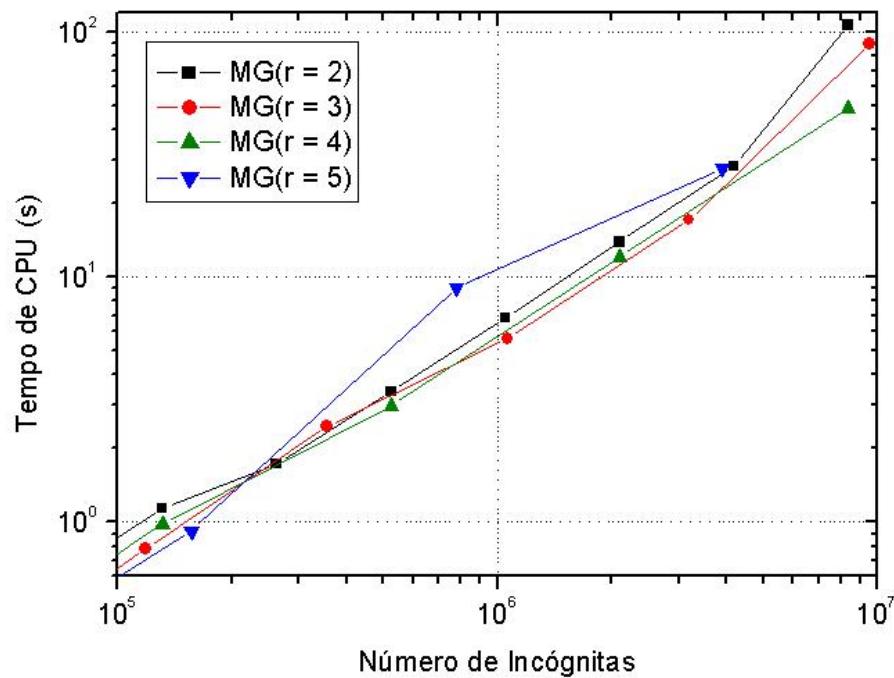


Figura 4.11: Detalhe da figura 4.10

Entretanto, Brandt (1977) em seu trabalho fez análises entre as razões de engrossamento $r = 2$, $r = 3$ e $r = 3/2$ para diversos problemas (mas não especificamente para as equações 4.4 e 4.5), entre eles, problemas elípticos e de escoamento transônico, e constatou que a razão $r = 2$ é a recomendável. Briggs et al. (2000) afirmam que $r \neq 2$, em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas.

Para os problemas lineares unidimensionais estudados na seção anterior, verificou-se que $r = 2$ é a razão ótima para o esquema CS. Aqui, para o problema linear unidimensional estudado, verificou-se que $r = 3$ é a razão ótima para o esquema FAS. Portanto, pode-se concluir que o esquema utilizado (CS ou FAS) influencia a razão ótima para o método *multigrid* para os casos estudados.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do algoritmo FAS. Para este problema, constatou-se que a razão ótima tem uma forte influência dos seguintes aspectos: a razão de queda do número de pontos de uma malha para a outra malha imediatamente mais grossa, a qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, o uso do esquema FAS, que transfere informações a respeito do resíduo e da solução para as demais malhas mais grossas.

4.4.4 Tolerância e estimativa inicial

Nesta seção estudam-se os casos com variações, tanto da tolerância ε como da estimativa inicial v , como na seção anterior.

Para a variação da tolerância, os resultados dos estudos do número ótimo de iterações internas sofreram uma variação substancial sobre o tempo de CPU: cerca de 6,87% a 24,53% ao se adotar $ITI = 3r$ para a equação de advecção-difusão.

Para a variação da estimativa inicial, os resultados dos estudos do número ótimo de iterações internas sofreram uma variação significativa sobre o tempo de CPU: cerca de 11,93% a 32,65% ao se adotar $ITI = 3r$ para a equação de advecção-difusão.

Em relação ao número de níveis, ao se adotar $L = L_{máximo}$, a maior variação encontrada para a equação de advecção-difusão foi de 24,78%, independentemente de se estar variando a tolerância ou a estimativa inicial.

Isto mostra que o $ITI_{óptimo}$ continua sendo fracamente dependente da tolerância ε e fortemente dependente da estimativa inicial v e $L_{óptimo}$ é fracamente dependente da tolerância

e estimativa inicial. Conclui-se que se pode utilizar $ITI_{\text{ótimo}} = 3r$ e $L_{\text{ótimo}} = L_{\text{máximo}}$ para as equações de advecção-difusão com o uso do esquema FAS.

4.5 Esquema de correção (CS) *versus* esquema de aproximação completa (FAS)

Usando-se as figuras 4.7 e 4.10 para uma comparação entre os esquemas *multigrid* CS e FAS para as razões de engrossamento $r = 2$ e $r = 3$, pode-se notar que o esquema FAS é mais rápido do que o esquema CS em qualquer N para as duas razões analisadas. Por exemplo, para a malha de 4.194.304 elementos o tempo de CPU do MG-FAS($r = 2$) e MG-CS($r = 2$) é de 28,48 s e 129,28 s, respectivamente, ou seja, o MG-FAS($r = 2$) é cerca e 4,5 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 3.188.646 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 3$) é de 17,19 s e 136,70 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca e 7,9 vezes mais rápido do que o MG-CS($r = 3$).

Na figura 4.12 se faz uma comparação entre os esquemas *multigrid* CS (com a melhor razão de engrossamento, $r = 2$) e FAS (com a melhor razão de engrossamento, $r = 3$) em função de diversos valores de N .

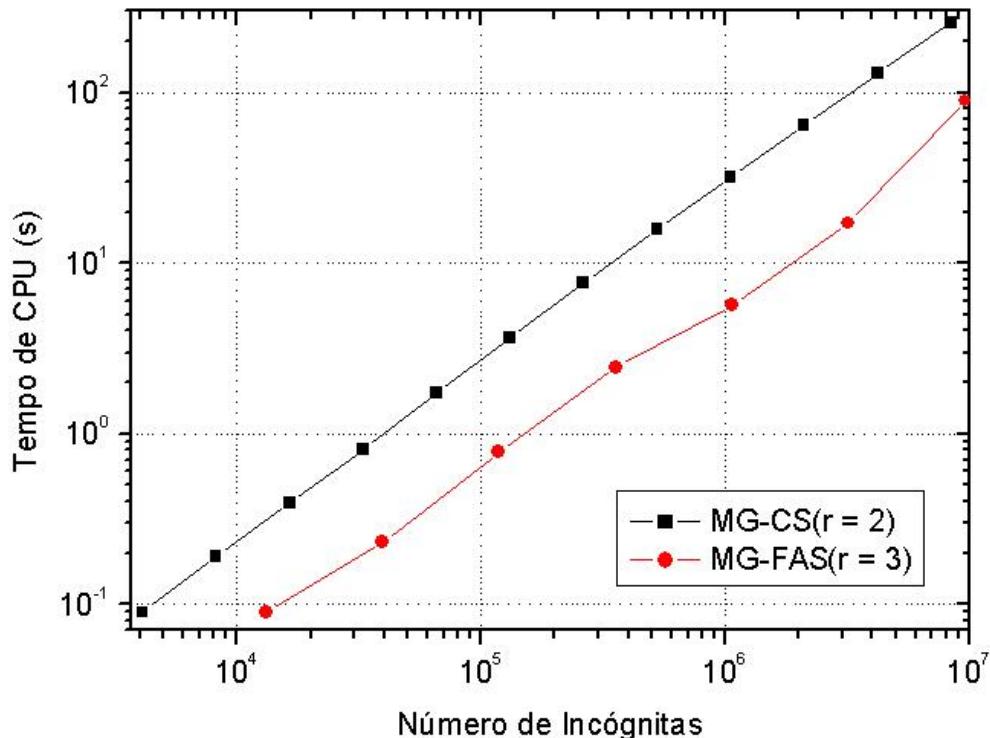


Figura 4.12: Tempo de CPU *versus* N para a equação de advecção-difusão com os esquemas CS e FAS, $r = 2$ e $r = 3$, com o esquema UDS/CDS

Pode-se notar que o esquema FAS é mais rápido do que o CS em qualquer N .

Usando-se ajuste geométrico de curvas, o tempo de CPU do MG-FAS($r = 3$) pode ser aproximado por $t_{CPU} = 3,85 \times 10^{-5} N^{1,04}$ e MG-CS($r = 2$) pode ser aproximado por $t_{CPU} = 1,53 \cdot 10^{-5} N^{1,05}$.

Pode-se notar que o esquema MG-FAS($r = 3$) é mais rápido do que o esquema MG-CS($r = 2$) em qualquer N . Por exemplo, para a malha de 10^5 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 0,60 s e 2,65 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,4 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 10^6 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 6,56 s e 29,59 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,5 vezes mais rápido do que o MG-CS($r = 2$). O motivo para este fato ocorra (FAS mais rápido do que CS) será explicado ao final desta seção.

Portanto, recomenda-se o uso do esquema FAS com a razão $r = 3$ para a equação de advecção-difusão. Esta constatação é inédita e inesperada, pois o esquema CS é indicado para resolver equações lineares e o esquema FAS, para equações não-lineares. Aqui vemos que o esquema CS perde a hegemonia perante o esquema FAS, mesmo que para um problema linear.

A conclusão do presente capítulo difere daquela de Brandt (1977). Ele fez análises teóricas e experimentais (numéricas) entre as razões de engrossamento $r = 2, 3$ e $3/2$ para diversos problemas, mas não se referiu à equação 4.5 exatamente. Brandt mostra preferência pelo esquema CS em relação ao esquema FAS no caso de problemas lineares. Segundo Brandt, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS devido aos cálculos exigidos no esquema FAS.

Seja o problema de minimização do tempo de CPU em função do uso dos diferentes esquemas CS e FAS. Para este problema, verificou-se que o esquema ótimo tem uma forte influência da qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, pela transferência de informações a respeito do resíduo e da solução para as malhas mais grossas. Entretanto, no presente trabalho verificou-se que o número de ciclos ou iterações externas do esquema FAS é significativamente menor do que o esquema CS. Por exemplo, para a equação de advecção-difusão e para a malha de 4.194.304 elementos, o número de ciclos, ou iterações externas (ITE), do MG-FAS($r = 2$) e MG-CS($r = 2$) é de 3 e 25, respectivamente.

4.6 Conclusão do capítulo 4

Neste capítulo foram resolvidos numericamente dois problemas: equações de Poisson e advecção-difusão unidimensionais. Estas equações são lineares e elas representam problemas de transferência de calor. Os esquemas numéricos de aproximação usados foram: CDS, UDS/CDS e CDS_CA/CDS. Foram descritos e usados dois esquemas do método *multigrid*: esquemas CS (*Correction Scheme*) e FAS (*Full Approximation Scheme*).

Com base nos resultados deste capítulo, verificou-se que:

1. Para os mesmos dados (ε , v e N), o esquema FAS é mais rápido do que o esquema CS;
2. O número de iterações internas (ITI) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) influencia o $ITI_{\text{ótimo}}$. Com o uso do esquema CS, $ITI_{\text{ótimo}} \approx 2r$, e com o uso do esquema FAS, $ITI_{\text{ótimo}} \approx 3r$;
3. O número de níveis de malhas (L) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) não tem muita influência no $L_{\text{ótimo}}$. Para ambos os esquemas $L_{\text{ótimo}} \approx L_{\text{máximo}}$;
4. Para os mesmos dados (ε , v e N), com $L = L_{\text{máximo}}$ para cada N , entre as razões de engrossamento de malhas testadas ($r = 2, 3, 4$ e 5), o esquema CS é mais rápido com $r = 2$; já o esquema FAS é mais rápido com $r = 3$;
5. O esquema FAS com razão de engrossamento $r = 3$ é aproximadamente 4 vezes mais rápido do que o esquema CS com razão de engrossamento $r = 2$.

Recomendações práticas:

Com base nos resultados deste capítulo são feitas as seguintes recomendações aos usuários do método *multigrid* para problemas iguais ou similares aos estudados.

Problemas unidimensionais lineares:

- Esquema: FAS;
- Razão de engrossamento: $r = 3$;
- Número de iterações internas: $ITI = 9$;
- Número de níveis de malhas: $L = L_{\text{máximo}}$.

5 Problema unidimensional não-linear: equação de Burgers

Neste capítulo descreve-se a segunda classe de problemas tratados neste trabalho, que é o problema unidimensional não-linear. Neste caso estuda-se a equação de Burgers⁴ (TANNEHILL et al., 1997).

Este capítulo está dividido da seguinte forma: primeiramente estuda-se o modelo matemático com sua definição e solução analítica. Em seguida estudam-se os modelos numéricos com suas discretizações. Finalmente detalham-se os principais resultados numéricos com os esquemas usados: esquema de correção (CS) e esquema de aproximação completa (FAS), dados em Briggs et al. (2000), como: iterações internas, nível de malha, razão de engrossamento, variação de tolerância e estimativa inicial.

5.1 Modelo matemático

Tem-se a seguir um modelo matemático geral para problemas em regime permanente de escoamento de fluido incompressível com termo fonte dado pela equação 4.1.

Considerando-se $\Phi = u_1$ (velocidade), S_Φ o termo fonte de cada Φ com pelo menos o gradiente de pressão $-\vec{\nabla}p$, onde p é a pressão, e $\Gamma = \mu$ na a equação 4.1, obtém-se a equação da conservação da quantidade de movimento linear na direção coordenada x

$$\vec{\nabla} \cdot (\rho \vec{u} u_1) = \vec{\nabla} \cdot (\mu \vec{\nabla} u_1) + S_\Phi, \quad (5.1)$$

onde μ é a viscosidade do fluido.

Para o caso unidimensional com propriedades constantes e sem gradiente de pressão, considerando $u = u_1$, tem-se a seguinte simplificação da equação 5.1

$$Re \frac{d(uu)}{dx} = \frac{d^2u}{dx^2} + S$$

ou

$$Re \frac{du^2}{dx} = \frac{d^2u}{dx^2} + S, \quad (5.2)$$

⁴ Apresentando anisotropia de coeficientes, ou seja, anisotropia física, como definida na seção 3.4 do capítulo 3.

onde x é a coordenada cartesiana espacial, $S_\phi = S$ é o termo fonte, $Re = (\rho U_{\max} C)/\mu$ é o número adimensional de Reynolds (TANNEHILL et al., 1997), U_{\max} é velocidade máxima de referência e C é o comprimento do domínio. Neste texto, dá-se preferência pela segunda notação, dada pela equação 5.2, onde se utiliza u^2 . Esta equação recebe o nome de equação de Burgers (TANNEHILL et al., 1997).

Neste capítulo serão consideradas as condições de contorno de Dirichlet num domínio unidimensional unitário, $\{x \in R : 0 \leq x \leq 1\}$, conforme a figura 4.1, portanto o problema a ser resolvido será

$$\begin{cases} Re \frac{du^2}{dx} = \frac{d^2u}{dx^2} + S, & 0 < x < 1 \\ u(0) = 0, \quad u(1) = 1 \end{cases}. \quad (5.3)$$

Com a finalidade de buscar-se uma solução analítica para o problema dado pela equação 5.3, usa-se a técnica de solução fabricada (ROACHE, 1998) para equações diferenciais ordinárias não-lineares de segunda ordem com termo fonte. Nesta técnica, designa-se uma solução desejada para a equação e busca-se o termo fonte para que tal solução designada seja de fato a solução do problema em questão. Esta técnica será descrita com mais detalhes e aplicada à equação 5.3 no apêndice B desta Tese.

O número de Reynolds será um dado fixo, ou seja, $Re = 20$.

5.2 Modelos numéricos

Tem-se a seguir um detalhamento dos esquemas numéricos a serem adotados neste capítulo para a equação unidimensional não-linear estudada (equação de Burgers), além de suas soluções numéricas.

5.2.1 Discretização

Como no capítulo anterior, o método numérico a ser utilizado será o método de diferenças finitas (BURDEN e FAIRES, 1997; TANNEHILL et al., 1997). A discretização do

domínio será desenvolvida, fazendo-se uso de malhas estruturadas e uniformes, introduzindo uma malha que se denota por Ω^h .

Vamos detalhar os esquemas numéricos para o problema dado pela equação 5.3. Neste problema são utilizados dois esquemas distintos:

- 1) Primeiro são utilizados o esquema *Upwind* (UDS) para a derivada de primeira ordem (termo advectivo) e a diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema UDS/CDS pode ser visto em Tannehill et al. (1997). Para cada um dos $N - 1$ pontos interiores da malha, a equação resultante é

$$\left\{ \begin{array}{l} Re \frac{v_i^2 - v_{i-1}^2}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{array} \right., \quad (5.4)$$

onde v_i é uma aproximação (solução numérica) para a solução exata $u(x_i)$ e $S_i = S(x_i)$.

A linearização da equação 5.4, como em Ferziger e Peric (1999), é da forma

$$\left\{ \begin{array}{l} Re \frac{v_i^* v_i - v_{i-1}^* v_{i-1}}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{array} \right., \quad (5.5)$$

onde $v_i^2 \approx v_i^* v_i$ e o índice * denota valores que são obtidos da iteração anterior.

Rearranjando os termos da equação 5.5, obtém-se a forma geral linearizada dada pela equação 4.7, onde os coeficientes são dados por

$$a_p = \frac{2}{h^2} + \frac{Re v_i^*}{h}, \quad (5.6 \text{ a})$$

$$a_w = \frac{1}{h^2} + \frac{Re v_{i-1}^*}{h}, \quad (5.6 \text{ b})$$

$$a_e = \frac{1}{h^2}, \quad (5.6\text{ c})$$

$$b_p = S_i. \quad (5.7)$$

- 2) Após são utilizadas a diferença centrada com correção adiada (CDS_CA) para a derivada de primeira ordem (termo advectivo) e diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema CDS_CA/CDS pode ser visto em Lilek et al. (1997). Para cada um dos $N-1$ pontos interiores da malha, a equação resultante é

$$\begin{cases} Re \left[\frac{v_i^2 - v_{i-1}^2}{h} + \beta \left(\frac{v_{i-1}^2 - 2v_i^2 + v_{i+1}^2}{2h} \right)^* \right] = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, & 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{cases}, \quad (5.8)$$

onde $0 \leq \beta \leq 1$ e o índice * denota valores que são obtidos da iteração anterior.

A linearização da equação 5.8, como em Ferziger e Peric (1999), é da forma

$$\begin{cases} Re \left[\frac{v_i^* v_i - v_{i-1}^* v_{i-1}}{h} + \beta \left(\frac{v_{i-1}^2 - 2v_i^2 + v_{i+1}^2}{2h} \right)^* \right] = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, & 2 \leq i \leq N \\ v_1 = 0; \quad v_{N+1} = 1 \end{cases}, \quad (5.9)$$

onde $v_i^2 \approx v_i^* v_i$ e o índice * denota valores que são obtidos da iteração anterior, como já descrito anteriormente.

Rearranjando os termos da equação 5.9, obtém-se a forma geral dada pela equação 4.7, onde os coeficientes são dados pelas equações 5.6 a, b, c e

$$b_p = S_i - \beta Re \left(\frac{v_{i-1}^2 - 2v_i^2 + v_{i+1}^2}{2h} \right)^*. \quad (5.10)$$

Note que para $\beta = 0$ tem-se o esquema UDS/CDS e para $\beta = 1$ tem-se o CDS_CA/CDS.

Se v e f são denotados por $v = (v_1, \dots, v_{N+1})^t$ e $f = (f_1, \dots, f_{N+1})^t$, respectivamente, então os sistemas dados pelas equações 5.5 e 5.9, podem ser representados pelo sistema de equações algébricas do tipo dado pela equação 4.15.

Como já foi dito no capítulo 4, esta equação 4.15 (sistema de equações algébricas) é resolvida com o método direto TDMA, para obter-se informações sobre a performance do método neste tipo de problema. Resolve-se também este problema, apenas na malha mais fina, com o método *singlegrid* (SG) de Gauss-Seidel. No caso do método *multigrid*, os sistemas de equações são resolvidos com o método de Gauss-Seidel a cada nível de malha. Estes métodos foram descritos no apêndice A desta Tese.

5.3 Resultados numéricos com o esquema de correção (CS)

De acordo com a definição dada pela equação 4.16a, pode-se definir os seguintes engrossamentos que foram usados nesta seção: engrossamento padrão, ou seja, $r = 2$ e alguns engrossamentos fortes, ou seja, $r \in (2, \infty)$, usando-se razões puras da forma $r = p/q$, onde $q = 1$, tais como: $r = 3$, $r = 4$ e $r = 5$. Um pouco mais de 100 simulações foram realizadas com outras variantes, como: número de incógnitas (elementos) do problema (N), número de iterações internas (ITI), número de níveis de malhas (L), tolerâncias (ε) e estimativas iniciais (v). Somente alguns destes testes são apresentados nesta seção, pois são os testes mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

A tabela 4.1 mostra as razões de engrossamento e alguns N mínimos e máximos utilizados nos diversos testes para a equação de Burgers.

5.3.1 Iteração interna

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro iterações internas com os esquemas de correção (CS) e de aproximação completa (FAS) para as equações de difusão e de advecção-difusão. Concluiu-se que $ITI = 2r$ é o recomendado para o esquema CS e $ITI = 3r$ para o esquema FAS, onde r é a razão de engrossamento pura.

Aqui se analisam os resultados numéricos com relação a este mesmo parâmetro com o esquema CS para a equação de Burgers.

A figura 5.1 mostra a influência do número de iterações internas (ITI) sobre o tempo de CPU para a equação de Burgers para o esquema CS e esquema numérico UDS/CDS. Esta figura mostra também o número ótimo de iterações internas ($ITI_{óptimo}$) para diversas razões de engrossamento para um número fixo de elementos da malha mais fina (N) do problema. Pode-se notar que os valores de N da tabela 4.1, e que são utilizados por este gráfico, são distintos entre si. Esta figura serve apenas para mostrar o efeito qualitativo da variação do número de iterações internas. Nota-se ainda que o número de iterações internas afeta o tempo de CPU e que $ITI_{óptimo} \approx 2r$ para a maioria das razões estudadas.

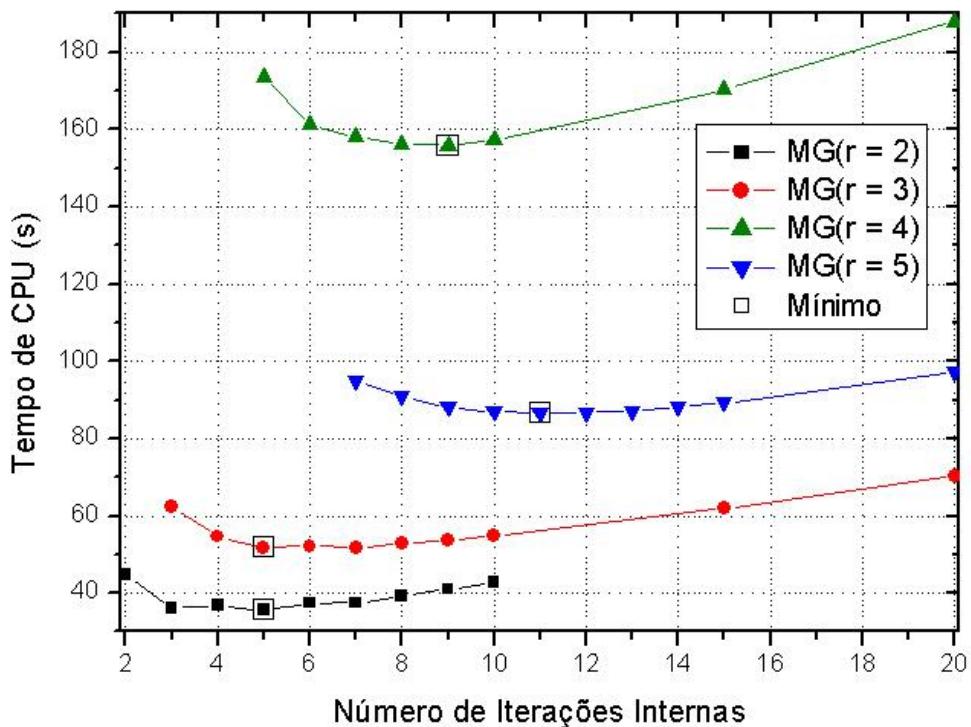


Figura 5.1: Tempo de CPU versus ITI para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS

Nota-se que para todas as razões estudadas, uma diminuição sensível do número de iterações internas, relativamente ao ponto mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, o aumento pequeno do número de iterações internas, relativamente ao ponto mínimo, aumenta pouco o tempo de CPU. Portanto

recomenda-se utilizar $ITI_{óptimo} = 2r$ para as razões puras; por exemplo, para $r = 2$, recomenda-se $ITI = 4$ para o problema em questão.

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema CS. Para este problema (equação de Burgers), como no capítulo 4, verificou-se que o número ótimo de iterações internas tem uma forte influência do número de vezes que o método de Gauss-Seidel necessita para atualizar o resíduo em todo o domínio unidimensional.

5.3.2 Nível de malha

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro número de níveis com os esquemas de correção (CS) e de aproximação completa (FAS) para as equações de difusão e de advecção-difusão. Concluiu-se que $L_{óptimo} = L_{máximo}$ é o recomendado, tanto para o esquema CS, como para o esquema FAS, onde $L_{máximo}$ é o número máximo de níveis. Aqui se analisam os resultados numéricos com relação a este mesmo parâmetro com o esquema CS para a equação de Burgers.

A figura 5.2 mostra a influência do número de níveis (L) sobre o tempo de CPU para a equação de Burgers. Esta figura mostra também o número ótimo de níveis ($L_{óptimo}$) para diversas razões de engrossamento para um número fixo de elementos da malha mais fina (N) do problema. Pode-se notar que os valores de N da tabela 4.1, e que são utilizados por este gráfico, são distintos entre si. Esta figura serve apenas mostrar o efeito qualitativo da variação do número de níveis. Nota-se ainda que o número de níveis pode afetar significativamente o tempo de CPU com o uso do esquema CS e que $L_{óptimo} \approx L_{máximo}$ para todas as razões estudadas. Este resultado é similar aos resultados de Tannehill et al. (1997), no caso da equação de Laplace bidimensional (caso linear); e Mesquita e De-Lemos (2004), no caso da equação de Navier-Stokes compressível bidimensional (caso não-linear), ambos para o caso onde $r = 2$.

Nota-se que, em geral o tempo de CPU tem uma taxa de decréscimo bastante acentuada até atingir os seus mínimos e, após estes pontos, a taxa de crescimento é bem menor até atingirem $L_{máximo}$. Portanto recomenda-se utilizar $L_{óptimo} = L_{máximo}$ para as razões puras; por exemplo, para $r = 2$ e $N = 2^{20} = 1.048.576$, recomenda-se usar $L = 20$, que é exatamente o $L_{máximo}$.

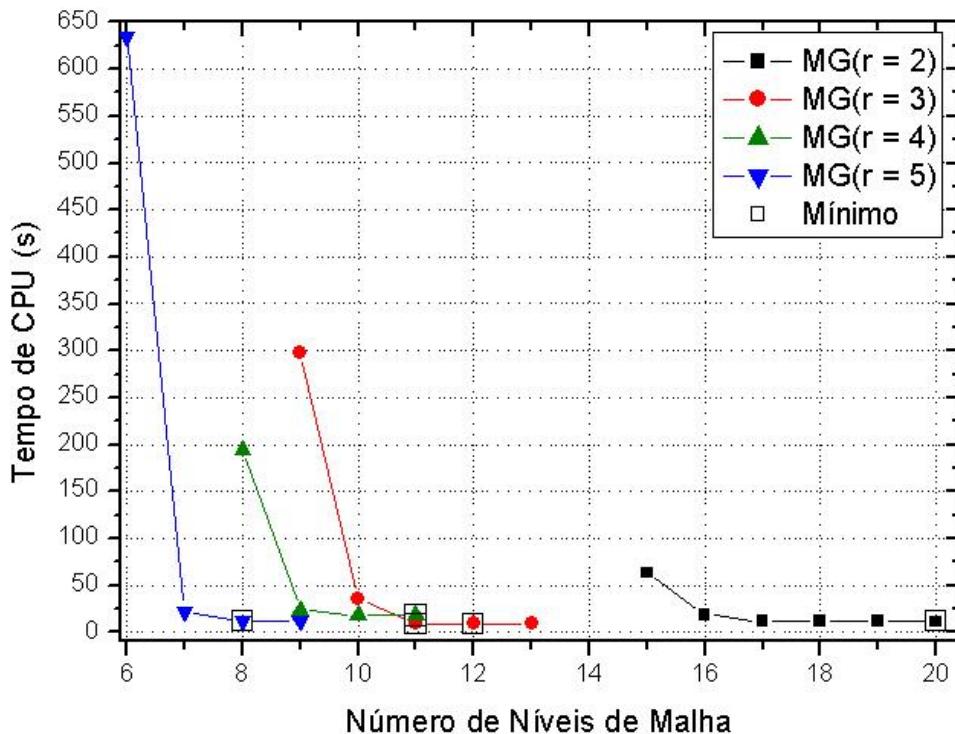


Figura 5.2: Tempo de CPU versus L para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS

Seja o problema de minimização do tempo de CPU em função do número de níveis para as diversas razões de engrossamento com o uso do esquema CS. Como no capítulo anterior, para este problema (equação de Burgers), verificou-se que o número ótimo de níveis tem uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas, ou seja, malhas com uma quantidade menor de elementos.

5.3.3 Razão de engrossamento

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro razão de engrossamento (r) com os esquemas de correção (CS) e de aproximação completa (FAS) para as equações de difusão e de advecção-difusão. Concluiu-se que $r = 2$ é o recomendado para o esquema CS e $r = 3$ é o recomendado para o esquema FAS. Aqui se analisam os resultados numéricos com relação a este mesmo parâmetro com o esquema CS para a equação de Burgers.

Nas simulações a seguir, considerem-se razões de engrossamento fortes usando-se razões puras dada pela equação 4.16a.

A figura 5.3 mostra as diversas razões de engrossamento para o método *multigrid* e também uma comparação entre os métodos *singlegrid* Gauss-Seidel e TDMA em função de diversos N (número de elementos da malha mais fina) para a equação estudada. Como foi dito no capítulo anterior, o método direto TDMA é utilizado apenas para efeito de comparação.

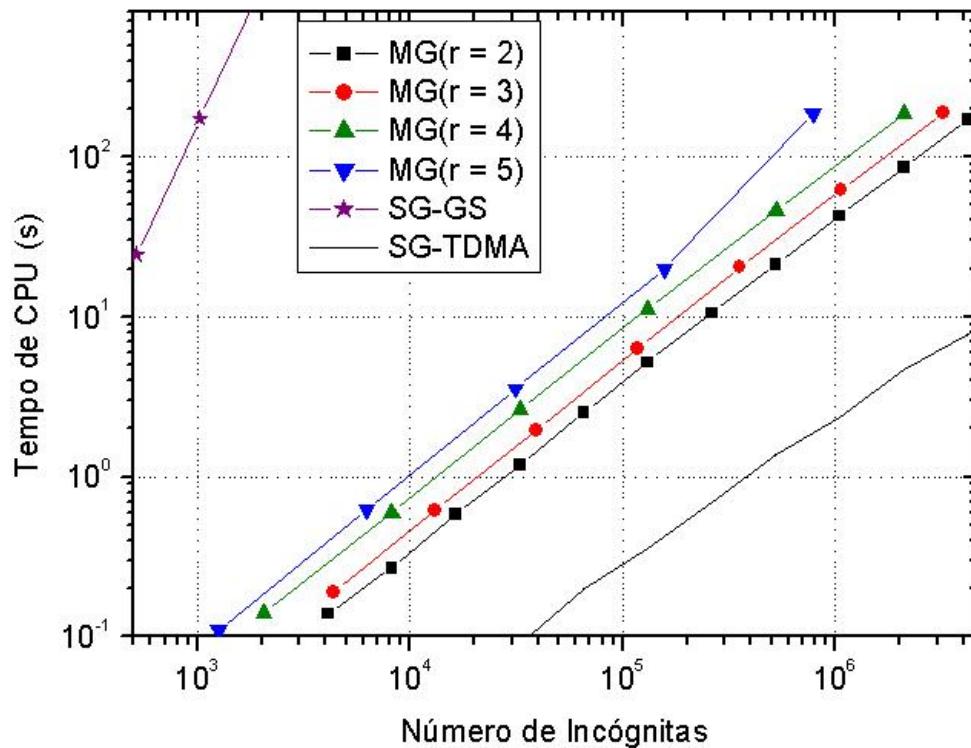


Figura 5.3: Tempo de CPU versus N para a equação de Burgers com o esquema CS, diversos r e com o esquema numérico UDS/CDS

Pode-se notar que o método TDMA é o mais eficiente de todos os métodos testados para o problema. Ele é seguido pelos métodos *multigrid* e finalmente pelo método de Gauss-Seidel. Portanto, para o problema unidimensional testado, recomenda-se o uso do método direto TDMA. Entretanto o objetivo nesta seção (esquema CS) e na próxima (esquema FAS) é verificar o efeito dos parâmetros do método *multigrid* geométrico no tempo de CPU com os diferentes esquemas (CS e FAS) em problemas unidimensionais não-lineares a fim de se obter informação para outros problemas que serão tratados nos capítulos seguintes.

Para os mesmos ε , v e N , com as razões de engrossamento testadas e o método *multigrid* com o uso do esquema CS, pode-se perceber que para o problema testado, tem-se o conjunto de desigualdades dado pela equação 4.23.

Portanto, no caso do método *multigrid* com o uso do esquema CS, recomenda-se o uso da razão $r = 2$ para a equação em questão.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento. Para este problema (equação de Burgers), verificou-se que a razão ótima tem uma forte influência de dois principais aspectos: a razão de queda do número de elementos de uma malha para a outra malha imediatamente mais grossa e a qualidade das informações (resíduos) passadas pela restrição e pela prolongação.

5.3.4 Tolerância e estimativa inicial

Nesta seção estudam-se também casos com variações tanto da tolerância ε como da estimativa inicial v .

Para a variação da tolerância, estudam-se os casos onde $\varepsilon = 10^{-4}$ e $\varepsilon = 10^{-10}$, além de $\varepsilon = 10^{-7}$ usado nos resultados anteriores. Os resultados dos estudos do número ótimo de iterações internas sofreram pouca variação sobre o tempo de CPU: cerca de 0,5% ao se adotar $ITI = 2r$.

Como para o problema de advecção-difusão dado no capítulo anterior, para a variação da estimativa inicial, estudam-se os casos onde $v = (0, 1/2, \dots, 1/2, 1)$ e $v = (0, 1, \dots, 1)$, além de $v = (0, 0, \dots, 1)$ usado nos resultados anteriores, devido ao fato de se ter a mesma solução analítica. Os resultados dos estudos do número ótimo de iterações internas sofreram uma variação sobre o tempo de CPU um pouco mais substancial: cerca de 15,75% ao se adotar $ITI = 2r$.

Em relação ao número de níveis, ao se adotar $L_{\text{ótimo}} = L_{\text{máximo}}$, a maior variação encontrada foi de 6,33%, independentemente de se estar variando a tolerância ou a estimativa inicial.

Isto mostra que o $ITI_{\text{ótimo}}$ é fracamente dependente da tolerância ε e fortemente dependente da estimativa inicial v e $L_{\text{ótimo}}$ é fracamente dependente da tolerância e estimativa inicial. Conclui-se que se pode utilizar $ITI_{\text{ótimo}} = 2r$ e $L_{\text{ótimo}} = L_{\text{máximo}}$ para a equação de Burgers com o uso do esquema CS.

5.4 Resultados numéricos com o esquema de aproximação completa (FAS)

De acordo com a definição dada pela equação 4.16a, pode-se definir os seguintes engrossamentos que foram usados nesta seção: $r = 2$, $r = 3$, $r = 4$ e $r = 5$ para o esquema UDS/CDS, e $r = 2$ e $r = 3$ para o esquema CDS_CA/CDS. Estes engrossamentos foram utilizados por terem apresentado boas características no que diz respeito à minimização do tempo de CPU quando se utilizou o esquema UDS/CDS na seção anterior. Um pouco mais de 500 simulações foram realizadas com outras variantes, como: número de incógnitas (elementos) do problema (N), número de iterações internas (ITI), número de níveis de malhas (L), tolerâncias (ε) e estimativas iniciais (v). Somente alguns destes testes são apresentados nesta seção, pois são os testes mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

A tabela 4.1 mostra as razões de engrossamento e algumas dimensões mínimas e máximas utilizadas nos diversos testes para a equação de Burgers.

Nas figuras 5.4 e 5.5 têm-se as soluções analítica e numérica para a equação de Burgers dada pela equação 5.3, usando-se os esquemas numéricos UDS/CDS e CDS_CA/CDS, dados pelas equações 5.5 e 5.9, respectivamente.

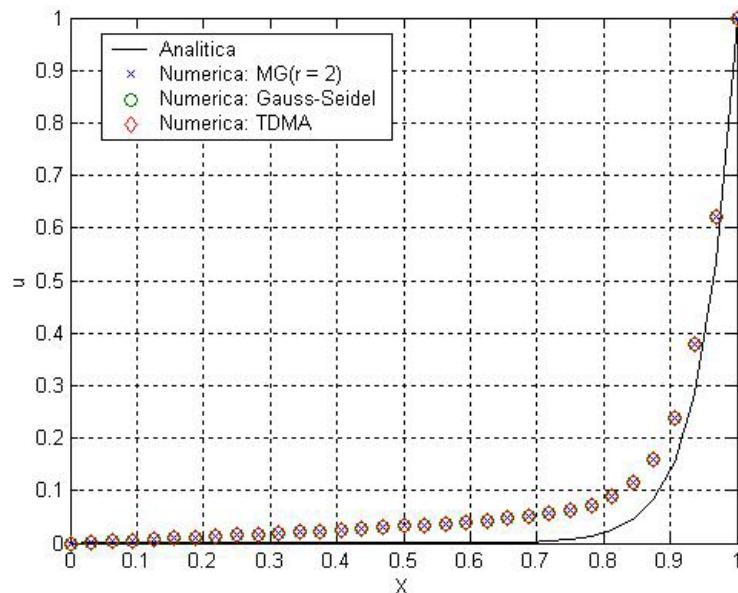


Figura 5.4: Soluções analítica e numéricas para $N = 32$ com esquema FAS e esquema numérico UDS/CDS para a equação de Burgers

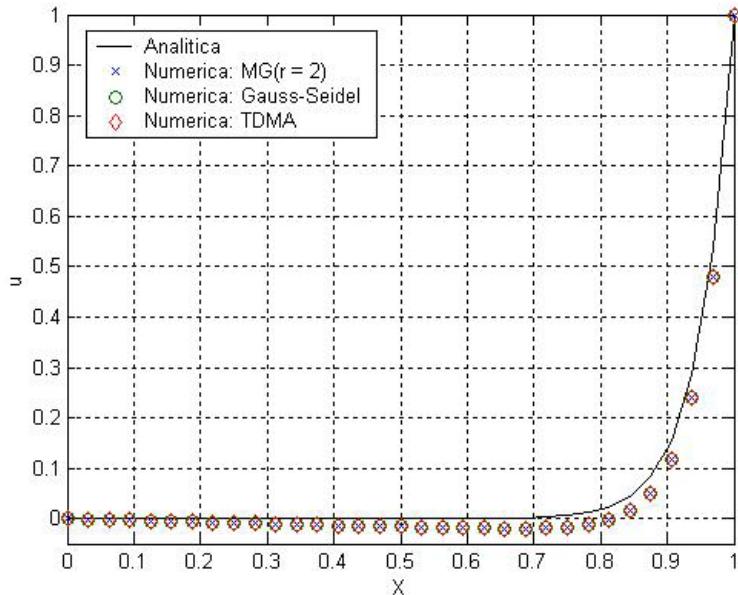


Figura 5.5: Soluções analítica e numéricas para $N = 32$ com esquema FAS e esquema numérico CDS_CA/CDS para a equação de Burgers

A solução analítica é comparada com as soluções numéricas usando-se o método *multigrid* com a razão de engrossamento $r = 2$ para ambos os esquemas numéricos.

Estipulou-se que as soluções numéricas deveriam atingir erro de máquina, eliminando-se assim o erro de iteração. Estipulou-se ainda o uso dos métodos de discretização UDS/CDS e CDS_CA/CDS, que são métodos de primeira e de segunda ordem, respectivamente (VERSTEEG e MALALASEKERA, 1995). Pode-se perceber visualmente com isto, que as soluções numéricas via métodos iterativos (comparando os esquemas numéricos UDS/CDS e CDS_CA/CDS, dados pelas figuras 5.4 e 5.5, respectivamente) não são coincidentes, sendo que, a solução que usa o esquema CDS_CA/CDS é a mais precisa por ser um esquema numérico de segunda ordem.

5.4.1 Iteração interna

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro iterações internas com os esquemas de correção (CS) e de aproximação completa (FAS) para as equações de difusão e de advecção-difusão. Concluiu-se que $ITI = 2r$ é o recomendado para o esquema CS e $ITI = 3r$ para o esquema FAS. Aqui, se analisam os resultados numéricos com relação a este mesmo parâmetro com o esquema FAS para a equação de Burgers.

Através da figura 5.1 concluiu-se que o número de iterações internas (ITI) influencia o tempo de CPU para a equação de Burgers com o uso do esquema CS. As figuras 5.6 e 5.7 mostram a influência de ITI sobre o tempo de CPU para a equação de Burgers com o uso do esquema FAS para os esquemas numéricos UDS/CDS e CDS_CA/CDS, respectivamente. Estas figuras mostram também que $ITI_{óptimo} \approx 3r - 1$.

Nota-se que, para todas as razões estudadas, uma diminuição sensível do número de iterações internas, relativamente ao ponto mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, o aumento pequeno do número de iterações internas, relativamente ao ponto mínimo, aumenta sensivelmente o tempo de CPU. Portanto recomenda-se utilizar $ITI = 3r$ para as razões puras; por exemplo, para $r = 2$, recomenda-se $ITI = 6$.

Para a equação de Burgers estudada na seção anterior, verificou-se $ITI = 2r$ para o esquema CS. Aqui, para o mesmo problema, verificou-se $ITI = 3r$ para o esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia o número ótimo de iterações internas para o método *multigrid*.

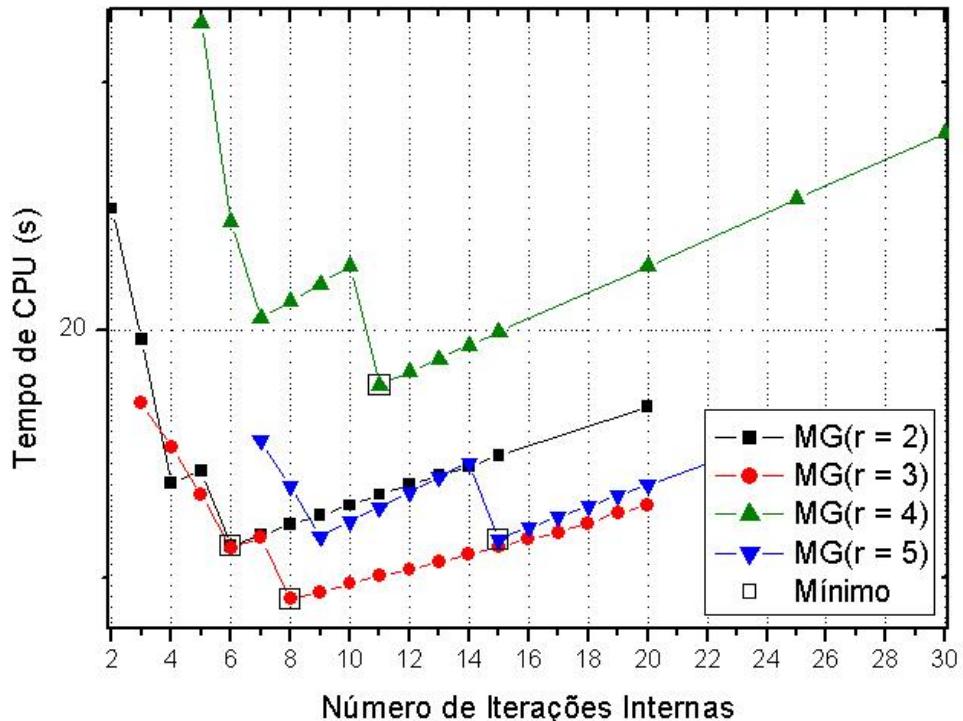


Figura 5.6: Tempo de CPU versus ITI para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico UDS/CDS

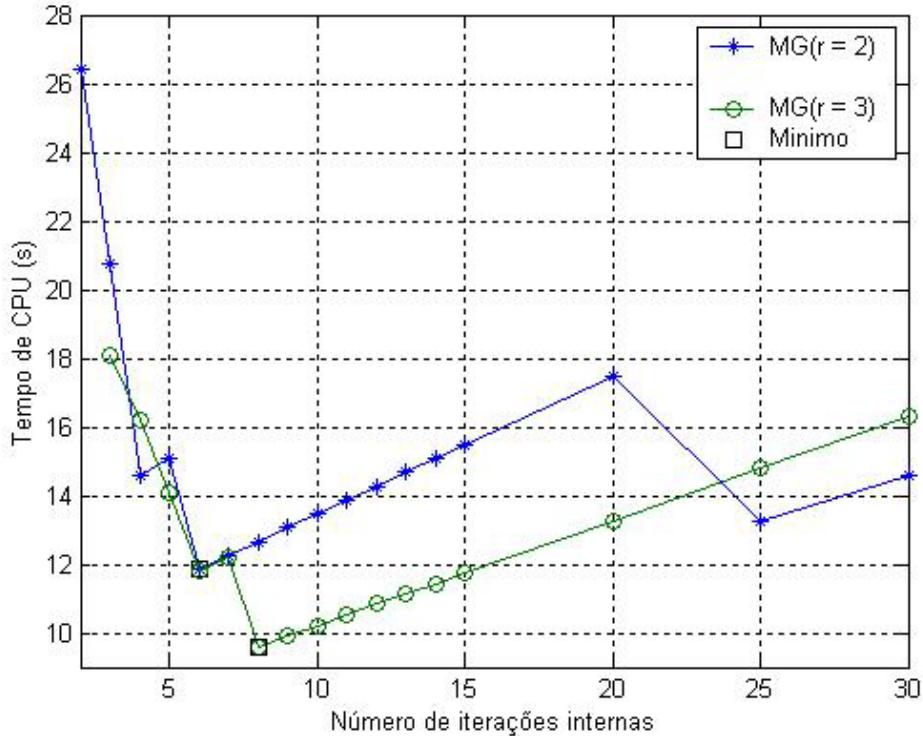


Figura 5.7: Tempo de CPU versus ITI para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico CDS_CA/CDS

Para os problemas lineares unidimensionais do capítulo anterior, verificou-se $ITI = 2r$ para o esquema CS e $ITI = 3r$ para o esquema FAS. Aqui, para problema não-linear unidimensional, verificaram-se os mesmos resultados tanto para o esquema CS, como para o esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia igualmente os problemas lineares e não-linear estudados no que diz respeito ao número ótimo de iterações internas.

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema FAS. Para este problema (equação de Burgers), verificou-se que o número ótimo de iterações internas tenha uma forte influência do número de vezes que o método de Gauss-Seidel necessita para atualizar o resíduo e a solução em todo o domínio unidimensional.

5.4.2 Nível de malha

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro número de níveis com os esquemas de correção (CS) e de aproximação completa (FAS) para

as equações de difusão e de advecção-difusão. Concluiu-se que $L_{óptimo} = L_{máximo}$ é o recomendado para os esquemas CS e FAS. Aqui, se analisam os resultados numéricos com relação a este mesmo parâmetro com o esquema FAS para a equação de Burgers.

Através da figura 5.2, concluiu-se que o número de níveis influencia o tempo de CPU para o esquema CS. As figuras 5.8 e 5.9 mostram a influência de L sobre o tempo de CPU para a equação de Burgers usando-se os esquemas numéricos UDS/CDS e CDS_CA/CDS, respectivamente. Pode-se notar que o número de níveis pode afetar significativamente o tempo de CPU e que $L_{óptimo} \approx L_{máximo}$ para todas as razões estudadas.

Nota-se que, em geral, o tempo de CPU tem uma taxa de decréscimo bastante acentuada até atingir os seus mínimos e, após estes pontos, a taxa de crescimento é bem menor até atingirem $L_{máximo}$. Portanto recomenda-se utilizar $L = L_{máximo}$ para as razões puras.

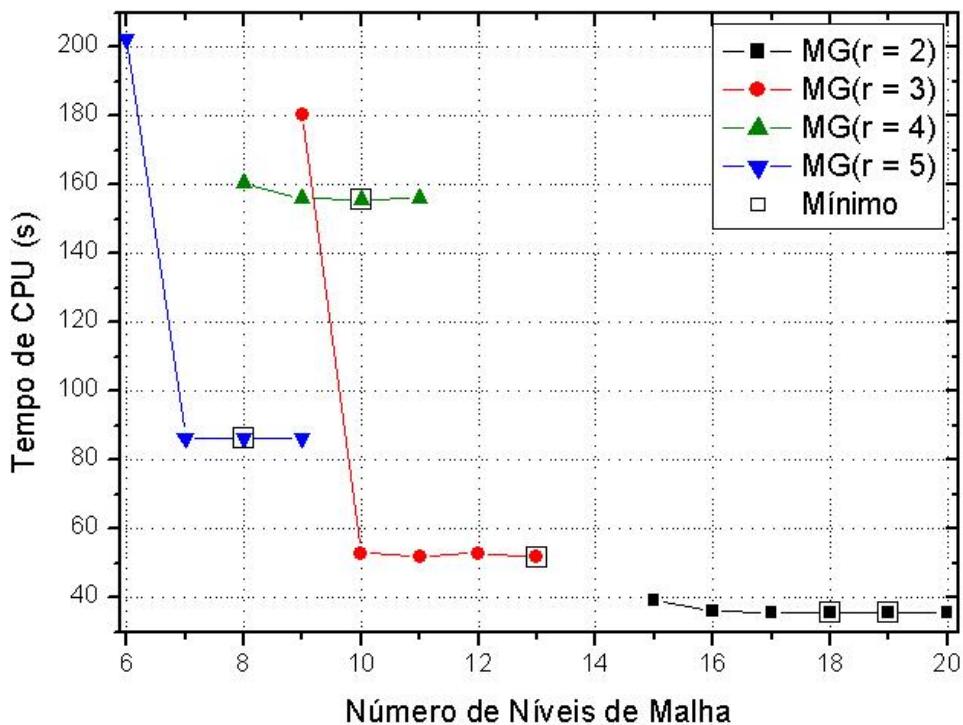


Figura 5.8: Tempo de CPU versus L para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico UDS/CDS

Para os problemas lineares unidimensionais do capítulo anterior, verificou-se $L = L_{máximo}$, tanto para o esquema CS como para o esquema FAS. Aqui, para problema não-linear unidimensional, também se verificou $L = L_{máximo}$ para os dois esquemas (CS e FAS).

Pode-se concluir que o algoritmo utilizado (CS ou FAS) não influencia o número ótimo de níveis para o método *multigrid*.

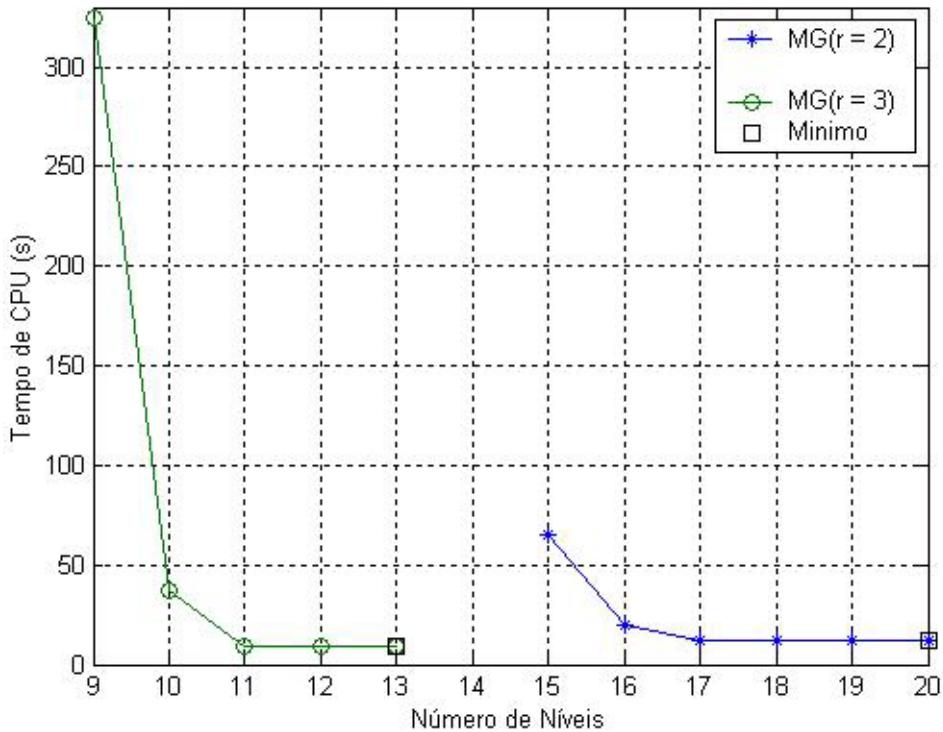


Figura 5.9: Tempo de CPU versus L para a equação de Burgers com o esquema FAS, diversos r e com o esquema numérico CDS_CA/CDS

Para os problemas lineares unidimensionais do capítulo anterior, verificou-se $L = L_{\text{máximo}}$ para os esquemas CS e FAS. Aqui, para problema não-linear unidimensional, se verificou $L = L_{\text{máximo}}$ para o esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia igualmente os problemas lineares e não-linear estudados no que diz respeito ao número ótimo de níveis.

Seja o problema de minimização do tempo de CPU em função do número de níveis para as diversas razões de engrossamento. Para este problema (equação de Burgers), verificou-se que o número ótimo de níveis tem uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas, ou seja, malhas com uma quantidade de pontos menor.

5.4.3 Razão de engrossamento

No capítulo anterior, analisaram-se os resultados numéricos com relação ao parâmetro razão de engrossamento com os esquemas de correção (CS) e de aproximação completa

(FAS) para as equações de difusão e de advecção-difusão. Concluiu-se que $r = 2$ é o recomendado para o esquema CS e $r = 3$ para o esquema FAS, onde r é a razão de engrossamento. Aqui se analisam os resultados numéricos com relação a este mesmo parâmetro com o uso do esquema FAS para a equação de Burgers.

As figuras 5.10 e 5.12 mostram o tempo de CPU para as razões de engrossamento consideradas para o método *multigrid* e também uma comparação entre métodos *singlegrid* Gauss-Seidel e TDMA em função de diversos N para a equação de Burgers com o esquema FAS e usando os esquemas numéricos UDS/CDS e CDS_CA/CDS, respectivamente. As figuras 5.11 e 5.13 mostram detalhes das figuras 5.10 e 5.12 para facilitar a visualização das curvas referentes às razões de engrossamento testadas nos métodos *multigrid*.

Pode-se notar novamente que o método TDMA é o mais eficiente dos métodos testados para ambos os esquemas numéricos. Ele é seguido pelos métodos *multigrid* e pelo método de Gauss-Seidel.

Para os mesmos ε , v e N , com as razões de engrossamento testadas para o método *multigrid* FAS, pode-se perceber que para a equação estudada, para a maioria dos pontos do domínio em estudo, tem-se para o esquema UDS/CDS

$$t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=2) < t_{CPU}(r=5), \quad (5.11)$$

e para o esquema CDS_CA/CDS

$$t_{CPU}(r=3) < t_{CPU}(r=2). \quad (5.12)$$

Portanto, no caso do método *multigrid* com o uso do esquema FAS, recomenda-se o uso da razão $r = 3$ para ambos os esquemas numéricos para a equação em questão. Esta constatação é inédita e inesperada, além de mostrar que a razão $r = 2$ perde a hegemonia perante outras razões.

Entretanto, Brandt (1977) em seu trabalho fez análises entre as razões de engrossamento $r = 2$, $r = 3$ e $r = 3/2$ para diversos problemas (mas não especificamente a equação 5.3) e constatou que a razão $r = 2$ é a recomendável. Briggs et al. (2000) afirmam que $r \neq 2$, em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas.

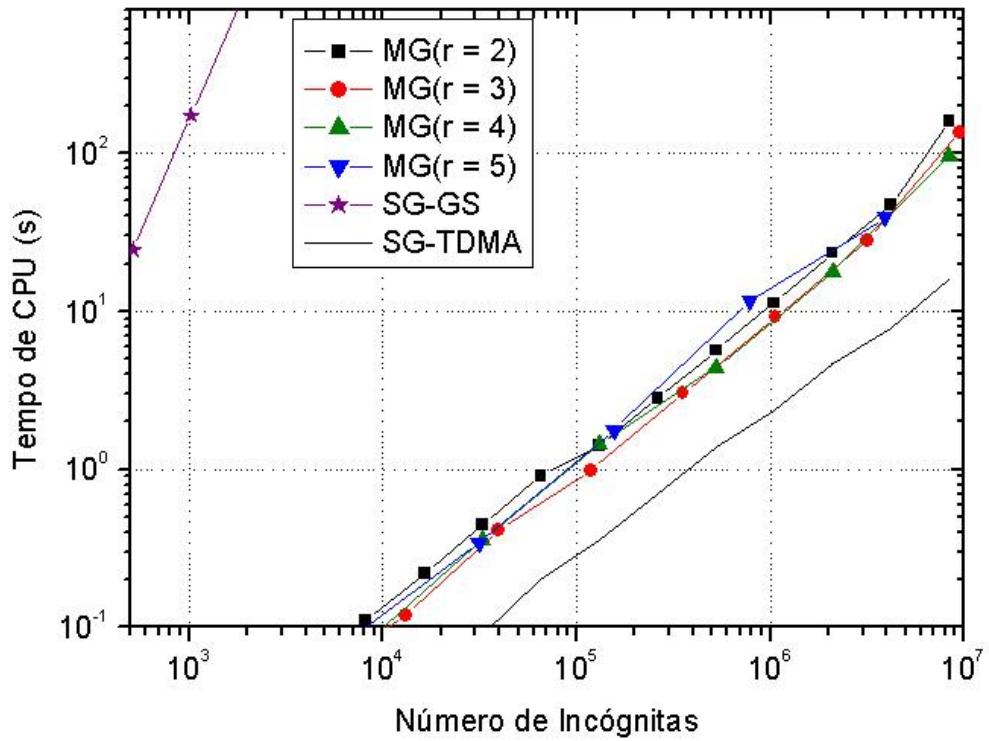


Figura 5.10: Tempo de CPU versus N para a equação de Burgers com o esquema FAS, diversos r , com o esquema numérico UDS/CDS e para diversos métodos

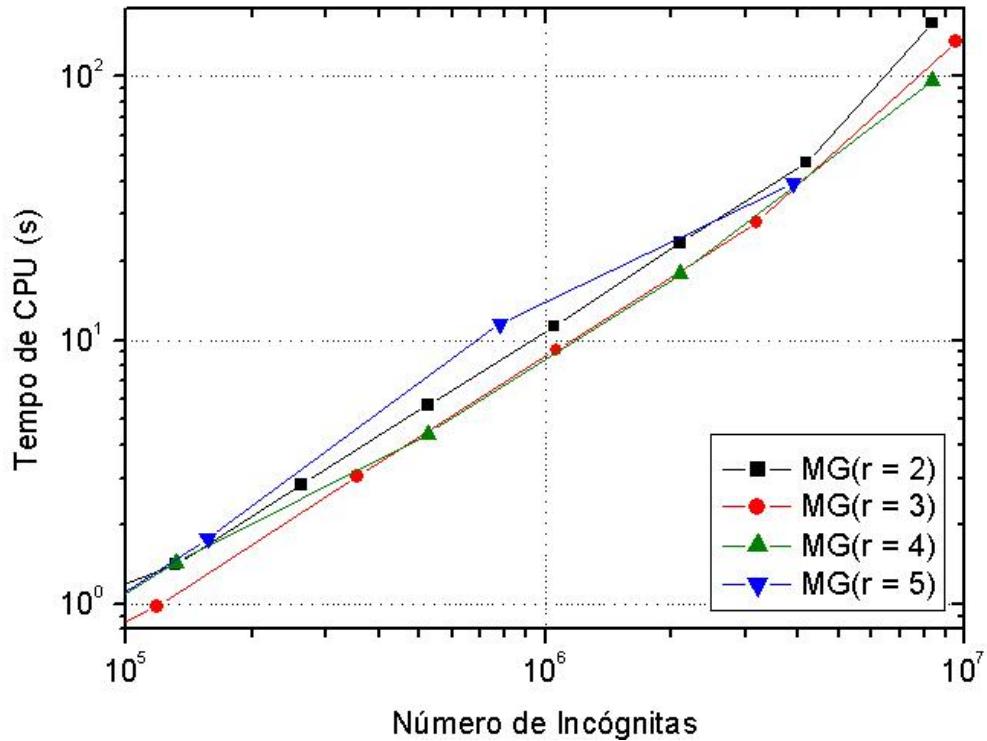


Figura 5.11: Detalhe da figura 5.10

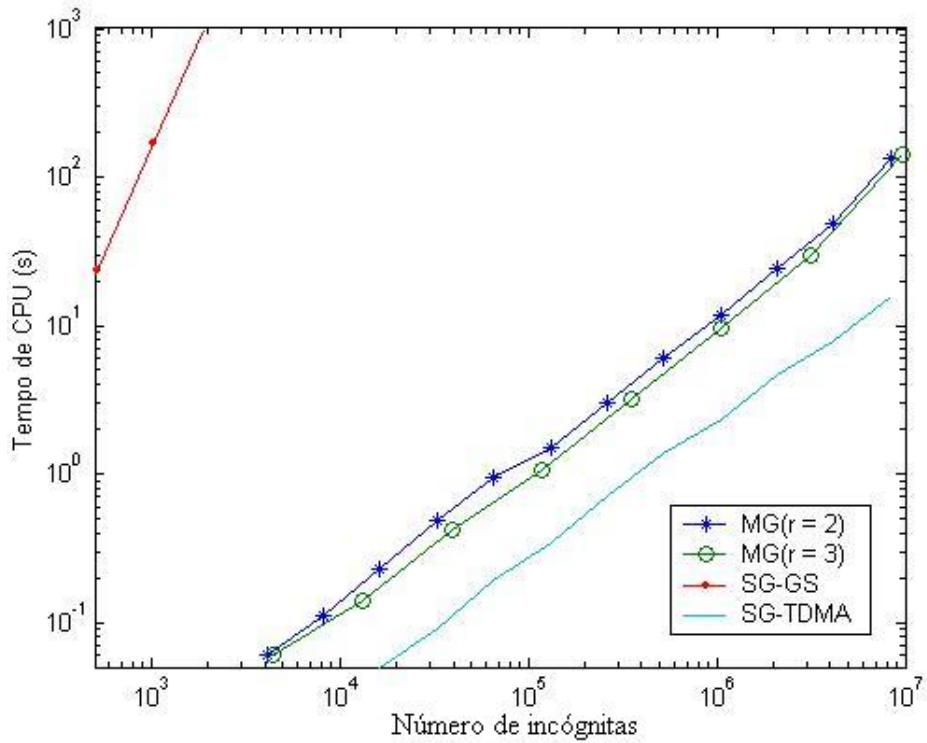


Figura 5.12: Tempo de CPU versus N para a equação de Burgers com o esquema FAS, $r = 2$ e $r = 3$, com o esquema numérico CDS_CA/CDS e para diversos métodos

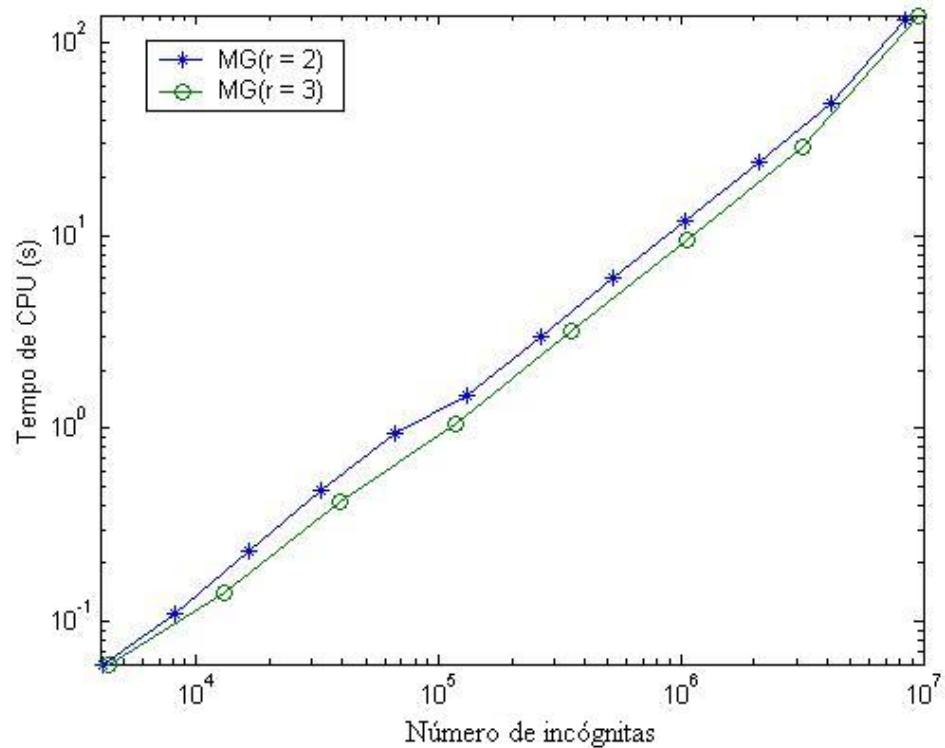


Figura 5.13: Detalhe da figura 5.12

Para os problemas lineares unidimensionais do capítulo anterior, verificou-se que $r = 2$ é a melhor razão de engrossamento para o esquema CS e $r = 3$ para o esquema FAS. Aqui, para o problema não-linear unidimensional estudado, verificaram-se os mesmos resultados tanto para o esquema CS, como para o esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia igualmente os problemas lineares e não-linear estudados no que diz respeito à razão de engrossamento ótima.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do algoritmo FAS. Para este problema (equação de Burgers), verificou-se que a razão ótima tem uma forte influência dos seguintes aspectos: a razão de queda do número de pontos de uma malha para a outra malha imediatamente mais grossa, a qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, o uso do algoritmo FAS, que transfere informações a respeito do resíduo e da solução para as demais malhas mais grossas.

5.4.4 Tolerância e estimativa inicial

No capítulo anterior, analisaram-se os resultados numéricos com relação à variação da tolerância e da estimativa inicial com os esquemas de correção (CS) e esquema de aproximação completa (FAS) para as equações de difusão e de advecção-difusão. Aqui, se analisam os resultados numéricos com relação a estes mesmos parâmetros com o uso do esquema FAS para a equação de Burgers.

Nesta seção estudam-se também casos com variações tanto da tolerância ε como da estimativa inicial v , como na seção anterior.

Para a variação da tolerância, os resultados dos estudos do número ótimo de iterações internas sofreram uma variação substancial sobre o tempo de CPU: cerca de 6,72% ao se adotar $ITI = 3r$.

Para a variação da estimativa inicial, estudam-se os casos onde $v = (0,1/2,\dots,1/2,1)$ e $v = (0,1,\dots,1)$, além de $v = (0,0,\dots,1)$ usado nos resultados anteriores. Os resultados dos estudos do número ótimo de iterações internas sofreram uma variação sobre o tempo de CPU um pouco mais substancial: de 0 à 27,04% ao se adotar $ITI = 3r$.

Em relação ao número de níveis, ao se adotar $L_{\text{ótimo}} = L_{\text{máximo}}$, a maior variação encontrada foi de 0,65%, independentemente de se estar variando a tolerância ou a estimativa inicial.

A tabela 5.1 mostra em suas colunas o estudo referente ao $ITI_{\text{ótimo}}$, $L_{\text{ótimo}}$ e t_{CPU} para as razões de engrossamento $r = 2$ (com $N = 1.048.576$ e $L_{\text{máximo}} = 20$) e $r = 3$ (com $N = 1.062.822$ e $L_{\text{máximo}} = 13$), para problemas variando a tolerância ε e considerando primeiramente a estimativa inicial fixa. Neste caso, adotou-se a estimativa inicial padrão $v = (0,0,\dots,1)$ usando-se o esquema numérico UDS/CDS.

Tabela 5.1: Variação de parâmetros de acordo com a variação da tolerância no esquema UDS/CDS

ε	$ITI_{\text{ótimo}}$		t_{CPU}		$L_{\text{ótimo}}$		t_{CPU}	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
10^{-4}	25	7	6,81	6,16	20	12	6,80	6,12
10^{-7}	6	8	11,30	9,17	18	10	11,30	9,16
10^{-10}	8	7	15,94	14,38	20	13	15,94	14,38

A tabela 5.2 mostra em suas colunas, o estudo referente ao $ITI_{\text{ótimo}}$, $L_{\text{ótimo}}$ e t_{CPU} para as razões de engrossamento $r = 2$ (com $N = 1.048.576$ e $L_{\text{máximo}} = 20$) e $r = 3$ (com $N = 1.062.822$ e $L_{\text{máximo}} = 13$), para problemas variando as estimativas iniciais e considerando agora a tolerância fixa. Neste caso, adotou-se a tolerância padrão $\varepsilon = 10^{-7}$ usando-se o esquema numérico UDS/CDS.

Tabela 5.2: Variação de parâmetros de acordo com a variação da estimativa inicial no esquema UDS/CDS

v	$ITI_{\text{ótimo}}$		t_{CPU}		$L_{\text{ótimo}}$		t_{CPU}	
	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$	$r = 2$	$r = 3$
$v = (0,0,\dots,1)$	6	8	11,30	9,17	20	12	11,30	9,16
$v = (0,1/2,\dots,1/2,1)$	6	9	11,31	9,50	20	10	11,31	9,48
$v = (0,1,\dots,1)$	7	10	11,72	9,80	20	13	11,72	9,80

Isto mostra que, como na seção anterior, $ITI_{\text{ótimo}}$ é fracamente dependente da tolerância e fortemente dependente da estimativa inicial e $L_{\text{ótimo}}$ é fracamente dependente da tolerância e estimativa inicial. Conclui-se que se pode utilizar $ITI_{\text{ótimo}} = 3r$ e $L_{\text{ótimo}} = L_{\text{máximo}}$ para a equação de Burgers com o uso do esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia igualmente os problemas lineares e não-linear estudados no que diz respeito à tolerância e estimativa inicial.

5.5 Esquema de correção (CS) *versus* esquema de aproximação completa (FAS)

Usando-se as figuras 5.3 e 5.10 para uma comparação entre os esquemas multigrid CS e FAS para as razões de engrossamento $r = 2$ e $r = 3$, pode-se notar que o esquema FAS é mais rápido do que o esquema CS em qualquer N para as duas razões analisadas. Por exemplo, para a malha de 4.194.304 elementos o tempo de CPU do MG-FAS($r = 2$) e MG-CS($r = 2$) é de 47.25 s e 142.47 s, respectivamente, ou seja, o MG-FAS($r = 2$) é cerca e 3 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 3.188.646 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 3$) é de 28.19 s e 156.74 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca e 5 vezes mais rápido do que o MG-CS($r = 3$).

Na figura 5.14 se faz uma comparação entre os esquemas *multigrid* CS (com a melhor razão de engrossamento, $r = 2$) e FAS (com a melhor razão de engrossamento, $r = 3$) em função de diversos valores de N . Pode-se notar que o esquema FAS é mais rápido do que o CS em qualquer N .

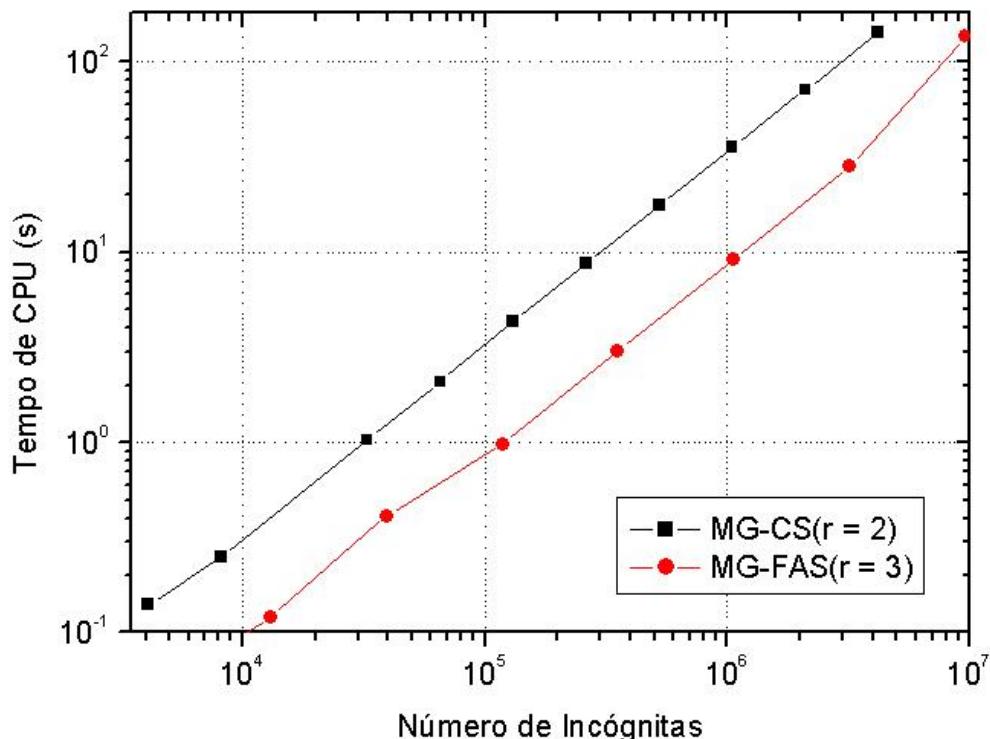


Figura 5.14: Tempo de CPU *versus* N para a equação de Burgers com os esquemas CS e FAS, $r = 2$ e $r = 3$ e com o esquema numérico UDS/CDS

Usando-se ajuste geométrico de curvas, o tempo de CPU do MG-FAS($r = 3$) pode ser aproximado por $t_{CPU} = 8,79 \times 10^{-6} N^{1,01}$ e MG-CS($r = 2$) pode ser aproximado por $t_{CPU} = 2,60 \times 10^{-5} N^{1,03}$.

Pode-se notar que o esquema MG-FAS($r = 3$) é mais rápido do que o esquema MG-CS($r = 2$) em qualquer N . Por exemplo, para a malha de 10^5 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 0,97 s e 3,78 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,0 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 10^6 elementos o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 9,92 s e 40,69 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,1 vezes mais rápido do que o MG-CS($r = 2$). Portanto, recomenda-se o uso do esquema FAS com a razão $r = 3$ para a equação de Burgers.

Seja o problema de minimização do tempo de CPU em função do uso dos diferentes esquemas CS e FAS. Para este problema, verificou-se que o esquema ótimo tem uma forte influência da qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, pela transferência de informações a respeito do resíduo e da solução para as malhas mais grossas. Entretanto, no presente trabalho verificou-se que o número de ciclos ou iterações externas do esquema FAS é significativamente menor do que o esquema CS. Por exemplo, para a equação de Burgers e para a malha de 4.194.304 elementos, o número de ciclos ou iterações externas (ITE) do MG-FAS($r = 2$) e MG-CS($r = 2$) é de 3 e 22, respectivamente.

5.6 Conclusão do capítulo 5

Neste capítulo foi resolvido numericamente um problema: equação de Burgers unidimensional. Esta equação é não-linear e ela representa o problema de escoamento de fluido. Os esquemas numéricos de aproximação usados foram: UDS/CDS e CDS_CA/CDS. Foram usados dois esquemas do método *multigrid*: esquemas CS (*Correction Scheme*) e FAS (*Full Approximation Scheme*).

Com base nos resultados deste capítulo, verificou-se que:

1. Para os mesmos dados (ε , v e N), o esquema FAS é mais rápido do que o esquema CS;
2. O número de iterações internas (ITI) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) influencia o

$ITI_{\text{óptimo}}$. Para o uso do esquema CS, $ITI_{\text{óptimo}} \approx 2r$, e o uso do esquema FAS, $ITI_{\text{óptimo}} \approx 3r$;

3. O número de níveis de malhas (L) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) não tem muita influência no $L_{\text{óptimo}}$. Para ambos os esquemas $L_{\text{óptimo}} \approx L_{\text{máximo}}$;
4. Para os mesmos dados (ε , v e N), com $L = L_{\text{máximo}}$ para cada N , entre as razões de engrossamento de malhas testadas ($r = 2, 3, 4$ e 5), o esquema CS é mais rápido com $r = 2$; já o esquema FAS é mais rápido com $r = 3$;
5. O esquema FAS com razão de engrossamento $r = 3$ é aproximadamente 4 vezes mais rápido do que o esquema CS com razão de engrossamento $r = 2$;
6. O comportamento qualitativo dos parâmetros estudados do método *multigrid* é idêntico para os problemas lineares do capítulo anterior e o problema não-linear deste capítulo.

Recomendações práticas:

Com base nos resultados deste capítulo são feitas as seguintes recomendações aos usuários do método *multigrid* para problemas iguais ou similares ao estudado.

Problemas unidimensionais não-lineares:

- Esquema: FAS;
- Razão de engrossamento: $r = 3$;
- Número de iterações internas: $ITI = 9$;
- Número de níveis de malhas: $L = L_{\text{máximo}}$.

6 Problema bidimensional linear isotrópico: equação de Laplace

Neste capítulo descreve-se a terceira classe de problemas tratados neste trabalho, que é o problema bidimensional linear em malhas isotrópicas⁵. Neste caso estuda-se a equação de Laplace (TANNEHILL et al., 1997).

Este capítulo está dividido da seguinte forma: primeiramente estuda-se o modelo matemático com sua definição e solução analítica. Em seguida estuda-se o modelo numérico com sua discretização. A seguir detalham-se os principais resultados numéricos com os algoritmos *multigrid* usados: esquema de correção (CS) e esquema de aproximação completa (FAS), ambos dados em Briggs et al. (2000), iterações internas, níveis de malha, razão de engrossamento e *solvers*. Finalmente, faz-se uma comparação entre os esquemas CS e FAS.

6.1 Modelo matemático

Tem-se a seguir um modelo matemático geral para problemas em regime permanente de advecção-difusão com termo fonte dado pela equação 4.1.

Como no capítulo 4, considerando $\Phi = T$ (temperatura), $S_\Phi = \dot{q}$ (geração de calor) e $\Gamma = \kappa/c_p$ na equação 4.1, obtém-se a equação da conservação da energia dada pela equação 4.2.

Para o caso bidimensional, com propriedades constantes e não havendo escoamento, ou seja, velocidade constante $\vec{u} = 0$, tem-se a seguinte simplificação da equação 4.2:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{q}}{\kappa} = 0,$$

ou ainda,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + f(x) = 0, \quad (6.1)$$

⁵ Isotropia de malha, ou seja, isotropia geométrica, como definida na seção 3.4 do capítulo 3.

onde x e y são as coordenadas cartesianas espaciais e $f(x)$ é o termo fonte. Esta equação recebe o nome de equação de difusão e é conhecida também como equação de Poisson (TANNEHILL et al., 1997). Se $f(x)=0$, ela é conhecida como equação de Laplace (TANNEHILL et al., 1997).

Neste trabalho serão consideradas condições de contorno de Dirichlet num domínio bidimensional unitário, $\{(x, y) \in R^2 : 0 \leq x, y \leq 1\}$, conforme a figura 6.1.

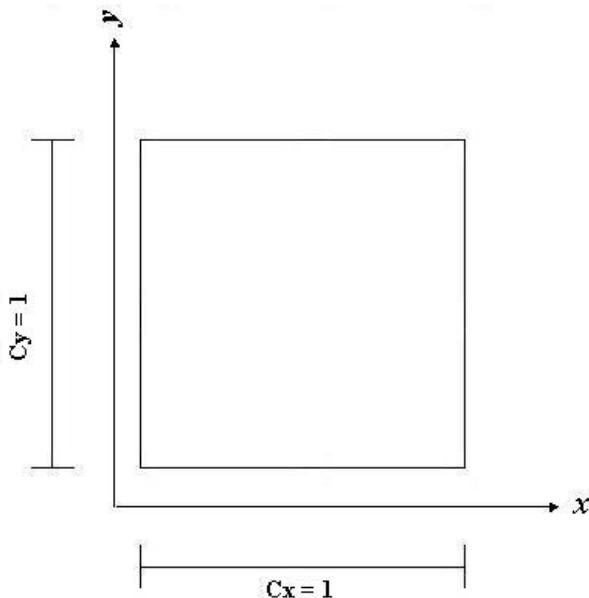


Figura 6.1: Domínio bidimensional de cálculo para a equação de Laplace

Portanto o problema linear de condução de calor bidimensional (equação de Laplace) com condições de contorno de Dirichlet, em coordenadas cartesianas, considerado neste e no próximo capítulo é (Maliska, 2004)

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, & 0 < (x, y) < 1 \\ T(x, 0) = T(0, y) = T(1, y) = 0, & T(x, 1) = \text{sen}(\pi x) \end{cases} . \quad (6.2)$$

Com a finalidade de buscar-se uma solução analítica para o problema dado pela equação 6.2, usa-se a técnica de solução por separação de variáveis para as equações diferenciais parciais elípticas (GREENBERG, 1998). Esta técnica está descrita no apêndice B desta Tese.

6.2 Modelo numérico

Tem-se a seguir um detalhamento do esquema numérico a ser adotado neste capítulo para a equação bidimensional linear estudada (equação de Laplace).

6.2.1 Discretização

Como nos capítulos anteriores, o método numérico a ser utilizado será o método de diferenças finitas (BURDEN e FAIRES, 1997; TANNEHILL et al., 1997). A discretização do domínio será desenvolvida fazendo-se uso de malhas estruturadas e uniformes, onde o domínio $\{(x, y) \in R^2 : 0 \leq x, y \leq 1\}$ é particionado em subconjuntos através de um número de incógnitas (ou número de pontos) dado por $N = N_x N_y$, onde N_x e N_y são os números de pontos nas direções coordenadas x e y , respectivamente. Como visto na seção 1.1, isto introduz uma malha, conforme a figura 6.2, com os pontos

$$(x_i, y_j) = ((i-1)h_x, (j-1)h_y), \text{ com } h_x = \frac{1}{N_x - 1} \text{ e } h_y = \frac{1}{N_y - 1}, \quad (6.3)$$

onde $i = 1, \dots, N_x$, $j = 1, \dots, N_y$ e h_x e h_y são os comprimentos de cada elemento nas direções coordenadas x e y , respectivamente.

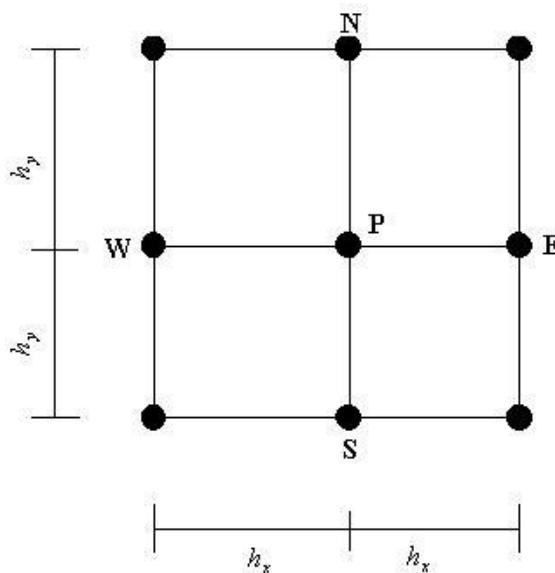


Figura 6.2: Malha 2D uniforme

Neste caso, estabelece-se uma malha com elementos de tamanho h_x por h_y que se denota por Ω^h , com razão de aspecto (RA) definida nas seções 1.1 e 3.4 e dada novamente neste capítulo por questões didáticas

$$RA = \frac{h_x}{h_y}. \quad (6.4)$$

Se $RA=1$, o problema chama-se isotrópico; caso contrário, anisotrópico⁶ (Briggs et al., 2000; Dendy et al., 1989). Neste capítulo trata-se apenas de problemas isotrópicos. Problemas anisotrópicos com $0 < RA \ll 1$ ou $RA \gg 1$ serão tratados no próximo capítulo.

Vamos detalhar o esquema numérico para o problema dado pela equação 6.2. Neste esquema numérico é utilizada a diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). O esquema CDS pode ser visto em Tannehill et al. (1997). Para cada um dos $(N_x - 2) \times (N_y - 2)$ pontos interiores da malha, a equação 6.2 resulta em

$$\left\{ \begin{array}{l} \frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{h_x^2} + \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{h_y^2} = 0, \quad 2 \leq i \leq N_x - 1, \quad 2 \leq j \leq N_y - 1 \\ v_{i,1} = v_{1,j} = v_{N_x,1} = 0, \quad v_{i,N_y} = \text{sen}(\pi x_{i,N_y}) \end{array} \right. , \quad (6.5)$$

onde $v_{i,j}$ é uma aproximação (solução numérica) para a solução exata $T(x_i, y_j)$.

Rearranjando os termos da equação 6.5, obtém-se a forma geral

$$a_p v_P = a_n v_N + a_s v_S + a_e v_E + a_w v_W + b_p, \quad (6.6)$$

onde os coeficientes são dados por

$$a_p = \frac{2}{h_x^2} + \frac{2}{h_y^2}, \quad (6.7 \text{ a})$$

$$a_n = a_s = \frac{1}{h_y^2}, \quad (6.7 \text{ b})$$

⁶ Isotropia geométrica.

$$a_e = a_w = \frac{1}{h_x^2}, \quad (6.7 \text{ c})$$

$$b_p = 0. \quad (6.8)$$

Se v e f são denotados por $v = (v_1, \dots, v_N)'$ e $f = (f_1, \dots, f_N)'$, respectivamente, onde f é o vetor independente formado pelos termos b_p e v é o vetor de incógnitas, então o sistema dado pela equação 6.5, pode ser representado por um sistema de equações algébricas do tipo da equação 4.15, onde A é uma matriz pentadiagonal N por N , simétrica e definida positiva (Briggs et al., 2000; Burden e Faires, 1997).

A equação 4.15 é resolvida com o método *multigrid* fazendo-se uso dos dois esquemas (CS e FAS) citados no capítulo 4. Neste caso, os sistemas de equações do tipo da equação 4.15, onde f agora representa o termo fonte (resíduo ou solução aproximada) a cada nível de malha, são resolvidos com os *solvers* Gauss-Seidel (GS), MSI (descritos no apêndice A) e ADI. Resolve-se também este problema na malha mais fina, usando-se o método padrão de malha única, aqui chamado de método *singlegrid*: GS, MSI e ADI.

6.3 Resultados numéricos com o esquema de correção (CS)

Neste e no próximo capítulo, os operadores de transferência entre malhas de restrição e prolongação, são dados respectivamente pelas equações 3.2a e 3.6a.

Os seguintes engrossamentos foram usados nesta seção: $r = 2, 3, 4$ e 5 . Estes engrossamentos foram utilizados por terem apresentado boas características no que diz respeito à minimização do tempo de CPU para os problemas unidimensionais (lineares e não-linear), apresentados nos capítulos anteriores e para os dois esquemas (CS e FAS). Um pouco mais de 300 simulações foram realizadas com as seguintes variantes: número de incógnitas (nós) da malha mais fina (N), número de iterações internas (ITI), número de níveis de malhas (L), *solvers* (GS, MSI e ADI) e razão de engrossamento (r). São apresentados nesta seção os resultados mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

A tabela 6.1 mostra alguns valores de N , mínimos e máximos nos diversos testes para os esquemas CS e FAS.

Tabela 6.1: Algumas malhas para cada uma das razões testadas para os esquemas CS e FAS

r	$N_{mínimo}$	$N_{máximo}$
SG	$3 \times 3 = 9$	$513 \times 513 = 263.169$
2	$3 \times 3 = 9$	$2049 \times 2049 = 4.198.401$
3	$3 \times 3 = 9$	$1459 \times 1459 = 2.128.681$
4	$3 \times 3 = 9$	$2049 \times 2049 = 4.198.401$
5	$3 \times 3 = 9$	$1251 \times 1251 = 1.565.001$

Na tabela 6.2 a solução analítica é comparada com a solução numérica via norma infinito ($\|\cdot\|_\infty$) para as diversas razões de engrossamento para os métodos *multigrid* e para o método *singlegrid*. Onde a norma vetorial $\|\cdot\|_\infty$, definida na seção 2.2, é repetida aqui por questão didática

$$\|e\|_\infty = \max_{1 \leq i \leq N} |e_i|,$$

onde e_i é a i -ésima coordenada do vetor e .

Tabela 6.2: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema CS

r	N	$\ T_{analítica} - T_{numérica}\ _\infty$
SG	$513 \times 513 = 263.169$	$3,20 \times 10^{-6}$
2	$2049 \times 2049 = 4.198.401$	$3,70 \times 10^{-7}$
3	$1459 \times 1459 = 2.128.681$	$1,33 \times 10^{-7}$
4	$2049 \times 2049 = 4.198.401$	$9,27 \times 10^{-8}$
5	$1251 \times 1251 = 1.565.001$	$1,82 \times 10^{-7}$

Em todos os casos (*multigrid* para as várias razões de engrossamento e *singlegrid*), o MSI foi o solver utilizado.

6.3.1 Iteração interna

A figura 6.3 mostra a influência do número de iterações internas (*ITI*) sobre o tempo de CPU para alguns valores de N , solver MSI e razão de engrossamento $r = 2$. Verificou-se que, para cada malha, em geral o menor tempo de CPU ocorre com o menor valor de *ITI*. Aumentando-se o valor de *ITI*, em geral aumenta-se significativamente o tempo de CPU. Este

comportamento da figura 6.3 também ocorre com os outros *solvers* testados (GS e ADI) e para outras razões de engrossamento.

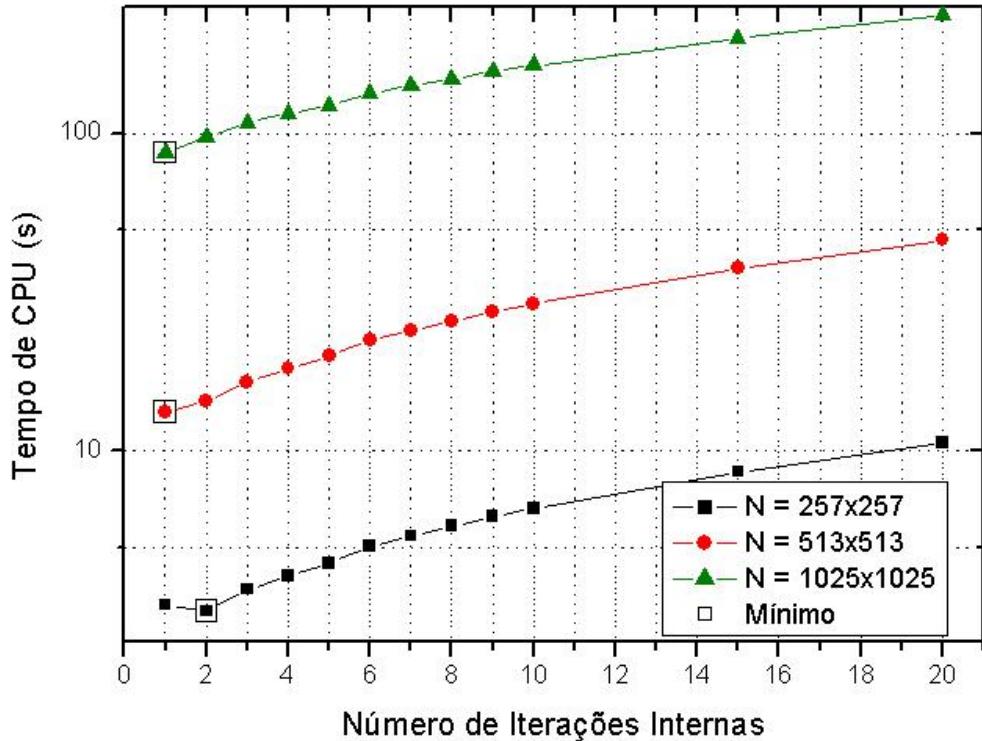


Figura 6.3: Tempo de CPU versus ITI para a equação de Laplace com o esquema CS, com o esquema numérico CDS/CDS, $r = 2$ e solver MSI

Para os três *solvers* testados e para a razão de engrossamento $r = 2$, a tabela 6.3 mostra o número ótimo de iterações internas ($ITI_{\text{ótimo}}$), que é o ITI que resulta no menor tempo de CPU. Observa-se que N e o *solver* influenciam pouco o valor de $ITI_{\text{ótimo}}$. Para N muito grande e $r = 2$, $ITI_{\text{ótimo}} = 1$ com qualquer *solver*.

Tabela 6.3: Número ótimo de iterações internas ($ITI_{\text{ótimo}}$) para diversos solvers, esquema CS e $r = 2$

N	GS	MSI	ADI
257×257	2	2	1
513×513	1	1	1
1025×1025	1	1	1

Para o *solver* MSI e para as diversas razões de engrossamento $r = 2, 3, 4$ e 5 , a tabela 6.4 mostra o número ótimo de iterações internas ($ITI_{\text{ótimo}}$), mostra-se também o $N_{\text{máximo}}$ utilizado por cada razão para se chegar a tal conclusão. Observa-se que a razão de

engrossamento influencia pouco o valor de $ITI_{\text{ótimo}}$. Estes valores de $ITI_{\text{ótimo}}$ diferem daqueles encontrados em problemas unidimensionais lineares, dados no capítulo 4, onde $ITI_{\text{ótimo}} = 3$ para o *solver* GS, $r = 2$ e o mesmo método *multigrid*.

Tabela 6.4: Número ótimo de iterações internas ($ITI_{\text{ótimo}}$) para solver MSI, esquema CS e diversas razões de engrossamento

R	$N_{\text{máximo}}$	$ITI_{\text{ótimo}}$
2	1025x1025	1
3	1459x1459	2
4	2049x2049	2
5	1251x1251	2

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema CS. Para este problema, verificou-se que o número ótimo de iterações internas sofre pouca influência do número de vezes que o *solver* necessita para atualizar o resíduo em todo o domínio bidimensional.

6.3.2 Nível de malha

A figura 6.4 mostra a influência do número de níveis (L) sobre o tempo de CPU, para $N = 513 \times 513$ nós e os três *solvers* testados (GS, MSI e ADI) e a razão de engrossamento $r = 2$. Para esta malha $L_{\text{máximo}} = 9$, isto é, usando-se a razão de engrossamento $r = 2$, para resolver a malha mais fina de 513x513 nós, pode-se usar no máximo mais 8 níveis, que são: 257x257, 129x129, 65x65, 33x33, 17x17, 9x9, 5x5 e 3x3, que representa a malha mais grossa, que é apenas um ponto interno. Portanto, $L_{\text{máximo}}$ é o número máximo possível de níveis.

Verificou-se que, para cada *solver*, o número ótimo de níveis de malha ($L_{\text{ótimo}}$), que é o L que resulta no menor tempo de CPU, ocorre em geral para $L_{\text{ótimo}} = L_{\text{máximo}} - 1$. Diminuindo-se o valor de L , em geral aumenta-se significativamente o tempo e CPU. Este resultado é praticamente o mesmo encontrado em problemas unidimensionais lineares e não-lineares, vistos nos capítulos 4 e 5, onde $L_{\text{ótimo}} \approx L_{\text{máximo}}$ para o *solver* Gauss-Seidel e o mesmo método *multigrid*.

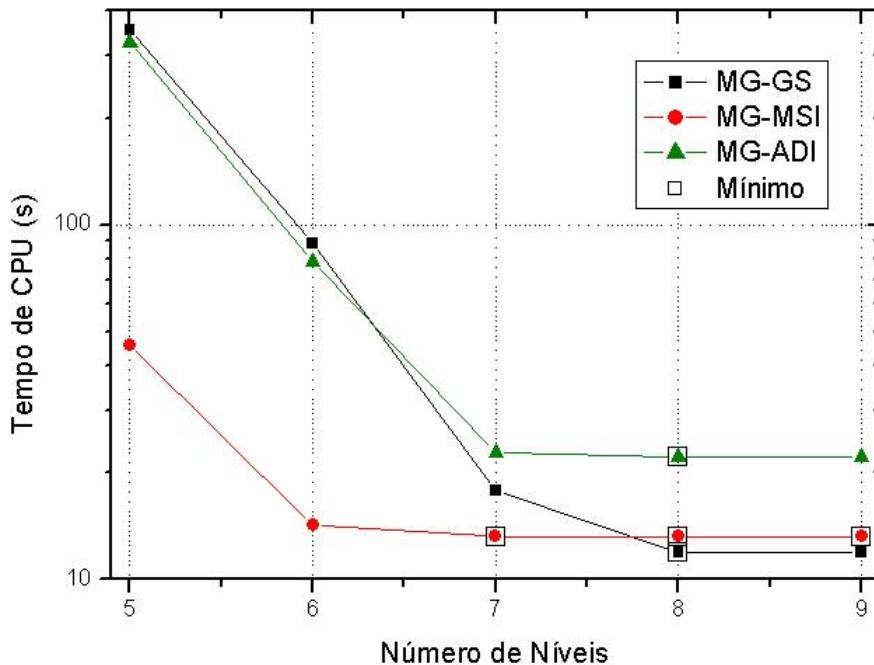


Figura 6.4: Tempo de CPU versus L para a equação de Laplace com o esquema CS, $r = 2$, com o esquema numérico CDS/CDS e diversos solvers

Este resultado é também similar aos resultados de Tannehill et al. (1997), no caso da equação de Laplace bidimensional; e Mesquita e De-Lemos (2004), no caso da equação de Navier-Stokes compressível bidimensional. Portanto recomenda-se utilizar $L_{\text{ótimo}} = L_{\text{máximo}}$ com o esquema CS, os solvers GS, MSI e ADI e razão de engrossamento $r = 2$. Devido a esta análise com $r = 2$, adotou-se $L_{\text{ótimo}} = L_{\text{máximo}}$ para outras razões de engrossamento estudadas.

Seja o problema de minimização do tempo de CPU em função do número de níveis para as diversas razões de engrossamento. Para este problema bidimensional, assim como nos casos unidimensionais dos capítulos 4 e 5, verificou-se que o número ótimo de níveis sofre uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas, ou seja, malhas com uma quantidade menor de pontos.

6.3.3 Razão de engrossamento

A figura 6.5 mostra a influência do número de incógnitas (N) sobre o tempo de CPU para os três solvers (GS, MSI e ADI) usados com o método *multigrid* (MG) com razão de engrossamento $r = 2$.

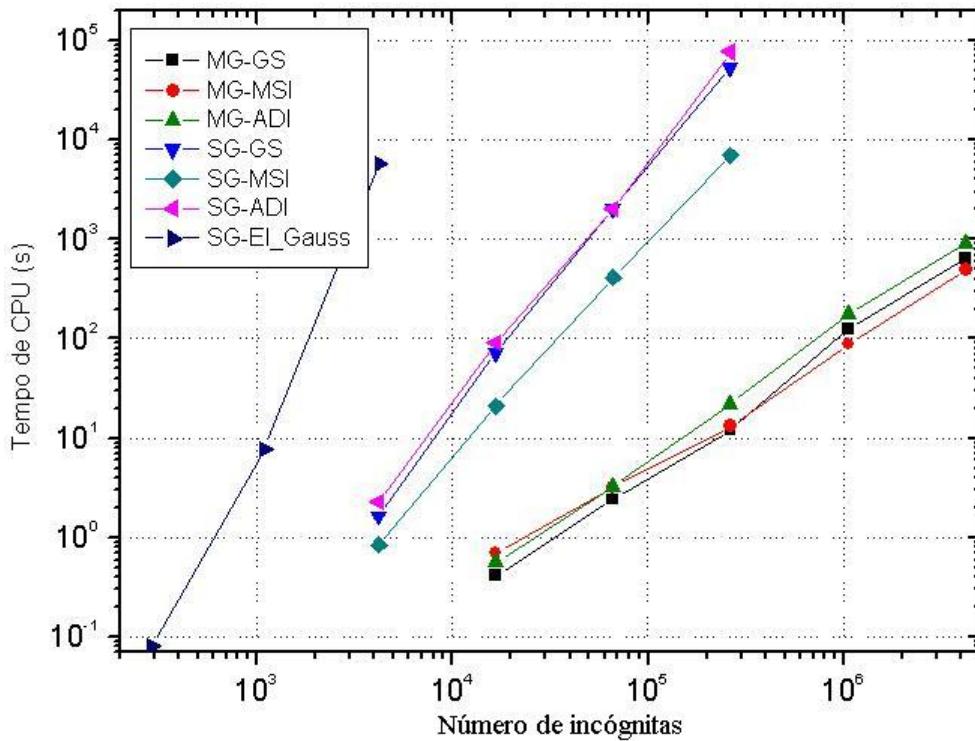


Figura 6.5: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, com o esquema numérico CDS/CDS e diversos métodos

Também são mostrados resultados para os mesmos *solvers*, mas usando-se o *singlegrid* (SG). Além disso, são apresentados resultados para um método direto, eliminação de Gauss (SG_Gauss), também baseado em malha única. Como foi dito nos capítulos anteriores, o método direto de Eliminação de Gauss é usado apenas para efeito de comparação para os métodos *singlegrid* e *multigrid*. As malhas usadas são: 5x5, 9x9, ... até 65x65 (SG_Gauss) ou 513x513 (SG) ou 2049x2049 (MG) nós.

Na figura 6.5 são mostrados apenas pontos cujo tempo de CPU não é muito influenciado pela incerteza de sua medição. Verificou-se que, para malhas com $N > 10^4$, o tempo de CPU do método *multigrid* com qualquer *solver* é significativamente menor do que o método *singlegrid* iterativo, que por sua vez, é significativamente menor do que o método *singlegrid* direto (SG_Gauss). Por exemplo, para a malha 65x65 nós, o tempo de CPU do SG-MSI (método *singlegrid* usando-se o *solver* MSI) e o SG-Gauss é de 0,84 s e 5737,22 s, respectivamente, ou seja, o SG-MSI é cerca de 6830,0 vezes mais rápido do que o SG-Gauss. Outro exemplo, para a malha 513x513 nós, o tempo de CPU do MG-MSI e SG-MSI é de 13,3 s e 7006 s, respectivamente, ou seja, o MG-MSI é cerca de 526,8 vezes mais rápido do que o SG-MSI. Estas diferenças ficam cada vez maiores à medida que N aumenta, pois as

inclinações das curvas com MG são menores do que com SG. Entre os métodos iterativos, em geral, observa-se que

$$t_{CPU}(MSI) < t_{CPU}(GS) < t_{CPU}(ADI). \quad (6.9)$$

Entre os três *solvers* testados, o MSI é o mais eficiente com qualquer método empregado, isto é, requer o menor tempo de CPU para resolver um dado sistema com N incógnitas. O MSI, que é o *solver* mais fortemente implícito (resolve o sistema em poucas iterações, tendo um comportamento quase de método direto) é o mais rápido com *multigrid* ou *singlegrid*.

Para os pontos da figura 6.5, a tabela 6.5 apresenta a inclinação (p) e os coeficientes (c) das curvas, obtida por ajuste geométrico de mínimos quadrados considerando a seguinte função

$$t_{CPU} = cN^p, \quad (6.10)$$

onde p representa a ordem do *solver* associado ao método empregado, e c é um coeficiente que depende de cada método e cada *solver*. O método *multigrid* ideal é aquele cujo $p = 1$, isto é, aquele cujo tempo de CPU cresce linearmente com o tamanho do problema, ou com o número de incógnitas N . Portanto, para cada *solver*, quanto mais próximo da unidade estiver sua ordem p , melhor é o seu desempenho.

Nota-se que o *solver* SG-Gauss foi aplicado apenas para o método *singlegrid*, pois não faz sentido usar o método *multigrid* juntamente com um *solver* direto, que é o caso do SG-Gauss.

Tabela 6.5: Coeficiente (c) e ordem (p) dos métodos e *solvers* para o esquema CS

<i>solver</i>	Tipo de <i>solver</i>	$SG - c$	$SG - p$	$MG - c$	$MG - p$
SG_Gauss	Direto	$3,15 \times 10^{-12}$	4,17	---	---
GS	Iterativo	$1,60 \times 10^{-9}$	2,50	$7,61 \times 10^{-7}$	1,35
MSI	Iterativo	$1,15 \times 10^{-8}$	2,18	$5,97 \times 10^{-6}$	1,19
ADI	Iterativo	$2,18 \times 10^{-9}$	2,50	$9,94 \times 10^{-7}$	1,36

Pela tabela 6.5 pode-se perceber que a ordem dos *solvers* é muito afetada em função do método adotado, se direto ou iterativo, e se *singlegrid* ou *multigrid*. Entre os três *solvers* testados, o MSI é o que possui o menor p para qualquer método empregado. Portanto, o *solver* mais fortemente implícito (MSI) é o que possui o menor p .

Para o *solver* MSI, a figura 6.6 mostra a influência do número de incógnitas (N) sobre o tempo de CPU para o método *multigrid* (MG) para as diversas razões de engrossamento ($r = 2, 3, 4$ e 5). Também são mostrados resultados usando-se o método *singlegrid* (SG-MSI). As malhas usadas para o *singlegrid* são: de 5×5 , 9×9 , ... até 513×513 nós. As malhas usadas para o *multigrid* são: de 5×5 , 9×9 , ... até 2049×2049 nós para $r = 2$, de 7×7 , 19×19 , ... até 1459×1459 nós para $r = 3$, de 9×9 , 33×33 , ... até 2049×2049 nós para $r = 4$ e de 11×11 , 55×55 , ... até 1251×1251 nós para $r = 5$.

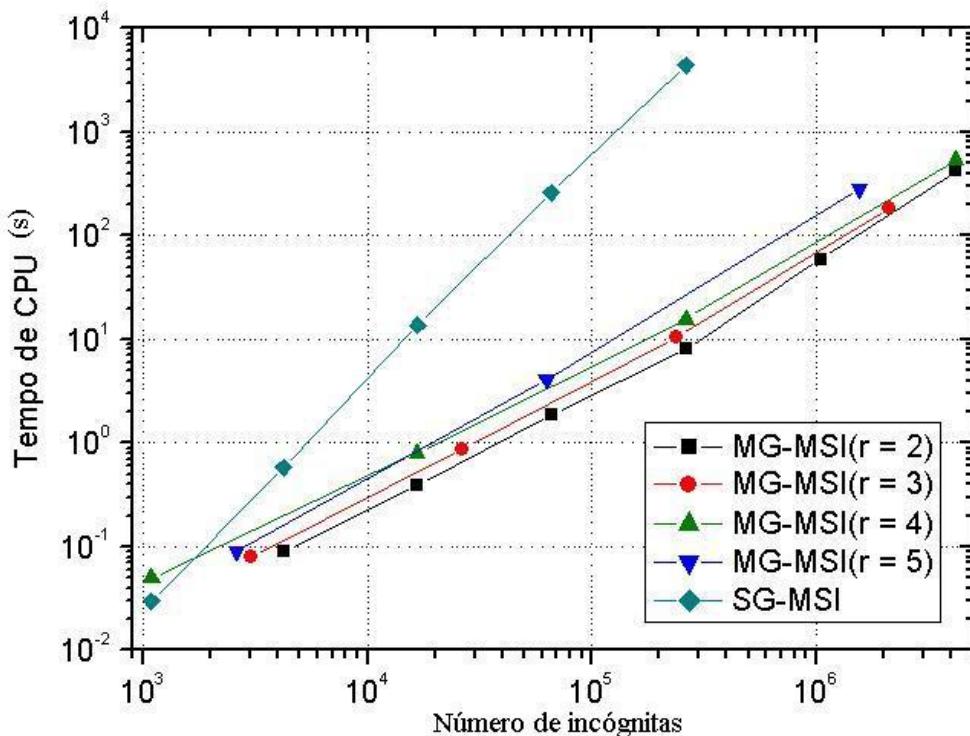


Figura 6.6: Tempo de CPU versus N para a equação de Laplace com o esquema CS, diversos r , com o esquema numérico CDS/CDS e o solver MSI

Verificou-se que o tempo de CPU do método *multigrid* com qualquer razão de engrossamento é显著mente menor do que o método *singlegrid*. Estas diferenças ficam cada vez maiores à medida que N aumenta, pois as inclinações das curvas com MG são menores do que com SG. Entre os métodos, em geral, observa-se que

$$t_{CPU}(r=2) < t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=5) < t_{CPU}(SG). \quad (6.11)$$

Este estudo confirma as análises de Brandt (1977). Em seu trabalho, Brandt fez análises teóricas e experimentais (numéricas) entre as razões de engrossamento $r = 2$, $r = 3$ e $r = 3/2$ para diversos problemas, mas não se referiu à equação 6.2 especificamente. Em seus estudos constatou que a razão $r = 2$ é a recomendável, por estar mais próxima da razão de engrossamento ótima e ser mais conveniente e econômica para o processo de interpolação, além de ser de mais fácil implementação. Briggs et al. (2000) afirmam que $r \neq 2$, em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas.

Para os pontos da figura 6.6, a tabela 6.6 apresenta a inclinação (p) das curvas, dada pela equação 6.10, como no caso anterior. A ordem p das razões de engrossamento é muito afetada em função do método adotado, se *singlegrid* ou *multigrid*, mas pouco afetada pelas razões de engrossamento. Entre as razões de engrossamento testadas, $r = 4$ é a razão que possui a menor ordem p , seguida pela razão $r = 3$.

Tabela 6.6: Coeficiente (c) e ordem (p) das razões de engrossamento para o solver MSI e esquema CS

r	$MG - c$	$MG - p$
SG	$7,59 \times 10^{-9}$	2,18
2	$2,90 \times 10^{-6}$	1,22
3	$5,85 \times 10^{-6}$	1,18
4	$1,72 \times 10^{-5}$	1,12
5	$4,26 \times 10^{-6}$	1,26

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do esquema CS. Para este problema, constatou-se que a razão ótima sofre uma forte influência de dois principais aspectos: a razão de queda do número de incógnitas de uma malha para a outra malha imediatamente mais grossa e a qualidade das informações passadas pela restrição e pela prolongação. Influências menores podem surgir do critério de parada, precisão das contas, etc.

6.4 Resultados numéricos com o esquema de aproximação completa (FAS)

Os seguintes engrossamentos foram usados nesta seção: $r = 2, 3, 4$ e 5 . Cerca de 500 simulações foram realizadas com as seguintes variantes: número de incógnitas (nós) da malha

mais fina (N), número de iterações internas (ITI), número de níveis de malhas (L), *solvers* (GS, MSI e ADI) e razão de engrossamento (r). Note que o número de simulações para gerar esta seção foi maior que da seção anterior. Isto se explica devido a alguns testes adicionais para poder detectar com mais precisão o $ITI_{óptimo}$ e o coeficiente (c) e ordem (p) dos métodos MG e SG para os diversos *solvers*. São apresentados nesta seção os resultados mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

A tabela 6.1 mostra os valores de alguns N mínimos e máximos nos diversos testes para o esquema FAS. Na tabela 6.7 a solução analítica é comparada com a solução numérica via norma infinito para as diversas razões de engrossamento com o uso do método *multigrid* e para o método *singlegrid*. Em todos os casos, o *solver* utilizado.

Tabela 6.7: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema FAS

r	N	$\ T_{analitica} - T_{numérica}\ _\infty$
2	$2049 \times 2049 = 4.198.401$	$3,49 \times 10^{-6}$
3	$1459 \times 1459 = 2.128.681$	$5,10 \times 10^{-6}$
4	$2049 \times 2049 = 4.198.401$	$8,40 \times 10^{-6}$
5	$1251 \times 1251 = 1.565.001$	$3,73 \times 10^{-7}$

Nota-se que pelas tabelas 6.2 e 6.7 que, quando se tem $\varepsilon = 10^{-7}$ (que é o caso padrão), a norma infinito do erro numérico (soma dos efeitos dos erros de iteração, arredondamento e discretização) pode ser significativamente diferente ao se usar CS e FAS. Em função de comentários da banca sobre uma possível inconsistência de tais tabelas, foram realizados testes e comentários adicionais que se encontram no apêndice C desta Tese.

6.4.1 Iteração interna

A figura 6.7 mostra a influência do número de iterações internas (ITI) sobre o tempo de CPU para alguns valores de N , *solver* MSI e razão de engrossamento $r = 2$.

Verificou-se que, para cada malha, em geral o menor tempo de CPU ocorre com $ITI_{óptimo} = 4$. Diminuindo-se ou aumentando-se o valor de ITI em relação ao ótimo, pode-se aumentar significativamente o tempo de CPU. Este comportamento da figura 6.7 também ocorre com os outros *solvers* testados (GS e ADI) e para outras razões de engrossamento.

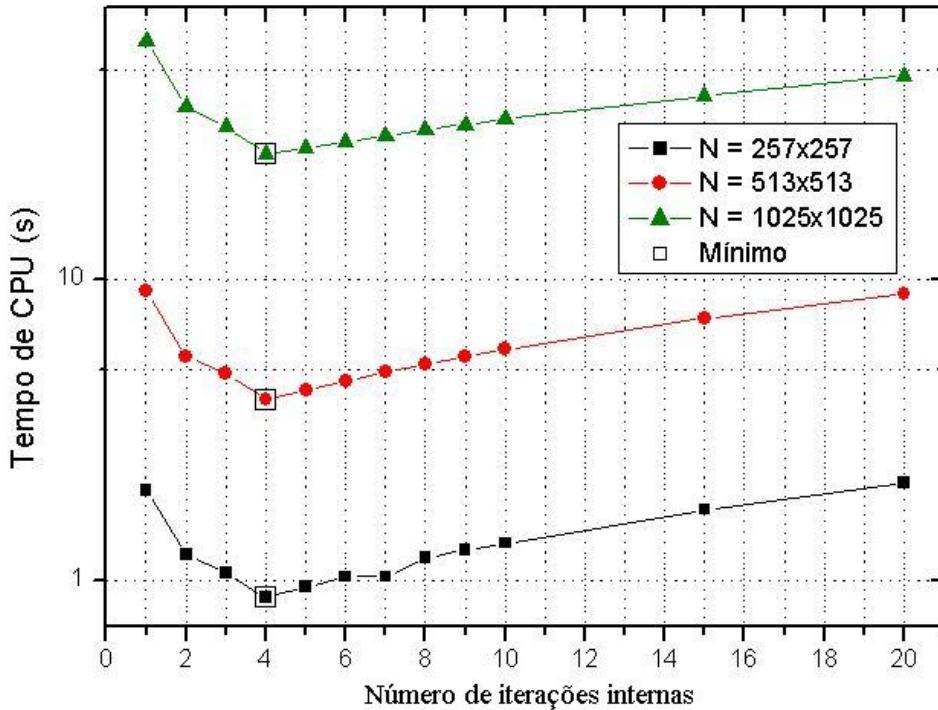


Figura 6.7: Tempo de CPU versus ITI para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e o solver MSI

Para os três *solvers* testados e para a razão de engrossamento $r = 2$, a tabela 6.8 mostra o número ótimo de iterações internas ($ITI_{óptimo}$). Observa-se que N e o *solver* influenciam pouco o valor de $ITI_{óptimo}$. Para os N da tabela 6.8 e $r = 2$, recomenda-se utilizar $ITI_{óptimo} = 3, 4$ e 5 , respectivamente, para os *solvers* GS, MSI e ADI.

Tabela 6.8: Número ótimo de iterações internas ($ITI_{óptimo}$) para diversos *solvers*, esquema FAS e $r = 2$

N	GS	MSI	ADI
257x257	4	4	5
513x513	3	4	5
1025x1025	3	4	5

Para o *solver* MSI e para as diversas razões de engrossamento $r = 2, 3, 4$ e 5 , a tabela 6.9 mostra o número ótimo de iterações internas ($ITI_{óptimo}$) e o $N_{máximo}$ utilizado por cada razão para se chegar a tal conclusão.

Note na tabela 6.9, que para $r = 2$ é recomendado $ITI_{óptimo} = 3$, o que difere da recomendação acima ($ITI_{óptimo} = 4$ para $r = 2$ e *solver* MSI), pois ao se estudar as diversas razões, levaram-se em conta problemas ainda maiores, devido ao fato do FAS ser mais rápido.

Deste estudo concluiu-se que, assintoticamente, para o *solver* MSI e $r = 2$, o $ITI_{\text{ótimo}}$ tende ao valor 3. Observa-se que a razão de engrossamento influencia pouco o valor de $ITI_{\text{ótimo}}$. Estes valores de $ITI_{\text{ótimo}}$ diferem daqueles encontrados em problemas unidimensionais lineares, dados no capítulo 4, onde $ITI_{\text{ótimo}} = 6$ para o *solver* GS, $r = 2$ e o mesmo método *multigrid*.

Tabela 6.9: Número ótimo de iterações internas ($ITI_{\text{ótimo}}$) para solver MSI, esquema FAS e diversas razões de engrossamento

r	$N_{\text{máximo}}$	$ITI_{\text{ótimo}}$
2	2049x2049	3
3	1459x1459	3
4	2049x2049	4
5	1251x1251	4

Na seção anterior, verificou-se que para o esquema CS e usando-se o *solver* MSI, $ITI_{\text{ótimo}} = 1$ para $r = 2$ e $ITI_{\text{ótimo}} = 2$ para as demais razões de engrossamento. Já nesta seção, verificou-se que para o esquema FAS, $ITI_{\text{ótimo}} = 3$ para as razões $r = 2$ e 3, e $ITI_{\text{ótimo}} = 4$ para as razões $r = 4$ e 5. Portanto, o esquema utilizado (CS ou FAS) influencia o número ótimo de iterações internas do método *multigrid*.

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema FAS. Para este problema, verificou-se que o número ótimo de iterações internas sofre pouca influência do número de vezes que o *solver* necessita para atualizar o resíduo e a solução em todo o domínio bidimensional.

6.4.2 Nível de malha

A figura 6.8 mostra a influência do número de níveis (L) sobre o tempo de CPU, para $N = 513 \times 513$ nós, os três *solvers* testados (GS, MSI e ADI) e a razão de engrossamento $r = 2$. Verificou-se que, para cada solver, o número ótimo de níveis de malha ($L_{\text{ótimo}}$), ocorre na média, para $L_{\text{ótimo}} = L_{\text{máximo}} - 1$, ou seja, $L_{\text{ótimo}} \approx L_{\text{máximo}}$. Diminuindo-se o valor de L , em geral aumenta-se significativamente o tempo e CPU. Portanto recomenda-se utilizar $L_{\text{ótimo}} = L_{\text{máximo}}$ com o esquema FAS, os *solvers* GS, MSI e ADI e razão de engrossamento $r = 2$.

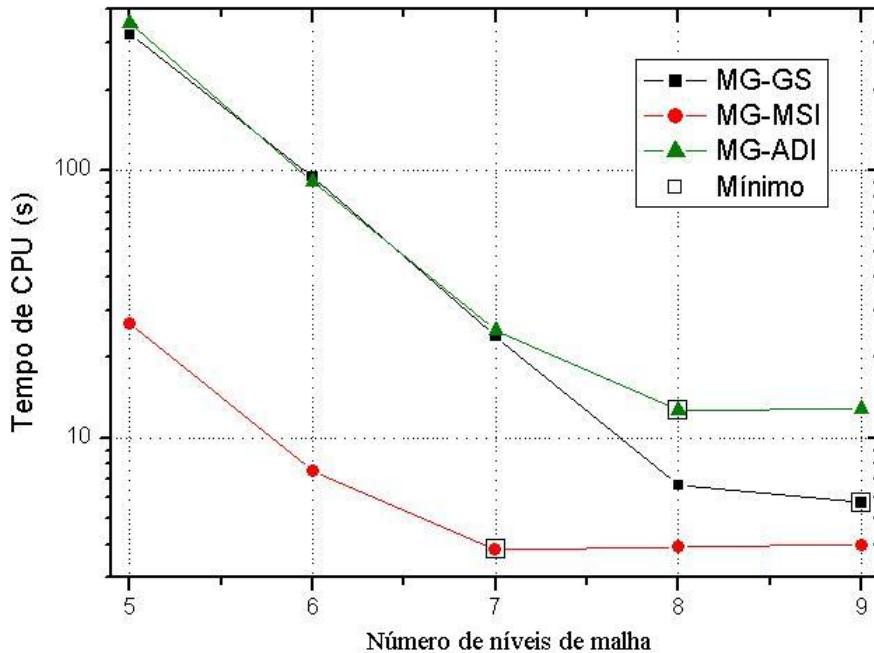


Figura 6.8: Tempo de CPU versus L para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e diversos solvers

Na seção anterior, verificou-se que para o esquema CS, $L_{\text{ótimo}} = L_{\text{máximo}} - 1$, com qualquer solver. Já nesta seção, ocorre o mesmo para o esquema FAS. Portanto, o esquema utilizado (CS ou FAS) não tem grande influência sobre o número ótimo de níveis para o método *multigrid*. Devido a esta análise com $r = 2$, adotou-se $L_{\text{ótimo}} = L_{\text{máximo}}$ para outras razões de engrossamento estudadas.

Seja o problema de minimização do tempo de CPU em função do número de níveis para as diversas razões de engrossamento. Para este problema bidimensional, assim como nos casos unidimensionais dos capítulos 4 e 5 e no esquema CS da seção anterior, verificou-se que o número ótimo de níveis sofre uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas.

6.4.3 Razão de engrossamento

A figura 6.9 mostra a influência do número de incógnitas (N) sobre o tempo de CPU para os três solvers (GS, MSI e ADI) usados com o método *multigrid* (MG) com razão de engrossamento $r = 2$.

Também são mostrados resultados para os mesmos *solvers*, mas usando-se o método *singlegrid* (SG) e eliminação de Gauss (SG_Gauss). As malhas usadas são: 5x5, 9x9, ... até 65x65 (SG_Gauss) ou 513x513 (SG) ou 2049x2049 (MG) nós.

Na figura 6.9 são mostrados apenas pontos cujo tempo de CPU não é muito influenciado pela incerteza de sua medição. Verificou-se que, para malhas com $N > 10^4$, o tempo de CPU do método *multigrid* com qualquer *solver* é significativamente menor do que o método *singlegrid* iterativo, que por sua vez, é significativamente menor do que o método *singlegrid* direto (SG_Gauss). Por exemplo, para a malha 513x513 nós, o tempo de CPU do MG-MSI e SG-MSI é de 3,8 s e 7006 s, respectivamente, ou seja, o MG-MSI é cerca de 1843,7 vezes mais rápido do que o SG-MSI. Estas diferenças ficam cada vez maiores à medida que N aumenta, pois as inclinações das curvas com MG são menores do que com SG. Entre os métodos iterativos, em geral, também vale a equação 6.9. Entre os três *solvers* testados, o *solver* mais fortemente implícito (MSI) é o mais rápido.

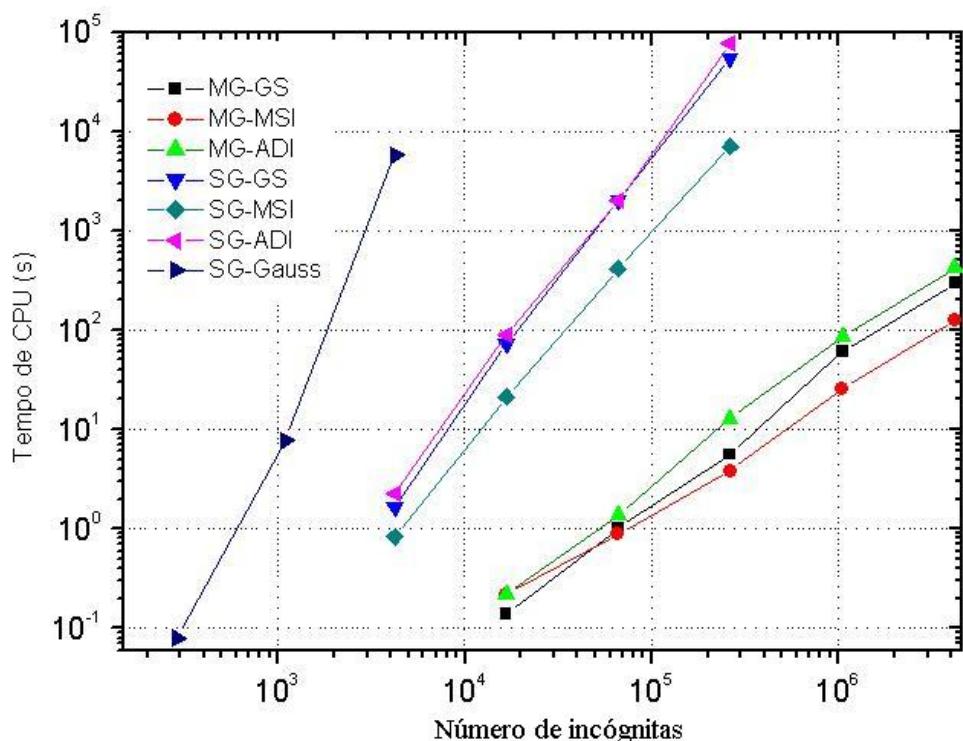


Figura 6.9: Tempo de CPU versus N para a equação de Laplace com o esquema FAS, $r = 2$, com o esquema numérico CDS/CDS e com diversos solvers

Para os pontos da figura 6.9, a tabela 6.10 apresenta o coeficiente (c) e a inclinação (p) das curvas, obtida por ajuste geométrico de mínimos quadrados considerando a função dada na equação 6.10. Pelas tabelas 6.5 e 6.10 pode-se perceber que a ordem dos *solvers* com

multigrid é pouco afetada em função do esquema adotado, se CS ou FAS, mas bastante afetada em função do método adotado, se direto ou iterativo, se *singlegrid* ou *multigrid*. Entre os três solvers testados, o MSI é o que possui o menor p com qualquer método empregado, ou seja, o *solver* mais fortemente implícito (MSI) é o que possui o menor p .

Tabela 6.10: Coeficiente (c) e ordem (p) dos métodos e *solvers* para o esquema FAS

Solver	tipo de solver	$SG - c$	$SG - p$	$MG - c$	$MG - p$
SG_Gauss	Direto	$3,15 \times 10^{-12}$	4,17	---	---
GS	Iterativo	$1,60 \times 10^{-9}$	2,50	$1,68 \times 10^{-7}$	1,40
MSI	Iterativo	$1,15 \times 10^{-8}$	2,18	$2,90 \times 10^{-6}$	1,12
ADI	Iterativo	$2,18 \times 10^{-9}$	2,50	$2,99 \times 10^{-7}$	1,39

Para o *solver* MSI a figura 6.10 mostra a influência do número de incógnitas (N) sobre o tempo de CPU para o método *multigrid* (MG) para as diversas razões de engrossamento ($r = 2, 3, 4$ e 5). Também são mostrados resultados usando-se o método *singlegrid* (SG-MSI). A figura 6.11 mostra um detalhe da figura 6.10. As malhas usadas para o *multigrid* são: de 5×5 , 9×9 , ... até 2049×2049 nós para $r = 2$, de 7×7 , 19×19 , ... até 1459×1459 nós para $r = 3$, de 9×9 , 33×33 , ... até 2049×2049 nós para $r = 4$ e de 11×11 , 55×55 , ... até 1251×1251 nós para $r = 5$.

Verificou-se que o tempo de CPU do método *multigrid* com qualquer razão de engrossamento é significativamente menor do que o método *singlegrid*. Estas diferenças ficam cada vez maiores à medida que N aumenta, pois as inclinações das curvas com MG são menores do que com SG. Entre os métodos, para problemas de grande porte, observa-se que

$$t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=2) < t_{CPU}(r=5) < t_{CPU}(SG). \quad (6.12)$$

Esta constatação é inédita e inesperada, além de mostrar que a razão $r = 2$ perde a hegemonia perante outras razões.

Para os pontos da figura 6.10, a tabela 6.11 apresenta a inclinação (p) das curvas, dada pela equação 6.10, como no caso anterior. A ordem p das razões de engrossamento é muito afetada em função do método adotado, se *singlegrid* ou *multigrid*, mas pouco afetada pelas razões de engrossamento. Entre as razões de engrossamento testadas, $r = 2$ é a razão que possui a menor ordem p , seguida pela razão $r = 4$.

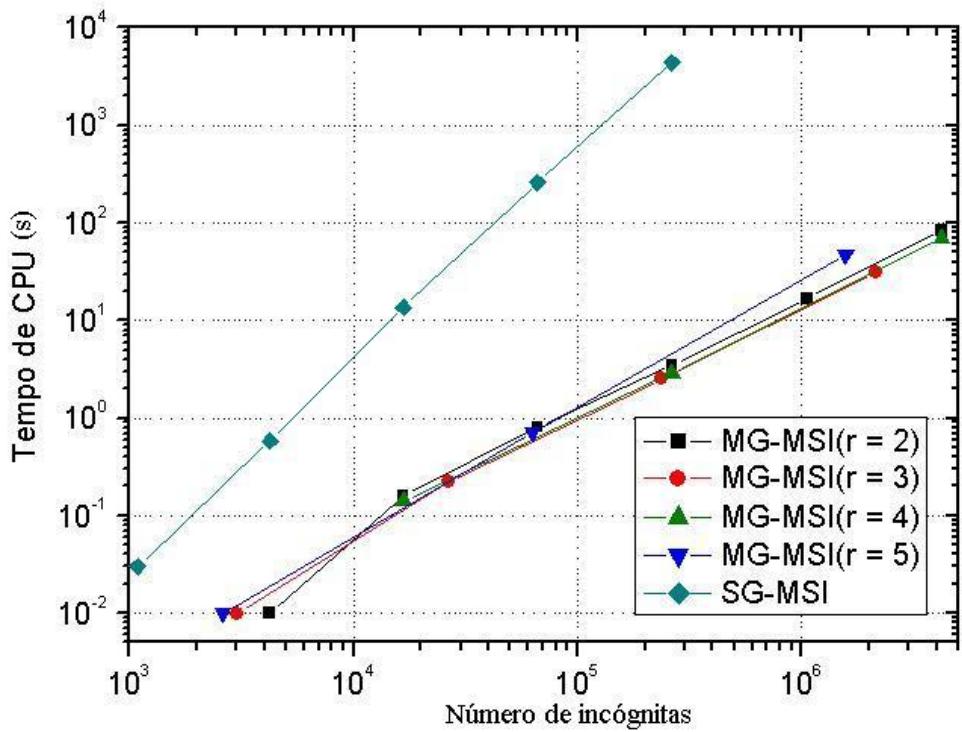


Figura 6.10: Tempo de CPU versus N para a equação de Laplace com o esquema FAS, diversos r , com o esquema numérico CDS/CDS e solver MSI

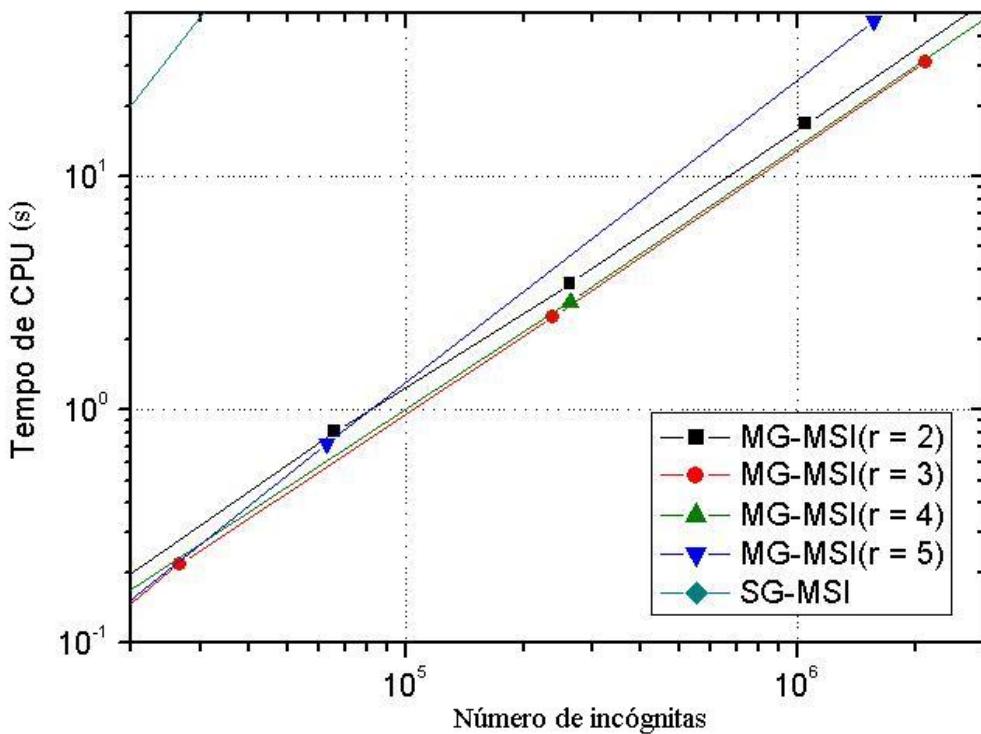


Figura 6.11: Detalhe da figura 6.10

Tabela 6.11: Coeficiente (c) e ordem (p) das razões de engrossamento para o solver MSI e esquema FAS

r	$MG - c$	$MG - p$
SG	$1,15 \times 10^{-8}$	2,18
2	$2,90 \times 10^{-6}$	1,12
3	$7,16 \times 10^{-7}$	1,21
4	$2,43 \times 10^{-6}$	1,13
5	$3,19 \times 10^{-7}$	1,32

Na seção anterior, verificou-se que para o esquema CS e usando-se o *solver* MSI, $r = 2$ é a razão de engrossamento ótima. Já nesta seção, verificou-se que para o esquema FAS, $r = 3$ é a razão de engrossamento ótima. Portanto, o esquema utilizado (CS ou FAS) influencia a razão de engrossamento ótima do método *multigrid*.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do esquema FAS. Para este problema, constatou-se que a razão ótima sofre uma forte influência dos seguintes aspectos: a razão de queda do número de incógnitas de uma malha para a outra malha imediatamente mais grossa, a qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, o uso do algoritmo FAS, que transfere informações a respeito do resíduo e da solução para as demais malhas mais grossas.

6.5 Esquema de correção (CS) versus esquema de aproximação completa (FAS)

Nas figuras 6.12 e 6.13, respectivamente para as razões $r = 2$ e $r = 3$, se faz uma comparação entre os esquemas *multigrid* CS e FAS em função de diversos valores de N com o *solver* MSI.

Pode-se notar que o esquema FAS é mais rápido do que o esquema CS em qualquer N para as duas razões analisadas.

Por exemplo, para a malha de 2049x2049 nós o tempo de CPU do MG-FAS($r = 2$) e MG-CS($r = 2$) é de 83,55 s e 416,55 s, respectivamente, ou seja, o MG-FAS($r = 2$) é cerca de 4,8 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 1459x1459 nós o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 3$) é de 30,94 s e 184,61 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 6,0 vezes mais rápido do que o MG-CS($r = 3$).

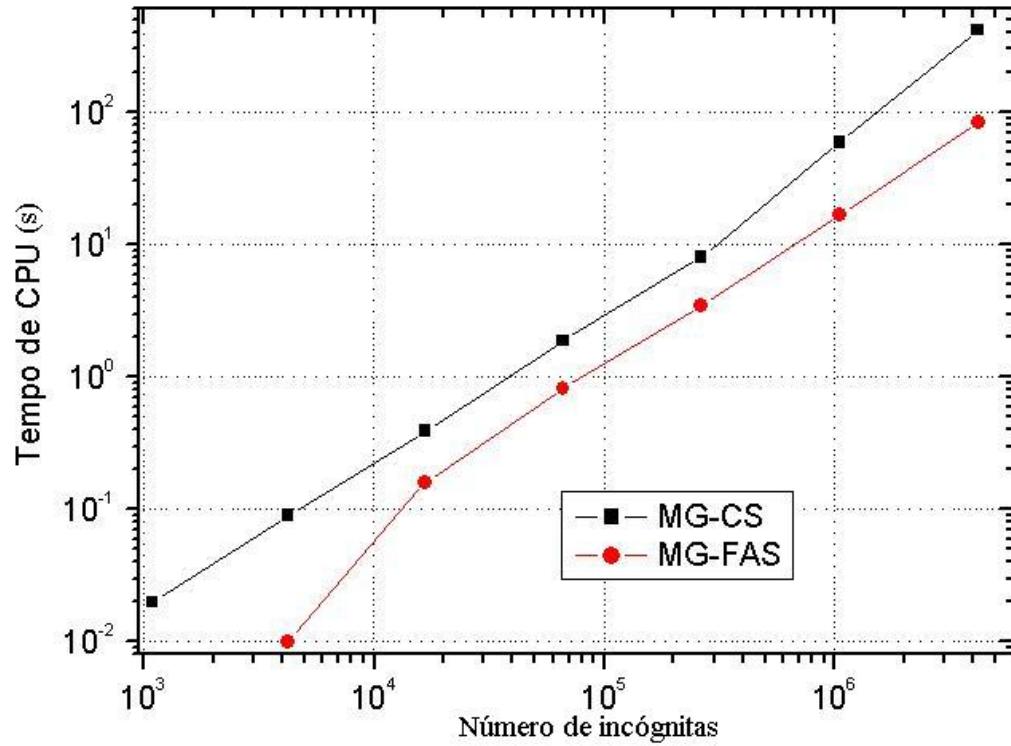


Figura 6.12: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 2$, com o esquema numérico CDS/CDS e solver MSI

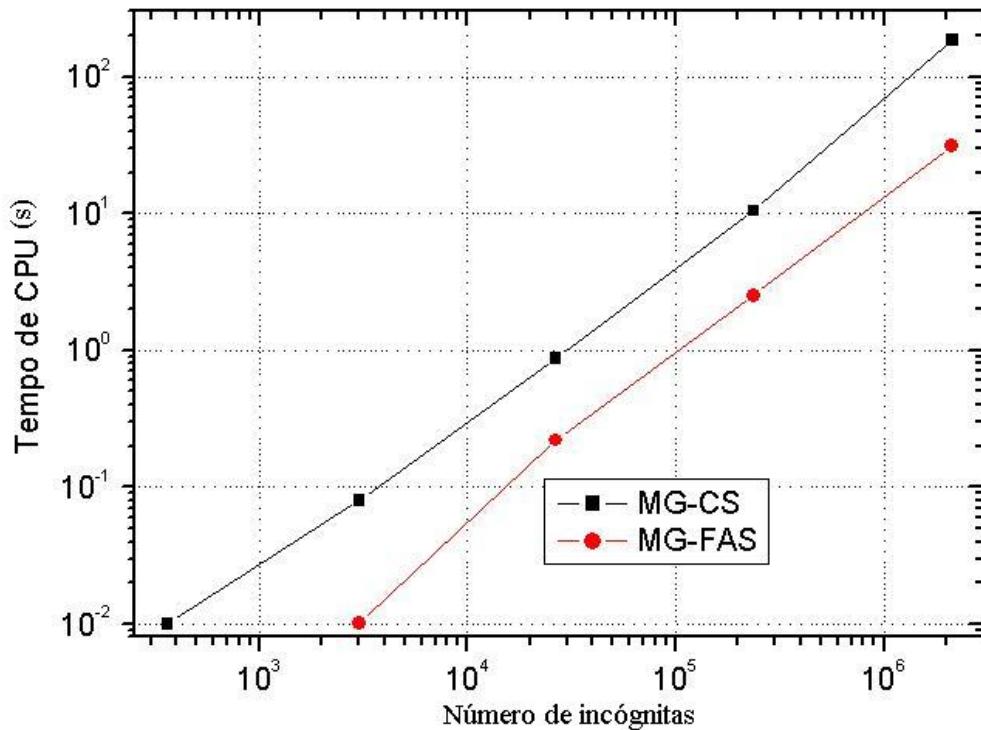


Figura 6.13: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 3$, com o esquema numérico CDS/CDS e solver MSI

Na figura 6.14 se faz uma comparação entre os esquemas *multigrid* CS (com a melhor razão de engrossamento, $r = 2$) e FAS (com a melhor razão de engrossamento, $r = 3$) em função de diversos valores de N com o *solver* MSI.

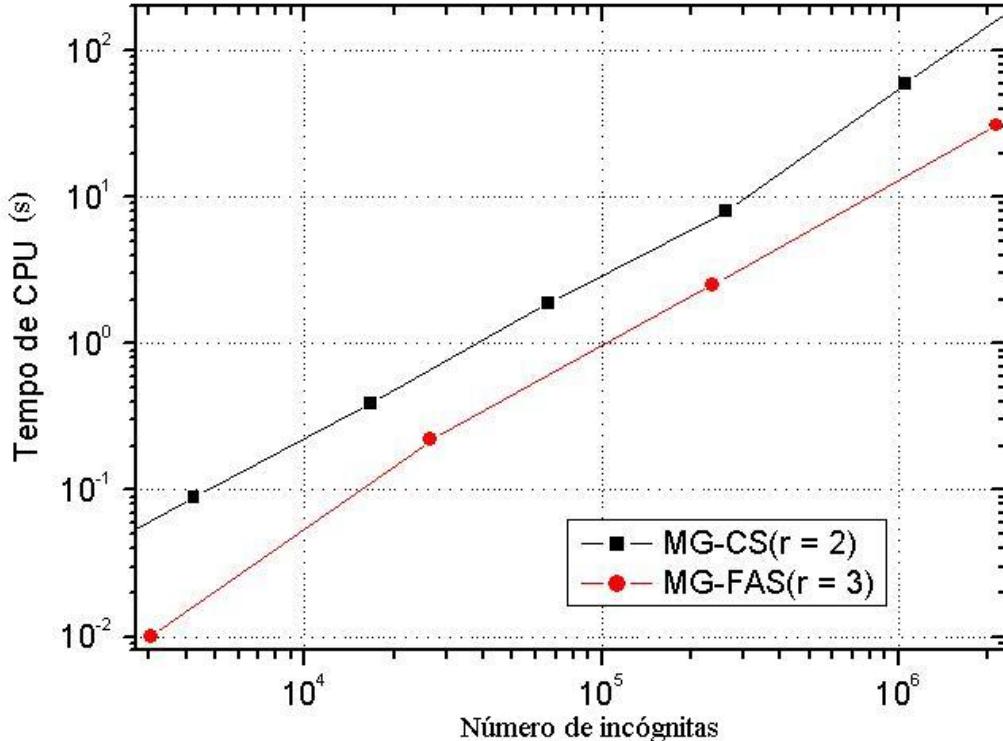


Figura 6.14: Tempo de CPU versus N para a equação de Laplace com os esquemas CS e FAS, $r = 2$ e $r = 3$, com o esquema numérico CDS/CDS e solver MSI

Usando-se ajuste geométrico de curvas, o tempo de CPU do MG-FAS($r = 3$) pode ser aproximado por $t_{CPU} = 7,16 \times 10^{-7} N^{1,21}$ e MG-CS($r = 2$) pode ser aproximado por $t_{CPU} = 2,90 \times 10^{-6} N^{1,22}$.

Pode-se notar que o esquema MG-FAS($r = 3$) é mais rápido do que o esquema MG-CS($r = 2$) em qualquer N . Por exemplo, para a malha de 10^5 nós o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 0,85 s e 3,45 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,1 vezes mais rápido do que o MG-CS($r = 2$). Outro exemplo, para a malha de 10^6 nós o tempo de CPU do MG-FAS($r = 3$) e MG-CS($r = 2$) é de 13,87 s e 56,65 s, respectivamente, ou seja, o MG-FAS($r = 3$) é cerca de 4,1 vezes mais rápido do que o MG-CS($r = 2$).

Portanto, recomenda-se o uso do esquema FAS com o *solver* MSI e razão de engrossamento $r = 3$ para a equação de Laplace bidimensional. Esta constatação é inédita e inesperada, pois o esquema CS é indicado para resolver equações lineares, caso deste

capítulo, e o esquema FAS, para equações não-lineares. Aqui se vê que o esquema CS perde a hegemonia perante o esquema FAS, mesmo para um problema linear.

Os resultados do presente trabalho diferem daqueles de Brandt (1977). Ele faz análises teóricas e experimentais (numéricas) entre as razões $r = 2, 3$ e $3/2$ para diversos problemas, mas não se referiu à equação 6.2 exatamente. Brandt mostra preferência pelo esquema CS em relação ao esquema FAS no caso de problemas lineares. Segundo Brandt, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS, devido aos cálculos exigidos no esquema FAS.

Seja o problema de minimização do tempo de CPU em função do uso dos diferentes esquemas CS e FAS. Para este problema, constatou-se que o esquema ótimo sofre uma forte influência da qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, transferência de informações a respeito do resíduo e da solução para as malhas mais grossas. Para o esquema FAS, por exemplo, a estimativa inicial para as malhas mais grossas é sempre a restrição da solução suavizada da malha imediatamente mais fina, enquanto que para o esquema CS, a estimativa inicial é sempre a estimativa inicial nula. Ou seja, o esquema FAS tem sempre estimativas iniciais melhores para os problemas nas malhas mais grossas. Com isto, constatou-se no presente capítulo que o número de ciclos ou iterações externas (ITE) do esquema FAS é significativamente menor do que o esquema CS, ou seja, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS, mas o número de ciclos iterativos deste esquema é significativamente menor que do esquema CS. Por exemplo, para a malha 2049×2049 nós, o ITE do MG-FAS($r = 2$) e o MG-CS($r = 2$) é de 2 e 15, respectivamente. Outro exemplo, para a malha de 1459×1459 nós o ITE do MG-FAS($r = 3$) e o MG-CS($r = 3$) é de 2 e 25, respectivamente.

6.6 Conclusão do capítulo 6

Neste capítulo foi resolvido numericamente um problema: equação de Laplace bidimensional. Esta equação é linear e ela representa um problema de transferência de calor. O esquema numérico de aproximação usado foi: CDS/CDS. Foram usados dois esquemas do método *multigrid*: esquemas CS (*Correction Scheme*) e FAS (*Full Approximation Scheme*).

Com base nos resultados deste capítulo, verificou-se que:

1. Para os mesmos dados (ε, v e N), o esquema FAS é mais rápido do que o esquema CS;

2. O *solver* MSI é mais rápido do que o Gauss-Seidel e o ADI, para os esquemas CS e FAS. Portanto, mesmo com o método *multigrid*, quanto mais fortemente implícito é o *solver*, mais rápido ele é;
3. O número de iterações internas (ITI) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) e a dimensão do problema (1D ou 2D) influenciam o $ITI_{óptimo}$. Para o uso do esquema CS e $r = 2$, $ITI_{óptimo} = 1$ com qualquer *solver*, e o uso do esquema FAS e $r = 2$, $ITI_{óptimo} = 3$ a 5, dependendo do *solver*. Para o *solver* MSI, o $ITI_{óptimo}$ é quase constante com a variação das razões de engrossamentos testadas;
4. O número de níveis de malhas (L) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) não tem muita influência no $L_{óptimo}$. Para ambos os esquemas e para os *solvers* testados, $L_{óptimo} \approx L_{máximo}$;
5. Para os mesmos dados (ε , v e N), com $L = L_{máximo}$ para cada N , entre as razões de engrossamento de malhas testadas ($r = 2, 3, 4$ e 5), o esquema CS é mais rápido com $r = 2$; já o esquema FAS é mais rápido com $r = 3$;
6. O esquema FAS (com razão de engrossamento $r = 3$) é aproximadamente 4 vezes mais rápido do que o esquema CS (com razão de engrossamento $r = 2$).

Recomendações práticas:

Com base nos resultados deste capítulo são feitas as seguintes recomendações aos usuários do método *multigrid* para problemas iguais ou similares ao estudado.

Problemas bidimensionais lineares isotrópicos:

- Esquema: FAS;
- *Solver*: MSI;
- Razão de engrossamento: $r = 3$;
- Número de iterações internas: $ITI = 3$;
- Número de níveis de malhas: $L = L_{máximo}$.

7 Problema bidimensional linear anisotrópico: equação de Laplace

Neste capítulo descreve-se a quarta classe de problemas tratados neste trabalho, que é o problema bidimensional linear em malhas anisotrópicas⁷. Neste caso, como no capítulo anterior, estuda-se a equação de Laplace (TANNEHILL et al., 1997).

No capítulo anterior estudaram-se problemas isotrópicos onde $RA=1$, ou seja, $h_x = h_y$. Aqui RA é a razão de aspecto definida pela equação 6.4. Este capítulo dedica-se a problemas anisotrópicos (BRIGGS et al., 2000; DENDY et al., 1989) onde $0 < RA \ll 1$ ou $RA \gg 1$, ou seja, anisotropia devido aos elementos da malha ser altamente alongados (anisotropia geométrica).

Na figura 7.1 tem-se um exemplo de malha isotrópica e um outro exemplo de malha anisotrópica com $RA = 4$.

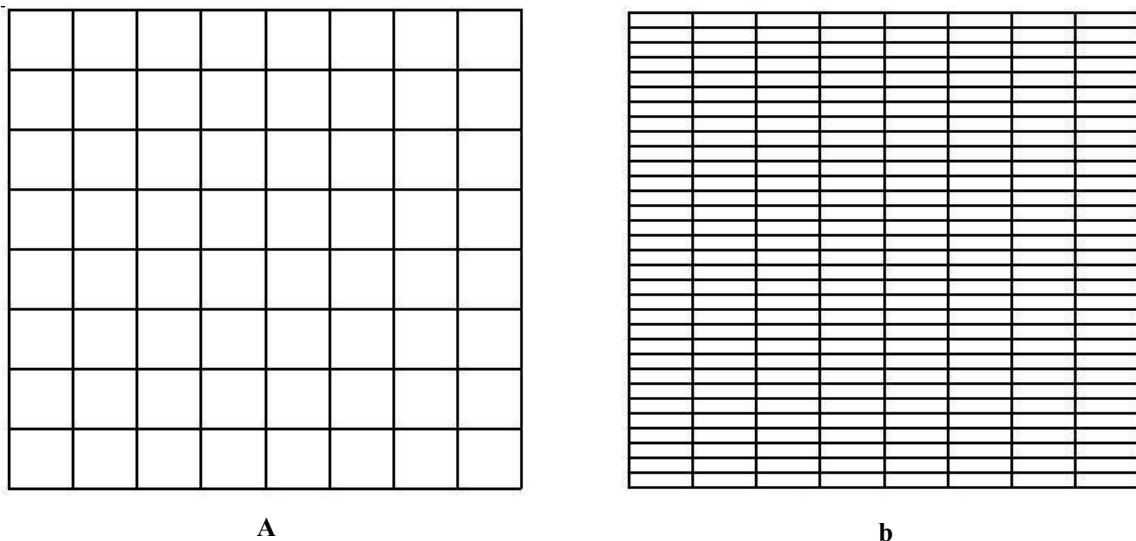


Figura 7.1: Exemplos de malhas: a) isotrópica e b) anisotrópica com $RA = 4$

Sabe-se que quanto mais distinta a razão de aspecto (RA) está da unidade, mais o desempenho do método *multigrid* se deteriora (Wesseling e Oosterlee, 2001), com isso a taxa de convergência piora, podendo até mesmo ocorrer divergência (Larsson et al., 2005). Este tipo de anisotropia é muito comum em problemas práticos de engenharia, como em problemas

⁷ Anisotropia de malha, ou seja, anisotropia geométrica, como definida na seção 3.4 do capítulo 3.

de camada limite, onde a razão de aspecto (RA) pode ser da ordem de 10^3 , 10^4 ou mais (Wesseling e Oosterlee, 2001).

Este capítulo está dividido da seguinte forma: primeiramente estudam-se os modelos matemático e numérico e dados de implementação. Em seguida detalham-se os principais resultados numéricos com os algoritmos *multigrid* usados para problemas anisotrópicos: EP, EP-SE, SE e SE-EP, que serão definidos nas seção 7.2.

7.1 Modelos matemático e numérico

O problema linear de condução de calor bidimensional (equação de Laplace), com condições de contorno de Dirichlet, em coordenadas cartesianas, considerado neste capítulo é o modelo dado pela equação 6.2.

Como nos capítulos anteriores, o método numérico a ser utilizado neste problema será o método de diferenças finitas (BURDEN e FAIRES, 1997; TANNEHILL et al., 1997). A discretização do domínio será desenvolvida, fazendo-se uso de malhas estruturadas e uniformes por direção⁸, onde o domínio $\{(x, y) \in R^2 : 0 \leq x, y \leq 1\}$ é particionado em subconjuntos através de um número de incógnitas (ou número de pontos), dado por $N = N_x N_y$, onde N_x e N_y são os números de pontos nas direções coordenadas x e y , respectivamente. Isto introduz uma malha, conforme equação 6.3. Neste caso, estabelece-se uma malha com elementos de tamanho h_x por h_y que se denota por Ω^h , com razão de aspecto (RA) definida pela equação 6.4.

O esquema numérico para este problema será o mesmo utilizado no capítulo anterior, ou seja, diferenças centradas (CDS) para a derivada de segunda ordem (termo difusivo), resultando, portanto, na equação 6.6 com os coeficientes dados pelas equações 6.7 e termo fonte pela equação 6.8.

Novamente a equação 4.15 é resolvida com o método *multigrid* fazendo-se uso do esquema CS e quatro algoritmos que serão detalhados na próxima seção. Neste caso, os sistemas de equações do tipo da equação 4.15, onde f representa o termo fonte (resíduo) a cada nível de malha, são resolvidos com o solver Gauss-Seidel (GS).

⁸ h_x e h_y são constantes, mas não necessariamente iguais.

7.2 Dados de implementação

Neste capítulo são descritos quatro algoritmos baseados no método *multigrid* para problemas com malhas anisotrópicas ($RA \neq 1$).

São eles: engrossamento padrão (EP), engrossamento padrão seguido de semi-engrossamento (EP-SE), semi-engrossamento (SE) e semi-engrossamento seguido de engrossamento padrão (SE-EP). A razão de engrossamento $r = 2$ é usada em todos os algoritmos apresentados a seguir:

- 1) Engrossamento padrão (EP): neste algoritmo, apresentado em Brandt (1977), é realizado simultaneamente o engrossamento em ambas as direções coordenadas x e y . Desta forma, mantém-se a mesma razão de aspecto em todas as malhas. Este algoritmo é mostrado nos passos 1a a 1c da figura 7.2.
- 2) Semi-engrossamento (SE): neste algoritmo, apresentado em Mulder (1989), é realizado o engrossamento em apenas uma das direções coordenadas, onde isto seja possível ou desejado. Com isso, diminui-se a razão de aspecto a cada nível percorrido até se atingir a malha mais grossa possível ou desejada, que idealmente é isotrópica, ou seja, $RA = 1$. Este algoritmo é mostrado nos passos 3a a 3c da figura 7.2.
- 3) Engrossamento padrão seguido de semi-engrossamento (EP-SE): neste algoritmo, proposto neste trabalho, primeiro aplica-se o algoritmo EP (passos 2a a 2c da figura 7.2) e, em seguida, o SE (passos 2c a 2e da figura 7.2).
- 4) Semi-engrossamento seguido de engrossamento padrão (SE-EP): neste algoritmo, apresentado por Zhang (2002), primeiro aplica-se o algoritmo SE (passos 4a a 4c da figura 7.2) e, em seguida, o EP (passos 4c a 4e da figura 7.2).

O método apresentado em Larsson et al. (2005) não foi objeto de comparação nesta seção, pois ele trata somente de semi-engrossamento condicional e a proposta aqui é comparar algoritmos que mesclam a técnica de engrossamento padrão com semi-engrossamento.

No presente capítulo, adota-se o número máximo possível de níveis de engrossamento padrão para os algoritmos EP e EP-SE. Como se tratam de problemas anisotrópicos, para os algoritmos EP-SE, SE e SE-EP, a direção de semi-engrossamento sempre existe (Zhang, 2002).

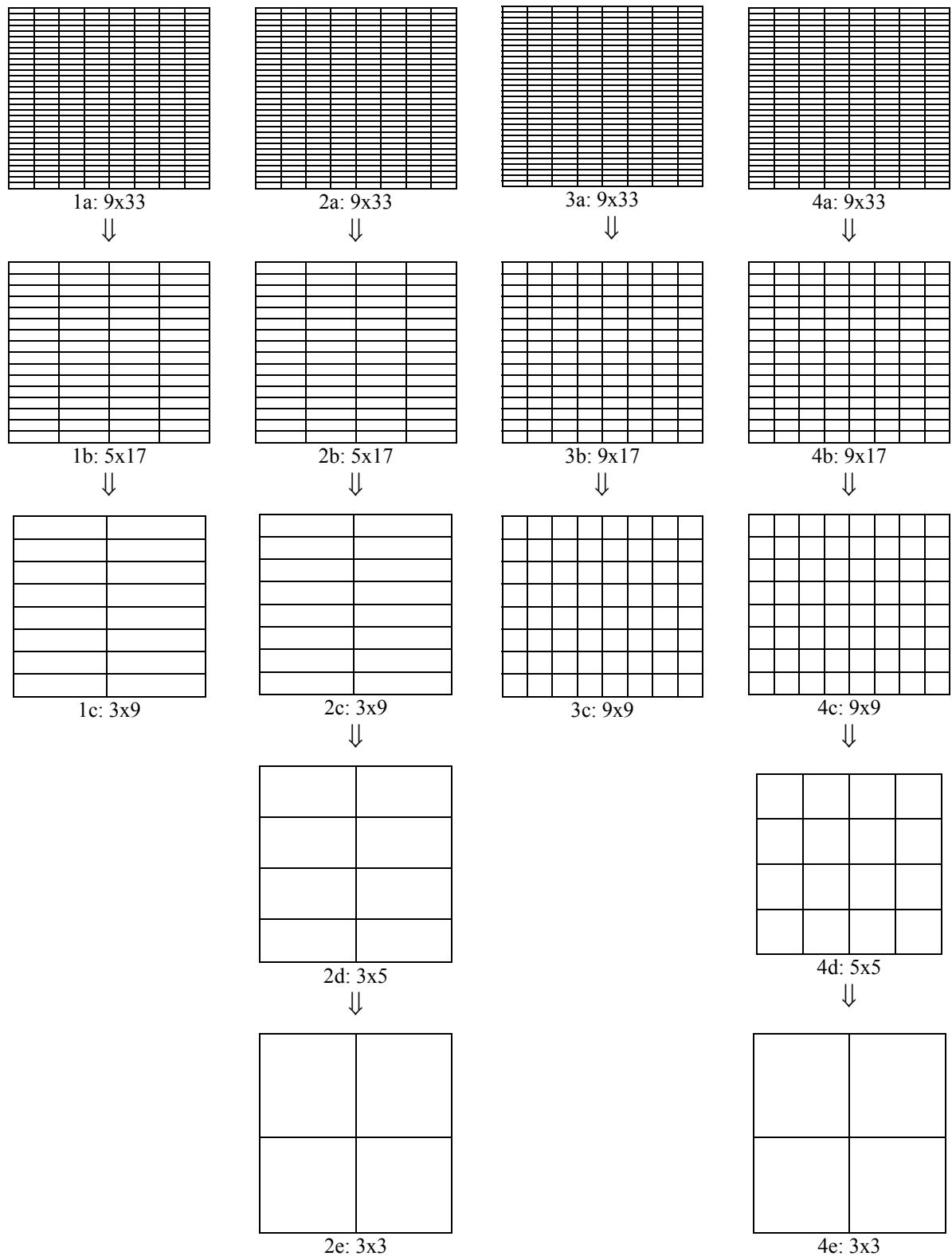


Figura 7.2: Engrossamentos: EP (1a-1c), EP-SE (2a-2e), SE (3a-3c) e SE-EP (4a-4e)

Esta direção também existe em problemas isotrópicos, entretanto não é interessante utilizar um algoritmo de semi-engrossamento em um domínio e uma malha onde o algoritmo EP é suficientemente bom.

O algoritmo proposto neste capítulo, EP-SE, confronta o algoritmo proposto por Zhang (2002), SE-EP, que se utiliza as técnicas de engrossamento padrão e semi-engrossamento na ordem inversa. Acredita-se que este algoritmo (EP-SE) possa ter bons resultados devido ao número reduzido de operações aritméticas se comparado com o algoritmo SE-EP.

7.3 Resultados numéricos

A razão de engrossamento padrão $r = 2$, o esquema CS e o *solver* Gauss-Seidel foram utilizados para todas as simulações descritas nesta seção. Cerca de 240 simulações foram realizadas com as seguintes variantes: algoritmos EP, EP-SE, SE e SE-EP; número de incógnitas (nós) da malha mais fina (N), número de iterações internas (ITI) e razões de aspecto (RA) $1/1024$, 1, 2, 16, 128, 1024 e 8192. São apresentados neste capítulo os resultados mais representativos, suficientes para mostrar qualitativamente o desempenho do método *multigrid*.

Na tabela 7.1 a solução analítica é comparada com a solução numérica, segundo o algoritmo SE-EP, via norma infinito para as diversas razões de aspecto para o método *multigrid*.

Tabela 7.1: Norma infinito do erro para as malhas mais finas para cada uma das razões de aspectos (RA) testadas com o *solver* GS e o esquema CS

RA	N	$\ T_{\text{analítica}} - T_{\text{numérica}}\ _\infty$
1/1024	16385x17	$5,54 \times 10^{-4}$
1	2049x2049	$6,80 \times 10^{-8}$
2	2049x4097	$4,25 \times 10^{-8}$
16	513x8193	$5,46 \times 10^{-7}$
128	129x16385	$8,71 \times 10^{-6}$
1024	33x32769	$1,39 \times 10^{-4}$
8192	17x131073	$5,57 \times 10^{-4}$

Nota-se desta tabela que os valores de N utilizados para as malhas com razão de aspecto $RA = 1/1024$ e $RA = 1024$ são diferentes, pois o processo de resolução para a malha com $RA = 1/1024$ é muito mais lento do que para $RA = 1024$, logo optou-se por uma malha com N menor para esta malha (com $RA = 1/1024$).

7.3.1 Iteração interna (ITI) e razão de aspecto (RA)

A tabela 7.2 mostra o número de incógnitas ($N = N_x N_y$) das malhas finas e as respectivas razões de aspecto (RA) utilizadas para determinar o número ótimo de iterações internas ($ITI_{\text{ótimo}}$) para cada conjunto de dados constituídos por: algoritmo, RA e número de incógnitas (N). Nota-se que, dada uma certa razão de aspecto RA , o N varia de um algoritmo para outro. Isto se explica porque o tempo de CPU varia muito de um algoritmo para outro. Para o algoritmo que apresentou o menor tempo de CPU, optou-se por investigar malhas mais refinadas.

Tabela 7.2: Malhas utilizadas para determinar $ITI_{\text{ótimo}}$

RA	EP	EP-SE	SE	SE-EP
1	1025x1025	-	-	-
2	257x513	257x513	129x257	257x513
16	33x513	33x513	33x513	33x513
128	9x1025	9x1025	9x1025	17x2049

A tabela 7.3 mostra o $ITI_{\text{ótimo}}$ para as malhas da tabela 7.2. Pode-se notar que o $ITI_{\text{ótimo}}$ é fortemente afetado pelo valor de RA e também pelo algoritmo utilizado, exceto o algoritmo SE-EP. Os algoritmos EP, EP-SE apresentam um comportamento crescente e acentuado do $ITI_{\text{ótimo}}$ à medida que se aumenta RA . O algoritmo SE apresentou o comportamento oposto. Finalmente, o algoritmo SE-EP apresentou o mesmo comportamento crescente dos algoritmos EP e EP-SE, porém de uma forma muito mais suave e parece ter estabilizado em $ITI_{\text{ótimo}} = 3$.

Tabela 7.3: Valor do $ITI_{\text{óptimo}}$

RA	EP	EP-SE	SE	SE-EP
1	1	-	-	-
2	2	2	350	2
16	70	82	25	3
128	450	250	7	3

7.3.2 Efeito do número de incógnitas (N) sobre a razão de aspecto (RA)

As figuras 7.3 a 7.5 apresentam uma comparação entre os diversos algoritmos (EP, EP-SE, SE e SE-EP) para as razões de aspecto $RA = 2, 16$ e 128 . Estas figuras mostram o tempo de CPU em função do número de incógnitas (N).

Na Fig. 7.3, onde se estuda o comportamento dos algoritmos para $RA = 2$, pode-se notar que o algoritmo SE-EP é o mais rápido entre os quatro algoritmos testados. Ele é seguido de perto pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos, e finalmente pelo algoritmo SE, com tempo de CPU muito maior. Nesta figura, as seguintes malhas foram usadas: $65 \times 129, 129 \times 257, 257 \times 513, 513 \times 1025$ e 1025×2049 para os algoritmos EP e EP-SE; $33 \times 65, 65 \times 129, 129 \times 257$ e 257×513 para o algoritmo SE; e $65 \times 129, 129 \times 257, 257 \times 513, 513 \times 1025, 1025 \times 2049$ e 2049×4097 para o algoritmo SE-EP.

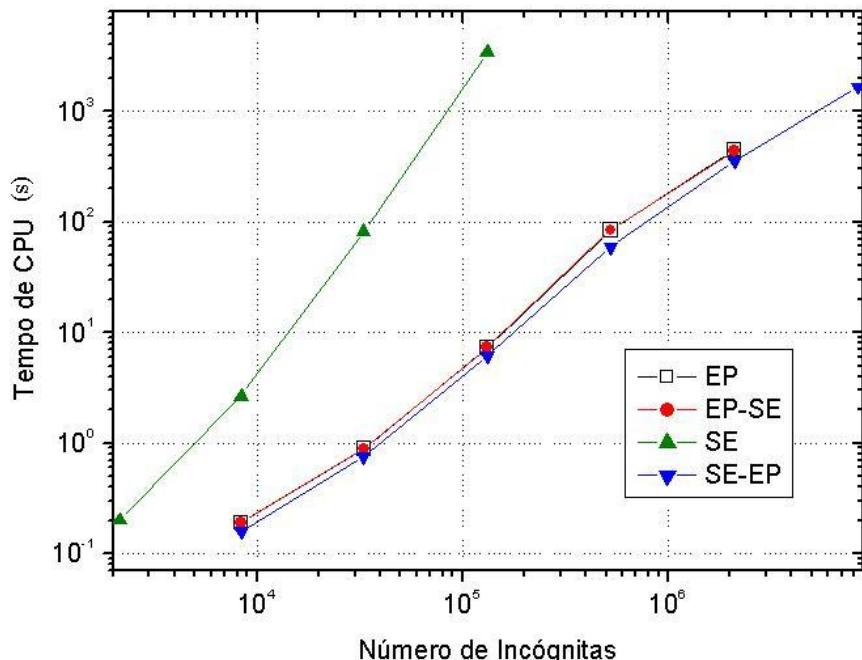


Figura 7.3: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, $RA = 2$, com o esquema numérico CDS/CDS e diversos algoritmos

Conforme a figura 7.4, onde se estuda o comportamento dos algoritmos para $RA = 16$, o algoritmo SE-EP também é o mais rápido. Ele é seguido pelo algoritmo SE em uma primeira parte dos pontos do domínio e a partir daí pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos em praticamente todos os pontos do domínio.

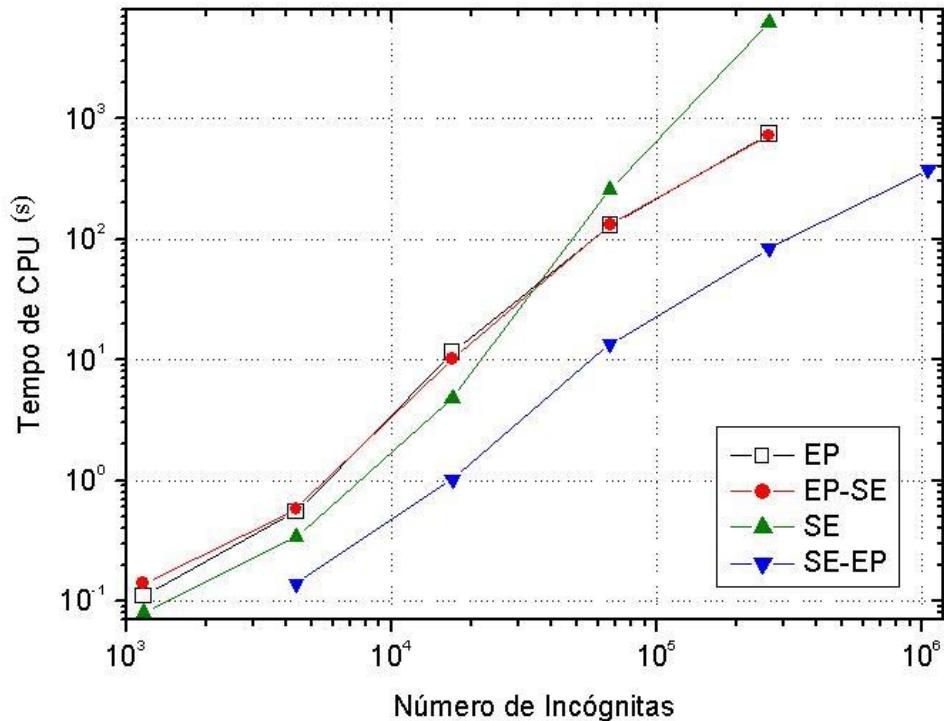


Figura 7.4: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, $RA = 16$, com o esquema numérico CDS/CDS e diversos algoritmos

Na figura 7.4, as seguintes malhas foram usadas: 9x129, 17x257, 33x513, 65x1025 e 129x2049 para os algoritmos EP, EP-SE e SE; e 17x257, 33x513, 65x1025, 129x2049 e 257x4097 para o algoritmo SE-EP.

Na figura 7.5, onde se estuda o comportamento dos algoritmos para $RA = 128$, pode-se notar que o algoritmo SE-EP continua sendo o mais rápido. Ele é seguido pelo algoritmo SE e finalmente pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos. Nesta figura, as seguintes malhas foram usadas: 5x513, 9x1025 e 17x2049 para o algoritmo EP; 5x513 e 9x1025 para o algoritmo EP-SE; 5x513, 9x1025, 17x2049 e 33x4097 para o algoritmo SE; e 5x513, 9x1025, 17x2049, 33x4097, 65x8193 e 129x16385 para o algoritmo SE-EP.

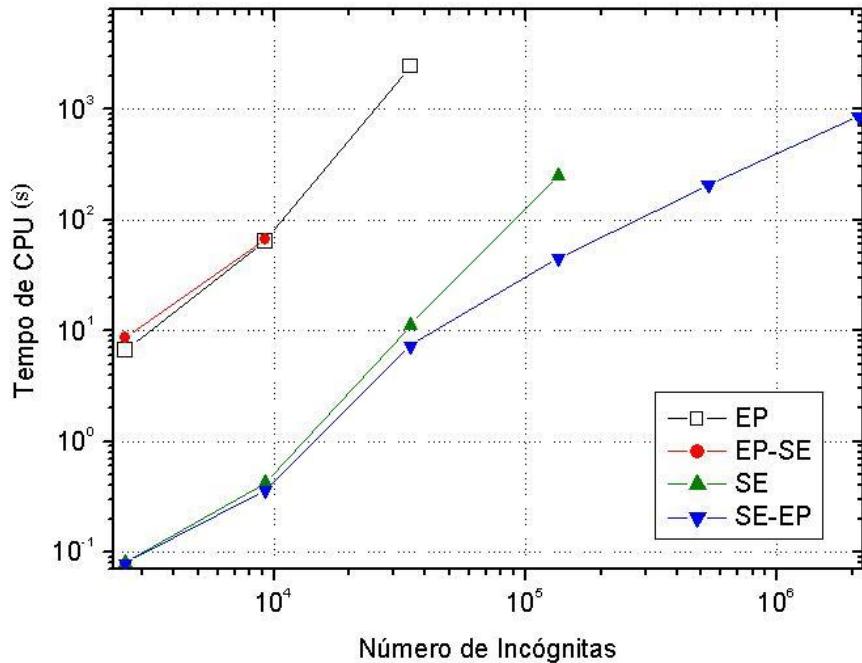


Figura 7.5: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, $RA = 128$, com o esquema numérico CDS/CDS e diversos algoritmos

Portanto, o algoritmo SE-EP de Zhang (2002) é o mais rápido entre os quatro algoritmos e para as razões de aspecto testadas. Pode-se notar que os algoritmos EP e EP-SE se tornam mais lentos à medida que se aumenta RA . O algoritmo SE torna-se mais rápido, mas nunca é melhor que o algoritmo SE-EP. Em resumo, para os mesmos dados (ε , \vec{v} , N e RA), dentre os quatro algoritmos testados, tem-se

$$t_{CPU}(SE - EP) < t_{CPU}(EP, EP - SE, SE), \quad (7.1)$$

em todos os pontos do domínio em estudo. Zhang (2002) já havia constatado que o algoritmo *partial semicoarsening* (aqui chamado de SE-EP) com os solvers *red-black* e *four-color* Gauss-Seidel é o mais rápido, se comparado ao algoritmo EP, mas somente para $RA = 2, 4, 8$ e 16 e usando-se a equação de Poisson bidimensional.

7.3.3 Algoritmo de semi-engrossamento parcial (SE-EP)

Pelo fato do algoritmo SE-EP ser o mais rápido entre os quatro algoritmos testados, foi analisado o seu comportamento para alguns valores de RA ainda maiores. A tabela 7.4

mostra o número de incógnitas das malhas ($N = N_x N_y$) e as respectivas razões de aspecto complementares ($RA = 1/1024, 1024$ e 8192) utilizadas para obter o $ITI_{\text{ótimo}}$. Os resultados para o $ITI_{\text{ótimo}}$ são apresentados na tabela 7.5.

Tabela 7.4: Malhas adicionais usadas para obter $ITI_{\text{ótimo}}$ com o algoritmo SE-EP

RA	1/1024	1024	8192
N	8193x9	9x8193	9x131072

Tabela 7.5: $ITI_{\text{ótimo}}$ para o algoritmo SE-EP

RA	1/1024	1024	8192
$ITI_{\text{ótimo}}$	3	3	3

Pode-se notar nas tabelas 7.3 e 7.5 que $ITI_{\text{ótimo}}$, para o algoritmo SE-EP, é fracamente afetado por RA , sendo igual a três, exceto num caso.

A tabela 7.6 mostra o valor do expoente p obtido por ajuste geométrico de mínimos quadrados, dado pela equação 6.10, para os resultados da figura 7.6 cujas malhas usadas foram: 4097x5, 8193x9 e 16385x17 para a $RA = 1/1024$; 129x129, 257x257, 513x513, 1025x1025, 2049x2049 para a $RA = 1$; 65x129, 129x257, 257x513, 513x1025, 1025x2049 e 2049x4097 para a $RA = 2$; 17x257, 33x513, 65x1025, 129x2049, 257x4097 e 513x8193 para a $RA = 16$; 9x1025, 17x2049, 33x4097, 65x8193 e 129x16385 para a $RA = 128$; 5x4097, 9x8193, 17x16385 e 33x32769 para a $RA = 1024$ e 5x32769, 9x65537 e 17x131073 para a $RA = 8192$.

Tabela 7.6: Valor do coeficiente (c) e ordem (p) da equação 6.10 para o algoritmo SE-EP

RA	c	p
1/1024	3.00×10^{-9}	1.93
1	7.61×10^{-7}	1.35
2	5.30×10^{-7}	1.39
16	1.76×10^{-6}	1.38
128	1.30×10^{-6}	1.42
1024	5.68×10^{-7}	1.49
8192	2.15×10^{-7}	1.51

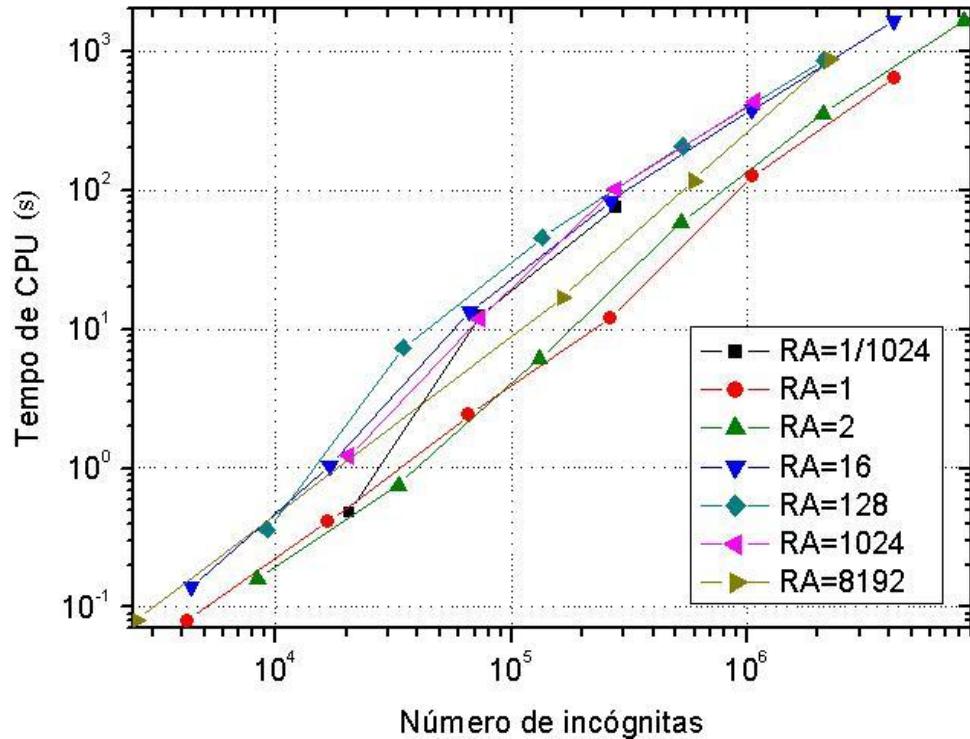


Figura 7.6: Tempo de CPU versus N para a equação de Laplace com esquema CS, $r = 2$, diversos RA , com o esquema numérico CDS/CDS e algoritmo SE-EP

Pode-se observar que $p(1/RA) > p(RA)$. E, em geral, dado $1 \leq RA_1 < RA_2$, tem-se que $p(RA_1) < p(RA_2)$. O comportamento crescente de p em função da $RA \geq 1$ só é interrompido para $RA = 16$. O crescimento de p em função da RA não ocorre de uma forma acentuada, ou seja, há uma fraca dependência entre p e RA .

Pode-se observar na figura 7.6 que, em geral

$$\begin{aligned} t_{CPU}(RA=1) &< t_{CPU}(RA=2) < t_{CPU}(RA=8192) < \\ t_{CPU}(RA=16) &< t_{CPU}(RA=1024) < t_{CPU}(RA=128). \end{aligned} \quad (7.2)$$

A primeira desigualdade da equação 7.2 era esperada, pois trata-se da comparação entre um problema isotrópico e um problema anisotrópico com a malha levemente alongada, ou seja, razão de aspecto próxima da unidade ($0 < RA \approx 1$). A segunda desigualdade impressiona, pois trata-se da comparação entre dois problemas anisotrópicos, porém com razões de aspecto bem distintas: uma levemente alongada e a outra fortemente alongada, ou seja, razão de aspecto muito distinta da unidade ($RA \gg 1$).

De uma forma geral, pode-se concluir da equação 7.2 que, razões de aspecto próximas da unidade ou muito distintas da unidade são razões de aspecto para as quais o algoritmo SE-EP possui um bom desempenho, por exemplo, $RA = 2$ e 8192 . Por outro lado, para razões de aspecto intermediárias, por exemplo, $RA = 128$, o algoritmo SE-EP possui um dos piores desempenhos entre as razões de aspecto testadas, mas mesmo assim, ainda é melhor que os outros algoritmos (EP, EP-SE e SE). Veja figura 7.5.

Isto pode ser explicado pelo fato de que para problemas anisotrópicos com malhas levemente alongadas ($0 < RA \approx 1$), o algoritmo SE-EP comporta-se de forma muito semelhante ao problema isotrópico, onde o *multigrid* funciona bem. Por outro lado, para problemas anisotrópicos com malhas fortemente alongadas ($RA \gg 1$), o algoritmo SE-EP comporta-se de forma muito semelhante aos problemas unidimensionais, devido à predominância de semi-engrossamentos; e neste caso, o *multigrid* também funciona bem.

Os resultados observados na figura 7.6 são mais confiáveis do que os resultados da tabela 7.6, pois estes últimos resultam do ajuste de curvas, que depende fortemente da quantidade e qualidade dos dados observados, tendo portanto, erros. E mais, os resultados da tabela 7.6 fornecem apenas uma aproximação da ordem de complexidade dos algoritmos para as diversas razões de aspecto, enquanto que os resultados da figura 7.6 fornecem o tempo de CPU em função do número de incógnitas (N).

7.4 Conclusão do capítulo 7

Neste capítulo foi resolvido numericamente o mesmo problema do capítulo anterior: equação de Laplace bidimensional, com malhas anisotrópicas. O esquema numérico de aproximação usado foi CDS/CDS. Foi usado o esquema CS (*Correction Scheme*) do método *multigrid*. Foram apresentados quatro algoritmos utilizados para problemas anisotrópicos: EP (engrossamento padrão), EP-SE (engrossamento padrão seguido de semi-engrossamento), SE (semi-engrossamento) e SE-EP (semi-engrossamento seguido de engrossamento padrão).

Com base nos resultados deste capítulo, verificou-se que:

1. Para os mesmos dados (ε , v , N e RA), o algoritmo SE-EP resulta no menor tempo de CPU entre os quatro algoritmos testados para o problema anisotrópico;
2. A variação de RA resulta em grande variação do $ITI_{ótimo}$ para os algoritmos EP, EP-SE e SE, e pequena variação para o algoritmo SE-EP.

3. O $ITI_{óptimo}$ para o algoritmo SE-EP é o mesmo para RA e $1/RA$, e para as razões de aspecto muito distintas da unidade, ou seja, $0 < RA \ll 1$ ou $RA \gg 1$;
4. A ordem do algoritmo SE-EP (p) cresce com RA ;
5. Para os mesmos dados (ε , v , N e RA), o algoritmo SE-EP resulta em menor tempo de CPU para as malhas levemente ($RA \approx 1$) e fortemente alongadas ($RA \gg 1$); e maior tempo de CPU para as malhas com RA intermediário.

Recomendações práticas:

Com base nos resultados deste capítulo são feitas as seguintes recomendações aos usuários do método *multigrid* para problemas iguais ou similares ao estudado.

Problemas bidimensionais lineares anisotrópicos:

- Algoritmo: SE-EP;
- Número de iterações internas: $ITI = 3$.

8 Conclusão

Neste trabalho foram resolvidos numericamente quatro problemas: equações de Poisson, advecção-difusão e Burgers, todas unidimensionais e equação de Laplace bidimensional. Estas equações são todas lineares, exceto a equação de Burgers. Elas representam problemas de transferência de calor e escoamento de fluido. Os esquemas numéricos de aproximação usados foram: CDS, UDS/CDS e CDS_CA/CDS. Foram descritos e usados dois esquemas do método *multigrid*: esquemas CS (*Correction Scheme*) e FAS (*Full Approximation Scheme*). Foram apresentados quatro algoritmos utilizados no caso de problemas anisotrópicos: EP (engrossamento padrão), EP-SE (engrossamento padrão seguido de semi-engrossamento), SE (semi-engrossamento) e SE-EP (semi-engrossamento seguido de engrossamento padrão).

Com base nos resultados deste trabalho, verificou-se que, para as equações estudadas:

1. Para os mesmos dados (ε , v e N), o esquema FAS é mais rápido do que o esquema CS, para os problemas lineares (uni e bidimensionais) e não-linear (unidimensional);
2. O *solver* MSI é mais rápido do que o Gauss-Seidel e o ADI, para os esquemas CS e FAS. Portanto, mesmo com o método *multigrid*, quanto mais fortemente implícito é o *solver*, mais rápido ele é;
3. O número de iterações internas (ITI) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) e a dimensão do problema (1D ou 2D) influencia o $ITI_{óptimo}$. Para o caso unidimensional com uso do esquema CS, $ITI_{óptimo} \approx 2r$, e com uso do esquema FAS, $ITI_{óptimo} \approx 3r$. Para o caso bidimensional com o uso do esquema CS, $ITI_{óptimo} = 1$ com qualquer *solver*, e com o uso do esquema FAS, $ITI_{óptimo} = 3$ a 5, dependendo do *solver*;
4. O número de níveis de malhas (L) afeta significativamente o tempo de CPU para os esquemas CS e FAS. O esquema utilizado (CS ou FAS) não tem muita influência no $L_{óptimo}$. Para ambos os esquemas e para os *solvers* testados, $L_{óptimo} \approx L_{máximo}$;

5. Para os mesmos dados (ε, v e N), com $L = L_{máximo}$ para cada N , entre as razões de engrossamento de malhas testadas ($r = 2, 3, 4$ e 5), o esquema CS é mais rápido com $r = 2$; já o esquema FAS é mais rápido com $r = 3$;
6. O esquema FAS (com razão de engrossamento $r = 3$) é aproximadamente 4 vezes mais rápido do que o esquema CS (com razão de engrossamento $r = 2$), tanto para problemas unidimensionais como para problemas bidimensionais;
7. O comportamento qualitativo dos parâmetros estudados do método *multigrid* é idêntico para problemas lineares e não-lineares unidimensionais;
8. Para os mesmos dados (ε, v, N e RA), o algoritmo SE-EP resulta no menor tempo de CPU entre os quatro algoritmos testados para o problema anisotrópico;
9. A variação de RA resulta em grande variação do $ITI_{ótimo}$ para os algoritmos EP, EP-SE e SE, e pequena variação para o algoritmo SE-EP.
10. O $ITI_{ótimo}$ para o algoritmo SE-EP é o mesmo para RA e $1/RA$, e para as razões de aspecto muito distintas da unidade, ou seja, $0 < RA \ll 1$ ou $RA \gg 1$;
11. A ordem do algoritmo SE-EP (p) cresce com RA ;
12. Para os mesmos dados (ε, v, N e RA), o algoritmo SE-EP resulta em menor tempo de CPU para as malhas levemente ($RA \approx 1$) e fortemente alongadas ($RA \gg 1$); e maior tempo de CPU para as malhas com RA intermediário.

Contribuições desta tese:

Esta tese contribui com a literatura existente no sentido de que:

- a) foram apresentados operadores de restrição por injeção e prolongação por interpolação linear generalizados para quaisquer razões de engrossamento $r \in (1, \infty)$ em problemas unidimensionais;
- b) verificou-se que o esquema FAS é mais rápido do que o esquema CS, mesmo em problemas lineares;
- c) verificou-se que $r = 2$ perde a hegemonia sobre outras razões de engrossamento, por exemplo, para $r = 3$ quando se utiliza o esquema FAS;
- d) constatou-se que para problemas anisotrópicos, o algoritmo SE-EP é o mais adequado para razões de aspecto bem distintas da unidade, problemas de interesse da engenharia.

Recomendações práticas:

Com base nos resultados desta tese são feitas as seguintes recomendações aos usuários do método *multigrid* para problemas iguais ou similares aos estudados.

Problemas unidimensionais lineares e não-lineares:

- Esquema: FAS;
- Razão de engrossamento: $r = 3$;
- Número de iterações internas: $ITI = 9$;
- Número de níveis de malhas: $L = L_{máximo}$.

Problemas bidimensionais lineares isotrópicos:

- Esquema: FAS;
- *Solver*: MSI;
- Razão de engrossamento: $r = 3$;
- Número de iterações internas: $ITI = 3$;
- Número de níveis de malhas: $L = L_{máximo}$.

Problemas bidimensionais lineares anisotrópicos:

- Algoritmo: SE-EP;
- Número de iterações internas: $ITI = 3$.

Extrapolação:

Os problemas abordados nesta Tese são problemas modelados por equações simples, onde todos apresentam soluções analíticas conhecidas. Espera-se que as conclusões sirvam também para equações mais gerais e para problemas que não possuam solução analítica conhecida. Portanto, apresentam-se a seguir algumas possibilidades de extrapolação dos resultados desta Tese:

- Equação de advecção-difusão bidimensional. Aqui tem-se os efeitos dos coeficientes constantes e variáveis. No último caso (coeficientes variáveis), por se tratar de anisotropia física, pode-se estender a aplicação dos algoritmos para anisotropia geométrica vistos no capítulo 7;
- Equação de termoelasticidade bidimensional. Aqui tem-se os efeitos do acoplamento entre as variáveis de interesse. Podem-se estender os aspectos qualitativos dos parâmetros estudados para a equação de Laplace

bidimensional ($ITI_{ótimo}$, $L_{ótimo}$, $r_{ótima}$, CS versus FAS), tanto na fase térmica do processo de resolução, como na fase elástica;

- Equação de Navier-Stokes bidimensional. Aqui tem-se o acoplamento pressão/velocidade. Podem-se aplicar os aspectos qualitativos dos parâmetros estudados para a equação de Laplace bidimensional ($ITI_{ótimo}$, $L_{ótimo}$, $r_{ótima}$, CS versus FAS), na equação de Poisson envolvida na resolução no ciclo da massa (Larsson et al., 2005 e Souza, et al., 2006).

Proposta de trabalhos futuros:

Apresentam-se a seguir algumas questões que abrem caminhos para novas pesquisas.

- **Equação de advecção-difusão bidimensional:** poucos aspectos qualitativos com relação ao $ITI_{ótimo}$, $L_{ótimo}$, $r_{ótima}$ e esquemas CS versus FAS variaram para o caso unidimensional e bidimensional. Acredita-se que tais propriedades também devam valer para a equação de advecção-difusão para o caso bidimensional. Nesta etapa podem-se estudar também a influência do número de Peclet em cada uma das direções coordenadas, o ângulo do vetor de velocidades, etc.
- **Equação de termoelasticidade bidimensional:** nada se estudou com relação a sistema de equações diferenciais. A equação de termoelasticidade é uma boa equação para se começar os estudos de propriedades do método *multigrid* em sistemas de equações, devido ao acoplamento entre as variáveis de interesse.
- **Equação de Navier-Stokes:** este é outro sistema de equações diferenciais com características bem peculiares. Pode-se analisar a influência do acoplamento pressão/velocidade no efeito *multigrid*. Devem-se analisar vários tipos de acoplamento, diferentes funções de interpolação, diferentes operadores de restrição e prolongação, etc.
- **Aspectos qualitativos no método dos volumes finitos:** todas as conclusões deste trabalho são extraídas do método das diferenças finitas. Propõe-se avaliar os aspectos qualitativos deste estudo alterando-se o método para o método dos volumes finitos.
- **Problemas anisotrópicos:** estudou-se apenas anisotropia geométrica, ou seja, anisotropia da malha. Propõe-se ampliar o estudo de anisotropia geométrica (outras anisotropias de malha) e também de anisotropia física (anisotropia dos coeficientes),

além de realizar um estudo detalhado sobre as diversas razões de engrossamento, *solvers* e esquemas CS e FAS para problemas anisotrópicos diversos;

- **Multigrid algébrico:** estudaram-se apenas problemas em malhas estruturadas com o uso do *multigrid* geométrico. Propõe-se ampliar o estudo em malhas não-estruturadas com o uso do *multigrid* algébrico, além de realizar um estudo comparando a eficiência do *multigrid* geométrico e algébrico, ambos fazendo-se uso de malhas estruturadas.

9 Referências bibliográficas

- 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 Interim Report, n. 32, NASA/CR, 1998.
- BRIGGS, W.L.; HENSON, V.E., MCCORMICK, S.F., "A *Multigrid Tutorial*", 2^a ed., SIAM, 2000.
- BURDEN, R. L.; FAIRES, J. D., "Numerical Analysis", 6^a ed., Brooks/Cole Publishing Company, 1997.
- DENDY Jr., J. E., MCCORMICK, S. F., RUGE, J. W., RUSSEL, T. F., SCHAFFER, T., "Multigrid Methods for Three-dimensional Petroleum Reservoir Simulation", SPE 18409, p. 19-25, 1989.
- FEDORENKO, R. P., "On the Speed of Convergence of an Iteration Process", USSR Comput. Math. and Math. Phys., v. 4 (3), 1964.
- FERM, L.; LÖTSTEDT, P., "Anisotropic grid adaptation for Navier-Stokes' equations", Journal of Computational Physics, v. 190, p. 22-41, 2003.
- FERZIGER, J. H. e PERIC, M., "Computational Methods for Fluids Dynamics", 2^a ed., Springer, 1999.
- FORTUNA, A.O., Técnicas computacionais para dinâmica dos fluidos, Edusp, 2000.
- 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.

GOLUB, G. H. e ORTEGA, J. M., "Scientific Computing and Differential Equations: an Introduction to Numerical Methods", Academic Press, Inc., 1992.

GOLUB, G. H. e VAN LOAN, C., "Matrix Computations", 2^a ed., Johns Hopkins Press, Baltimore, 1989.

GREENBERG, M. D., "Advanced Engineering Mathematics", 2^a ed., Prentice Hall, Inc., New Jersey, 1998.

HAM, F. E.; LIEN, F. S.; STRONG, A. B., "A Cartesian Grid Method with Transient Anisotropic Adaptation", Journal of Computational Physics, v. 179, p. 469-494, 2002.

HESTENES, M. e STIEFEL, E., "Methods of Conjugate Gradients for Solving Linear Systems", Journal of Research of National Bureau of Standards, v. 49, p. 409-436, 1952.

HACKBUSCH, W., "*Multi-grid* Methods and Applications", Springer-Verlag, Berlim, 1985.

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.

HUTCHINSON, B.R.; RAITHBY, G.D., "A *Multigrid* Method Based on the Additive Correction Strategy", Numerical Heat Transfer, v. 9, p. 511-537, 1986.

KARKI, K.C.; SATHYAMURTHY, P.S.; PATANKAR, S.V., "Performance of a *Multigrid* Method with an Improved Discretization Scheme Flow Calculations", Numerical Heat Transfer, Part B, v. 29, p. 275-288, 1996.

KELKER, K.M., "Iterative Method for the Numerical Prediction of Heat Transfer in Problems Involving Large Difference in Thermal Conductivities", Numerical Heat Transfer, v. 17, p. 113-128, 1990.

LAI, Y.G.; PRZEKWA, A.J., "A *Multigrid* Algorithm for a Multiblock Pressure-based Flow and Heat Transfer Solver", Numerical Heat Transfer, Part B, v. 30, p. 239-254, 1996.

LARSSON, J., LIEN, F. S., YEE, E., “Conditional Semicoarsening *Multigrid* Algorithm for the Poisson Equation on Anisotropic Grids”, Journal of Computational Physics, v. 208, p. 368-383, 2005.

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.

MALISKA, C.R., Transferência de Calor e Mecânica dos Fluidos Computacional, LTC, 2^a ed., 2004.

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., LLORENSE, I. M., SALAS, M. D., “Robust *Multigrid* Algorithm for Navier-Stokes Equations”, Journal of Computational Physics, v. 173, p. 412-432, 2001.

MORO FILHO, R. C., “Aplicação da Técnica *Multigrid* em Transferência de Calor Computacional”, XXV Iberian Latin-American Congress in Computational Methods in Engineering – CILAMCE 2004, Recife – PE, Brazil, 2004.

MULDER, W.A., “A New *Multigrid* Approach to Convection Problems”, Journal of Computational Physics, v. 83, p. 303-323, 1989.

NAIK, N.H.; VAN ROSENDALE, J., “The Improved Robustness of *Multigrid* Elliptic Solvers Based on Multiple Semicoarsened Grids”, SIAM Journal in Numerical Analysis, v. 30, p. 215-229, 1993.

NISHIDA, H.; SATOFUKA, N., “Higher-order Solutions of Square Driven Cavity Flow Using a Variable-order *Multigrid* Method”, International Journal for Numerical Methods in Engineering, v. 34, p. 637-653, 1992.

OOSTERLEE, C.W.; WESSELING, P., “A Robust *Multigrid* Method for a Discretization of the Incompressible Navier-Stokes Equations in General Coordinates”, Impact of Computing in Science and Engineering, v. 5, p. 128-151, 1993.

PERNG, C.Y.; STREET, R.L., “A Coupled *Multigrid*-Domain-Splitting Technique for Simulating Incompressible Flows in Geometrically Complex Domains”, International Journal for Numerical Methods in Fluids, v. 13, p. 269-286, 1991.

PIERCE, N.A.; ALONSO, J.J., “Efficient Computation of Unsteady Viscous Flows by an Implicit Preconditioned *Multigrid* Method”, AIAA Journal, v. 36, n. 3, p. 401-408, 1998.

RADESPIEL, R. e SWANSON, R.C., “Progress with *Multigrid* Schemes for Hyperbolic Flow Problems”, Journal of Computational Physics, v. 116, p. 103-122, 1995.

ROACHE, P. J., “Fundamentals of Computational Fluid Dynamics”, Hermosa Publishers, 1998.

RUBINI, P.A., BECKER, H. A., GRANDMAISON, E. W., POLLARD, A., SOBIESIAK, A., THURGOOD, C., “*Multigrid* Acceleration of Three-dimensional, Turbulent, Variable-density Flows”, Numerical Heat Transfer, v. 22, p. 163-177, 1992.

SATHYAMURTHY, P.S.; PATANKAR, S.V., “Block-Correction-Based *Multigrid* Method for Fluid Flow Problems”, Numerical Heat Transfer, Part B, v. 25, p. 375-394, 1994.

SCHNEIDER, G. E., ZEDAN, M., “A Modified Strongly Implicit Procedure for Numerical Solution of Field Problems”, Numerical Heat Transfer, v. 4, p. 1-19, 1981.

SETTAR, A., AZIZ, K., “A Generalization of the Additive Correction Methods for the Iterative Solution of Matrix Equations”, SIAM Journal of Numerical Analysis, v. 10, p. 506-521, 1973.

SIVALOGANATHAN, S., “The Use of Local Mode Analysis in the Design in Comparison of *Multigrid* Methods”, Computer Physics Communications, v. 65, p. 246-252, 1991.

SMITH, K.M.; COPE, W.K.; VANKA, S.P., “A *Multigrid* Procedure for Three-dimensional Flows on Non-orthogonal Collocated Grids”, International Journal for Numerical Methods in Fluids, v. 17, p. 887-904, 1993.

SOCKOL, P.M., “*Multigrid* Solution of the Navier-Stokes Equations on Highly Stretched Grids”, International Journal for Numerical Methods in Fluids, v. 17, p. 543-566, 1993.

SOUZA, L. F., OISHI, C. M., CUMINATO, J. A., SILVEIRA NETO, A., “Comparison of Poisson Solvers in the Transient 2D Fluid Flow Problem with Dirichlet Boundary Conditions”, Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering – ENCIT 2006, Curitiba – PR, Brazil, 2006.

STÜBEN, K., “Algebraic *Multigrid* (AMG): An Introduction with Applications”, em: GMD-Report 70, Novembro 1999.

STÜBEN, K., “A Review of Algebraic *Multigrid*”, Journal of Computation and Applied Mathematics, v. 128, p. 281-309, 2001.

TANNEHILL, J. C.; ANDERSON, D. A.; PLETCHER, R. H., “Computational Fluid Mechanics and Heat Transfer”, 2^a ed., Washington: Taylor & Francis, 1997.

VERSTEEG, H. K., MALALASEKERA, W., “An Introduction to Computational Fluid Dynamic, The Finite Volume Method”, England: Longman, 1995.

TROTTERBERG, U., OOSTERLEE, C., SCHÜLLER, A., “*Multigrid*”, Academic Press, 2001.

WESSELING, P., “An Introduction to *Multigrid* Methods”, John Wiley & Sons, 1992.

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

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.

ZHANG, J., “*Multigrid* Method and Fourth-Order Compact Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization”, Journal of Computational Physics, v. 179, p. 170-179, 2002.

Apêndice A - Métodos iterativos básicos e outros solvers

Neste apêndice são apresentados alguns dos métodos iterativos básicos e estudam-se alguns dos métodos de solução dos sistemas de equações algébricas que serão tratados neste trabalho.

A.1 Exemplos de métodos iterativos básicos

Esta seção será dedicada à apresentação dos métodos iterativos básicos mais conhecidos, mostrando-se apenas como deve ocorrer a divisão $A = M - N'$:

i) Método de Jacobi por pontos⁹

Este esquema de relaxação pode ser expresso em forma matricial. Separando-se a matriz A na forma

$$A = D - L - U, \quad (\text{A.1})$$

onde D é a diagonal de A , e $-L$ e $-U$ são estritamente as partes inferior e superior de A , respectivamente. No caso do método de Jacobi tem-se $M = \text{diag}(A) = D$. Assim

$$Ax = b \Rightarrow (D - L - U)x = b \Rightarrow Dx = (L + U)x + b \Rightarrow x = D^{-1}(L + U)x + D^{-1}b. \quad (\text{A.2})$$

Definindo-se

$$S_J = D^{-1}(L + U), \quad (\text{A.3})$$

o método de Jacobi na forma matricial dar-se-á por

$$x \leftarrow S_J x + D^{-1}b. \quad (\text{A.4})$$

ii) Método de Jacobi Ponderado:

Na forma matricial, tem-se

⁹ Ou simplesmente Método de Jacobi.

$$x^{(m+1)} = [(1-\omega)I + \omega S_J]x^{(m)} + \omega D^{-1}b, \quad (\text{A.5})$$

onde $\omega \in R$ é chamado de fator de ponderação. Definindo-se a matriz de iteração para o método de Jacobi ponderado como $S_\omega = (1-\omega)I + \omega S_J$, tem-se

$$x \leftarrow S_\omega x + \omega D^{-1}b. \quad (\text{A.6})$$

Nota-se que se $\omega = 1$, tem-se o método de Jacobi original.

iii) Método de Jacobi por linhas

Neste caso tem-se M obtida de A trocando-se todo a_{ij} por zero, para todo i, j com $j \neq i$, $j \neq i-1$ e $j \neq i+1$.

iv) Método Gauss-Seidel por pontos¹⁰

Novamente representa-se a matriz A na forma dada pela equação A.1. Neste caso M é obtida de A trocando-se todo a_{ij} por zero, para todo i, j com $j > i$. Tem-se

$$Ax=b \Rightarrow (D-L-U)x=b \Rightarrow (D-L)x=Ux+b \Rightarrow x=(D-L)^{-1}Ux+(D-L)^{-1}b.$$

Definindo-se a matriz de iteração para o método de Gauss-Seidel por

$$S_G = (D-L)^{-1}U, \quad (\text{A.7})$$

pode-se expressar o método de Gauss-Seidel como

$$x \leftarrow S_G x + (D-L)^{-1}b. \quad (\text{A.8})$$

v) Método Gauss-Seidel por linhas

Neste caso M é obtida de A trocando-se todo a_{ij} por zero, para todo i, j com $j > i+1$.

¹⁰ Ou simplesmente Método de Gauss-Seidel.

Nota-se que os métodos do tipo relaxação por linhas, discutidos anteriormente, exigem a solução de um sistema tridiagonal. Para a resolução deste tipo de algoritmo, deve-se recorrer ao algoritmo de Thomas, mais conhecido como, algoritmo para matrizes tridiagonais¹¹ (TDMA), que será visto como um dos métodos para sistemas de equações do apêndice A.2.

A.2 Métodos para a resolução de sistemas de equações

Nesta seção são descritos alguns dos métodos numéricos (diretos e iterativos) usados neste trabalho para a resolução dos sistemas de equações, também chamados de *solvers*. Começa-se primeiramente pelo método direto TDMA, que é usado para o caso unidimensional, em seguida descreve-se o método de Gauss-Seidel com a versão para o caso bidimensional e a versão unidimensional sendo um caso particular desta. Finalmente descreve-se o método MSI¹² para o caso bidimensional.

Os algoritmos tratados nesta seção consideram um sistema linear do tipo $Au = f$. Caso tenha-se um sistema não-linear do tipo $A(u) = f$ eles podem ser usados igualmente, mas tomando-se o cuidado de haver um ciclo externo para a atualização dos coeficientes da matriz A do sistema.

Método TDMA

As matrizes de banda tridiagonais são freqüentes na dinâmica dos fluidos computacional, particularmente, em problemas unidimensionais. O algoritmo para a resolução de sistemas tridiagonais (GOLUB e VAN LOAN, 1989) que será descrito nesta seção é chamado de TDMA (FORTUNA, 2000 e MALISKA, 2004). Ele é conhecido também como algoritmo de Thomas. Seja a matriz A dos coeficientes tridiagonal não-singular

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & \dots & 0 & a_N & b_N \end{bmatrix}, \quad (\text{A.9})$$

¹¹ *Tridiagonal Matrix Algorithm* (TDMA).

¹² *Modified Strongly Implicit Method* (MSI).

cuja decomposição LU é da forma

$$L = \begin{bmatrix} \beta_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \alpha_2 & \beta_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & \alpha_3 & \beta_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \alpha_{N-1} & \beta_{N-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & \alpha_N & \beta_N \end{bmatrix} \quad (\text{A.10})$$

e

$$U = \begin{bmatrix} 1 & \gamma_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \gamma_2 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 & \gamma_{N-1} \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}. \quad (\text{A.11})$$

Pode-se determinar os valores de α_i , β_i e γ_i das matrizes A.10 e A.11 em função dos elementos da matriz A.9 comparando-se A com os respectivos elementos do produto LU . Dessa comparação obtém-se

$$\begin{cases} \beta_1 = b_1, & \gamma_1 = c_1/\beta_1 \\ 2 \leq i \leq N-1, & \alpha_i = a_i, \quad \beta_i = b_i - a_i\gamma_{i-1}, \quad \gamma_i = c_i/\beta_i \\ \alpha_N = a_N, & \beta_N = b_N - a_N\gamma_{N-1} \end{cases} \quad (\text{A.12})$$

Uma vez determinado os valores de todos os elementos de L e U , o sistema original $Au = f$ é escrito da seguinte forma

$$Au = f \Rightarrow (LU)u = f \Rightarrow Ls = f, \quad \text{com} \quad Uu = s.$$

Para se resolver o sistema $Ls = f$, tem-se a forma progressiva

$$\begin{cases} s_1 = f_1/\beta_1 \\ 2 \leq i \leq N, & s_i = (f_i - \alpha_i s_{i-1})/\beta_i \end{cases} \quad (\text{A.13})$$

Para se resolver o sistema $Uu = s$, tem-se a forma retroativa

$$\begin{cases} u_N = s_N \\ N-1 \leq i \leq 1, \quad u_i = s_i - \gamma_i u_{i+1} \end{cases}. \quad (\text{A.14})$$

Tabela A.1: Procedimento direto TDMA para problemas 1D

Procedimento TDMA para $Au = f$

1. Dados a matriz A e o vetor f do sistema $Au = f$;
2. Determinar os valores de α_i , β_i e γ_i pela equação A.12;
3. Calcular s a partir da equação dada por A.13;
4. Calcular u a partir da equação dada por A.14.

Este método apresenta uma boa eficiência computacional. Ele soluciona um sistema tridiagonal com N equações necessitando de um número de operações aritméticas proporcionais a N , ou seja, o algoritmo TDMA é da ordem de $O(N)$.

Método Gauss-Seidel

As matrizes de banda tridiagonais e pentadiagonais são freqüentes na dinâmica dos fluidos computacional para problemas uni e bidimensionais, respectivamente. Um conhecido método iterativo para a resolução desses sistemas será descrito nesta seção.

Dada uma equação diferencial governante para o caso bidimensional, uma equação algébrica é formada para cada elemento de controle. Veja abaixo um exemplo de um esquema de 5 pontos, usando-se uma notação muito comum em DFC

$$A_{i,j}^p u_{i,j} = A_{i,j}^n u_{i,j+1} + A_{i,j}^s u_{i,j-1} + A_{i,j}^e u_{i+1,j} + A_{i,j}^w u_{i-1,j} + f_{i,j}, \quad (\text{A.15})$$

onde $A_{i,j}^p = a_{i,j}$, $A_{i,j}^n = a_{i,j+1}$, $A_{i,j}^s = a_{i,j-1}$, $A_{i,j}^e = a_{i+1,j}$ e $A_{i,j}^w = a_{i-1,j}$.

Note que para o caso unidimensional, algumas simplificações devem ser feitas. Como, por exemplo, tem-se $A_{i,j}^p = A_i^p$ para todos os superíndices, $u_{i,j} = u_i$ e $f_{i,j} = f_i$ para todos os pares (i, j) e $A_{i,j}^n = A_{i,j}^s = 0$, ou seja,

$$A_i^P u_i = A_i^e u_{i+1} + A_i^w u_{i-1} + f_i . \quad (\text{A.16})$$

Voltando ao caso bidimensional: a coleção de equações na forma da equação A.15, escrita para cada par (i, j) do domínio, produz uma equação matricial da forma $Au = f$. O método de Gauss-Seidel (BURDEN e FAIRES, 1997) é um método iterativo usado para se resolver este tipo de sistema de equações. Esse método resolve o sistema visitando equação por equação, iterativamente, usando-se em um mesmo ciclo, os valores das variáveis já calculadas nesse ciclo iterativo.

Pode-se reescrever a equação A.15 na forma

$$A_p u_P^{k+1} = A_n u_N^k + A_s u_S^{k+1} + A_e u_E^k + A_w u_W^{k+1} + f_P , \quad (\text{A.17})$$

onde o superíndice k representa a k -ésima iteração e o subíndice a posição do nó na malha computacional. Neste caso usou-se o ordenamento lexicográfico, ou seja, de oeste (W) para leste (E) e do sul (S) para o norte (N), podendo-se considerar como conhecidas, na mesma iteração, as variáveis u_W e u_S .

Tabela A.2: Procedimento iterativo de Gauss-Seidel para problemas 1D e 2D

Procedimento de Gauss-Seidel para $Au = f$

1. Dados $k = 0$, k_maximo e u^0 ;
2. Calcular u_P^{k+1} pela equação A.17 para todo P ;
3. Faça k receber $k + 1$;
4. Volte ao passo 2 até convergir ou atingir k_maximo .

Método MSI

Como já foi dito antes matrizes de banda pentadiagonais são freqüentes na dinâmica dos fluidos computacional, especificamente, para problemas bidimensionais. Com a equação diferencial governante para o caso bidimensional forma-se uma equação algébrica para cada elemento de controle. Abaixo tem-se um exemplo para 5 pontos

$$A_{i,j}^s u_{i,j-1} + A_{i,j}^w u_{i-1,j} + A_{i,j}^p u_{i,j} + A_{i,j}^n u_{i,j+1} + A_{i,j}^e u_{i+1,j} = f_{i,j}. \quad (\text{A.18})$$

A coleção de equações na forma da equação A.18, escrita para cada par (i,j) do domínio, produz uma equação matricial da forma $Au = f$. O MSI (Modified Strongly Implicit Method, SCHNEIDER e ZEDAN, 1981), é um método iterativo para se resolver este sistema de equações. Dentre suas vantagens estão:

- remover ou enfraquecer a influência da assimetria das matrizes $[L][U]$;
- reduzir o esforço computacional demandado para se obter a solução convergida (solução numérica após atingir o critério de convergência);
- reduzir a sensibilidade à razão de aspecto da malha ($RA = h_x/h_y$, onde h_x e h_y são os comprimentos do elementos bidimensionais, nas direções x e y respectivamente).

Neste método, uma decomposição LU é proposta, tal que, $[L][U] = [A']$, onde as matrizes $[L]$ e $[U]$ são matrizes triangulares inferior e superior, respectivamente, com U tendo a diagonal unitária. Ao se formar o produto LU , em A' existem duas diagonais adicionais não-nulas denotadas por ϕ_1 e ϕ_2 . Desta forma, a matriz A' pode ser escrita da seguinte forma: $[A'] = [A] + [B]$, onde a matriz B consiste apenas das diagonais ϕ_1 e ϕ_2 .

Os coeficientes das matrizes L e U e os coeficientes adicionais não-nulos que aparecem na matriz modificada A' podem ser determinados diretamente pela definição das equações determinadas pela formação do produto LU . Um procedimento iterativo e corretivo é necessário para anular a forte influência destes termos no sistema de equações modificado. Um parâmetro iterativo α é empregado para o cancelamento parcial da influência dos termos adicionais de A' . Isto é feito, escrevendo-se as equações modificadas de A' para o esquema de 5 pontos da seguinte forma

$$\begin{aligned} & A_{i,j}^s u_{i,j-1} + A_{i,j}^w u_{i-1,j} + A_{i,j}^p u_{i,j} + A_{i,j}^n u_{i,j+1} + A_{i,j}^e u_{i+1,j} + \\ & \phi_{i,j}^1 [u_{i+1,j-1} - \alpha(-2u_{i,j} + 2u_{i+1,j} + u_{i,j-1})] + \\ & \phi_{i,j}^4 [u_{i-2,j+1} - \alpha(-2u_{i,j} + 2u_{i-1,j} + u_{i,j+1})] = f_{i,j}. \end{aligned} \quad (\text{A.19})$$

Tabela A.3: Procedimento iterativo MSI para problemas 2D

Procedimento MSI para $Au = f$
1. Dado: $Au = f$, $it = 1$ e it_max ;
2. Some Bu em ambos os lados: $(A + B)u = f + Bu$;
3. Crie o processo iterativo: $(A + B)u^{k+1} = A'u^{k+1} = f + Bu^k$;
4. Aplique a decomposição LU em A' : $(LU)u^{k+1} = f + Bu^k$;
5. Dados $k = 0$, k_maximo e u^0 ;
6. Resolva por substituição retroativa: $Lv^{k+1} = f + Bu^k$;
7. Resolva por substituição progressiva: $Uu^{k+1} = v^{k+1}$;
8. Faça k receber $k + 1$;
9. Volte ao passo 5 até convergir ou atingir k_maximo .
10. Atualize os coeficientes de A ;
11. Faça it receber $it + 1$;
12. Volte ao passo 2 até convergir ou atingir it_max .

Detalhes sobre a obtenção dos coeficientes das matrizes L e U e as diagonais da matriz B podem ser encontrados em Schneider e Zedan (1981).

No método MSI, o sistema é resolvido em poucas iterações, com um comportamento parecido ao de um método direto, por isto ele é considerado um método fortemente implícito.

Apêndice B - Soluções analíticas para os problemas uni e bidimensionais

Neste apêndice descrevem-se as soluções analíticas dos problemas tratados nesta Tese. No apêndice B.1, encontra-se a solução do problema linear unidimensional (equação de difusão) do capítulo 4 desta Tese. No apêndice B.2, encontra-se a solução do problema linear unidimensional (equação de advecção-difusão) também do capítulo 4. No apêndice B.3, encontra-se a solução do problema não-linear unidimensional (equação de Burgers), referente à equação tratada no capítulo 5. Finalmente, no apêndice B.4, encontra-se a solução do problema linear bidimensional (equação de Laplace) dos capítulos 6 e 7 desta Tese.

B.1 Solução da Equação de difusão 1D

Primeiramente busca-se a solução para a equação de difusão, ou seja, a equação 4.3. Para tal equação, usa-se uma técnica para equações não-homogêneas (GREENBERG, 1998). Busca-se então, uma solução da forma

$$T(x) = T_h(x) + T_p(x), \quad (\text{B.1})$$

onde $T_h(x)$ é a solução da equação homogênea e $T_p(x)$ é uma solução particular da equação não-homogênea. Esta solução particular $T_p(x)$ pode ser dada da seguinte forma

$$T_p(x) = T_{p1}(x) + T_{p2}(x) + \dots + T_{pm}(x), \quad (\text{B.2})$$

onde m é o número de partes em que pode-se dividir o termo fonte.

No caso particular da equação 4.3 e considerando-se especificamente $f(x) = 1 + 3x + 26x^2$, tem-se

$$T_h(x) = C_1 + C_2 x, \quad (\text{B.3})$$

$$T_p(x) = T_{p1}(x) + T_{p2}(x) + T_{p3}(x). \quad (\text{B.4})$$

Usando-se o método dos coeficientes indeterminados (GREENBERG, 1998), pode-se deduzir que, após o processo de integração, tem-se

$$T_{p1}(x) = -\frac{1}{2}x^2,$$

$$T_{p2}(x) = -\frac{1}{2}x^3,$$

$$T_{p3}(x) = -\frac{13}{6}x^4.$$

Portanto, a equação B.4 possui o seguinte formato

$$T_p(x) = -\frac{1}{2}x^2 - \frac{1}{2}x^3 - \frac{13}{6}x^4. \quad (\text{B.5})$$

Com as equações B.3 e B.5 substituídas na equação B.1, tem-se a solução analítica geral

$$T(x) = C_1 + C_2x - \frac{1}{2}x^2 - \frac{1}{2}x^3 - \frac{13}{6}x^4. \quad (\text{B.6})$$

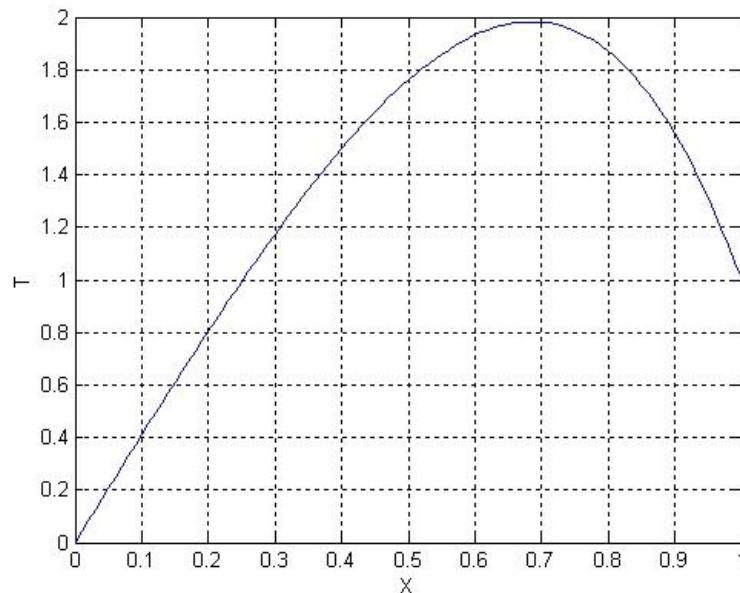
Com as condições de contorno aplicadas à solução analítica geral B.6, o problema dado pela equação 4.4 apresenta a seguinte solução

$$T(x) = \frac{25}{6}x - \frac{1}{2}x^2 - \frac{1}{2}x^3 - \frac{13}{6}x^4. \quad (\text{B.7})$$

A seguir tem-se a tabela B.1 com o valor da solução analítica dada pela equação B.7 em alguns pontos do domínio de cálculo. A figura B.1 ilustra esta solução.

Tabela B.1: Valor da solução analítica para a equação da difusão

Domínio: x	Solução analítica: equação B.7
0,0	0,00000000000000
0,1	0,41095000000000
0,2	0,80586666666667
0,3	1,17395000000000
0,4	1,49920000000000
0,5	1,76041666666667
0,6	1,93120000000000
0,7	1,97995000000000
0,8	1,86986666666667
0,9	1,55895000000000
1,0	1,00000000000000

**Figura B.1: Solução analítica para a equação de difusão**

B.2 Solução da Equação de advecção-difusão 1D

Neste momento busca-se a solução para a equação de advecção-difusão, ou seja, a equação 4.3a. Para tal usa-se uma técnica para equações homogêneas (GREENBERG, 1998). Busca-se uma solução da forma

$$T(x) = e^{\lambda x}. \quad (\text{B.8})$$

Se B.8 é uma solução de 4.3a, então deve ser verdade que

$$\lambda^2 e^{\lambda x} - (Pe)\lambda e^{\lambda x} = 0, \quad (\text{B.9})$$

ou ainda,

$$e^{\lambda x} (\lambda^2 - (Pe)\lambda) = 0. \quad (\text{B.10})$$

Como $e^{\lambda x} \neq 0$, $\forall \lambda, x \in R$, tem-se que $\lambda_1 = 0$ ou $\lambda_2 = Pe$. Pelo Teorema da superposição de soluções (GREENBERG, 1998), tem-se que

$$T(x) = C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x}, \quad (\text{B.11})$$

ou seja,

$$T(x) = C_1 + C_2 e^{Pe x}. \quad (\text{B.12})$$

Com as condições de contorno aplicadas à solução analítica geral B.12, o problema dado pela equação 4.5 apresenta a seguinte solução

$$T(x) = \frac{e^{x(Pe)} - 1}{e^{(Pe)} - 1}. \quad (\text{B.13})$$

A seguir tem-se a tabela B.2 com o valor da solução analítica dada pela equação B.13 em alguns pontos do domínio de cálculo. A figura B.2 ilustra esta solução.

Tabela B.2: Valor da solução analítica para a equação de advecção-difusão

Domínio: x	Solução analítica: equação B.13
0,0	0,00000000000000
0,1	0,00000001316883
0,2	0,00000011047402
0,3	0,00000082946757
0,4	0,00000614215121
0,5	0,00004539786870
0,6	0,00033546056744
0,7	0,00247875012062
0,8	0,01831563686533
0,9	0,13533528145441
1,0	1,00000000000000

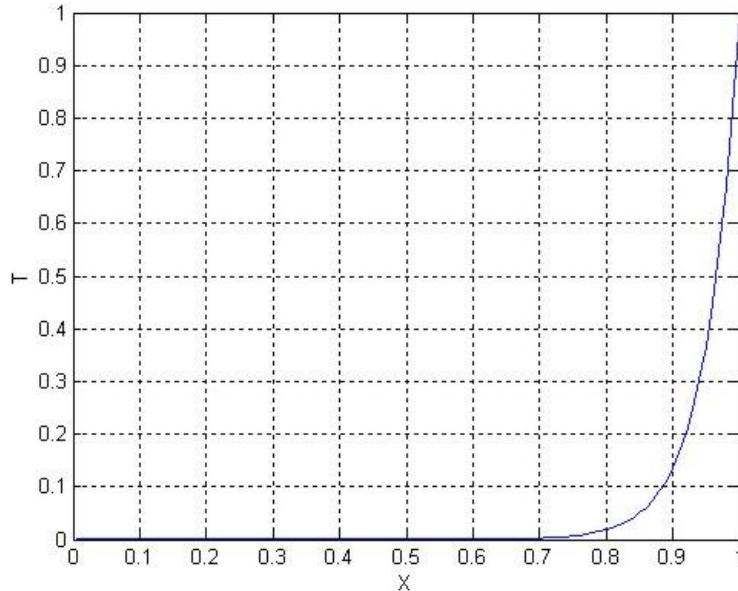


Figura B.2: Solução analítica para a equação de advecção-difusão

B.3 Solução da Equação de Burgers 1D

Para fins de comparação com a solução do problema de advecção-difusão dado pela equação 4.5, busca-se para o problema dado pela equação 5.3 a seguinte solução

$$u(x) = \frac{e^{x(Re)} - 1}{e^{(Re)} - 1}, \quad (\text{B.14})$$

que é a mesma solução analítica dada pela equação B.13 quando tem-se $Re = Pe$.

Neste caso tem-se a solução analítica do problema 5.3, dada pela equação B.14, idêntica a solução do problema 4.5, dada pela equação B.13.

Se B.14 é solução da equação 5.3, então deve ser verdade que

$$S(x) = \frac{(Re)^2 e^{x(Re)} (2e^{x(Re)} - e^{(Re)} - 1)}{(e^{(Re)} - 1)^2}. \quad (\text{B.15})$$

O número de Reynolds, $Re = 20$, será um dado fixo.

Na tabela B.2 do apêndice anterior, tem-se também os valores da solução analítica para a equação de Burgers, que também serve para a equação B.14, em alguns pontos do

domínio de cálculo. Desta forma, a figura B.2 ilustra esta solução, onde o eixo das ordenadas recebe a variável u , ao invés de T .

B.4 Solução da Equação de Laplace 2D

Primeiramente busca-se uma solução para a equação 6.1 da forma

$$T(x, y) = X(x)Y(y). \quad (\text{B.16})$$

Substituindo B.16 na equação 6.1 com $f(x) = 0$ e manipulando-se algebraicamente o resultado, tem-se

$$\frac{d^2X/dx^2}{X} = -\frac{d^2Y/dy^2}{Y} = \sigma^2, \quad (\text{B.17})$$

onde σ é uma constante.

Então, resolve-se o seguinte sistema de equações diferenciais ordinárias, onde cada uma das equações é uma equação homogênea de segunda ordem com coeficiente constante (GREENBERG, 1998)

$$\begin{cases} \frac{d^2X}{dx^2} - \sigma^2 X = 0 \\ \frac{d^2Y}{dy^2} - \sigma^2 Y = 0 \end{cases}. \quad (\text{B.18})$$

Como a equação de Laplace é uma equação diferencial linear, pode-se utilizar o teorema da superposição das soluções (GREENBERG, 1998), resultando na seguinte solução geral

$$T(x, y) = (A_1 + A_2x)(A_3 + A_4y) + [A_5 \cosh(\sigma x) + A_6 \operatorname{senh}(\sigma x)][A_7 \cos(\sigma y) + A_8 \operatorname{sen}(\sigma y)], \quad (\text{B.19})$$

onde $A_1, A_2, A_3, A_4, A_5, A_6, A_7$ e A_8 são constantes a determinar.

Com as condições de contorno dadas pela equação 6.2, estas constantes acima assumem seus valores e este problema apresenta a seguinte solução analítica, que é apresentada na figura B.3

$$T(x, y) = \sin(\pi x) \frac{\operatorname{senh}(\pi y)}{\operatorname{senh}(\pi)}. \quad (\text{B.20})$$

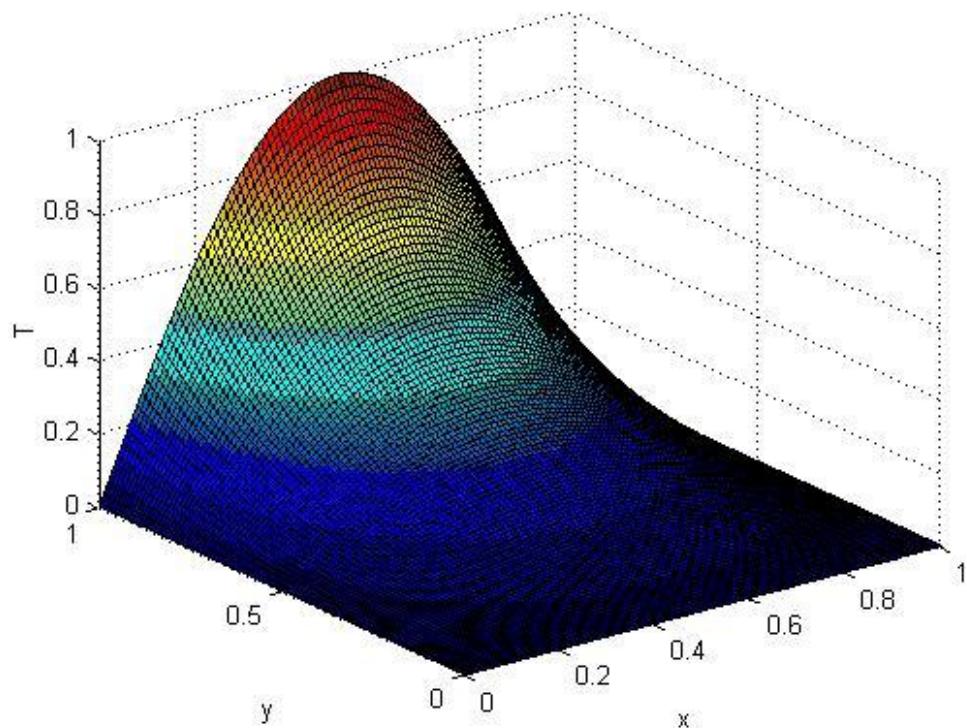


Figura B.3: Solução analítica para a equação de Laplace bidimensional

Apêndice C - Testes adicionais para critério de convergência

Nas tabelas 6.2, 6.7, C.1 e C.2 a solução analítica é comparada com a solução numérica via erro numérico (soma dos efeitos dos erros de iteração, arredondamento e discretização) com o uso da norma infinito ($\|\cdot\|_\infty$) para diversos N , diversas razões de engrossamento (r) e diversos valores de tolerâncias (ε), até mesmo ε_π = erro de máquina (onde não há erro de iteração, apenas o erro de arredondamento e erro de discretização inerente à malha) para o método *multigrid*. A norma vetorial $\|\cdot\|_\infty$, definida na seção 2.2, é repetida aqui por questão didática

$$\|e\|_\infty = \max_{1 \leq i \leq N} |e_i|,$$

onde e_i é a i -ésima coordenada do vetor e .

Tabela C.1: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema CS

r	N	$\ T_{analitica} - T_{numérica}\ _\infty$		
		$\varepsilon = 10^{-7}$	$\varepsilon = 10^{-10}$	ε_π
2	$513 \times 513 = 263.169$	$8,05 \times 10^{-7}$	$1,09 \times 10^{-6}$	$1,09 \times 10^{-6}$
3	$487 \times 487 = 237.169$	$1,21 \times 10^{-6}$	$1,21 \times 10^{-6}$	$1,21 \times 10^{-6}$
4	$513 \times 513 = 263.169$	$1,08 \times 10^{-6}$	$1,09 \times 10^{-6}$	$1,09 \times 10^{-6}$
5	$251 \times 251 = 630.001$	$4,62 \times 10^{-6}$	$4,56 \times 10^{-6}$	$4,56 \times 10^{-6}$

Tabela C.2: Norma infinito do erro para as malhas mais finas para cada uma das razões testadas com o solver MSI para o esquema FAS

r	N	$\ T_{analitica} - T_{numérica}\ _\infty$		
		$\varepsilon = 10^{-7}$	$\varepsilon = 10^{-10}$	ε_π
2	$513 \times 513 = 263.169$	$1,08 \times 10^{-6}$	$1,09 \times 10^{-6}$	$1,09 \times 10^{-6}$
3	$487 \times 487 = 237.169$	$1,19 \times 10^{-6}$	$1,21 \times 10^{-6}$	$1,21 \times 10^{-6}$
4	$513 \times 513 = 263.169$	$1,05 \times 10^{-6}$	$1,09 \times 10^{-6}$	$1,09 \times 10^{-6}$
5	$251 \times 251 = 630.001$	$4,58 \times 10^{-6}$	$4,56 \times 10^{-6}$	$4,56 \times 10^{-6}$

Em todos os casos (*multigrid* para as várias razões de engrossamento), o MSI foi o *solver* utilizado.

Nota-se que pelas tabelas 6.2, 6.7, C.1 e C.2 que, quando se tem $\varepsilon = 10^{-7}$, a norma infinito do erro numérico pode ser significativamente diferente para alguns valores de N ao se usar CS e FAS. Nota-se também que, quando se tem $\varepsilon = 10^{-10}$ ou se atinge ε_π , o erro numérico é idêntico nos dois esquemas CS e FAS.

Logo os resultados descritos no capítulo 6 e na maior parte desta Tese, valem apenas para a tolerância padrão $\varepsilon = 10^{-7}$. Os aspectos qualitativos e quantitativos sofrem pouca influência quando se tem $\varepsilon = 10^{-10}$ ou mesmo ε_π .

Pode-se concluir disto que, o critério de parada dado pela equação 4.22 e adotado nesta Tese (critério corrente na literatura, Ferziger e Peric, 1999), não é um bom critério, pois ele não controla de forma adequada o erro de iteração. Portanto, deve-se ter cautela nas extrapolações desses resultados.

Apêndice D - Artigos completos com resultados desta Tese publicados em anais de Eventos

Neste apêndice são apresentados alguns dos artigos completos publicados nos Anais de Eventos, que têm como resultado, os resultados desta Tese.

D.1 Efeito de Parâmetros do Método *Multigrid* sobre o Tempo de CPU para a Equação de Burgers Unidimensional

D.2 Effect of Parameters of *Multigrid* Method on the CPU Time for One-dimensional Problems

D.3 Efeito de Parâmetros do Método *Multigrid* CS e FAS sobre o Tempo de CPU em Problemas 1D Lineares e Não-lineares

D.4 Efeito da Razão de Aspecto de Malha sobre o Tempo de CPU para a Equação de Laplace 2D Resolvida com o Método *Multigrid*

D.5 Efeito de Parâmetros do Método *Multigrid* CS e FAS sobre o Tempo de CPU para a Equação de Laplace Bidimensional

D.1 Efeito de Parâmetros do Método *Multigrid* sobre o Tempo de CPU para a Equação de Burgers Unidimensional

Pinto, M. A. V., Santiago, C. D., Marchi, C. H.

Proceedings of the XXVI Iberian Latin-American Congress on Computational Methods in Engineering – CILAMCE- 2005

Brazilian Association for Computational Mechanics (ABMEC) & Latin American Association of Computational Methods in Engineering (AMC), Guarapari, Espírito Santo, Brazil, 19th – 21th October 2005.

D.2 Effect of Parameters of *Multigrid* Method on the CPU Time for One-dimensional Problems

Pinto, M. A. V., Santiago, C. D., Marchi, C. H.

Proceedings of the 18th International Congress of Mechanical Engineering – COBEM 2005
November 6-11, 2005, Ouro Preto, MG

D.3 Efeito de Parâmetros do Método *Multigrid CS e FAS* sobre o Tempo de CPU em Problemas 1D Lineares e Não-lineares

Oliveira, F., Pinto, M. A. V., Santiago, C. D., Marchi, C. H.

Proceedings of the XXVII Iberian Latin-American Congress on Computational Methods in Engineering – CILAMCE- 2006

September 3 to 6, 2006 – Belém, Pará - Brazil.

D.4 Efeito da Razão de Aspecto de Malha sobre o Tempo de CPU para a Equação de Laplace 2D Resolvida com o Método *Multigrid*

Pinto, M. A. V., Marchi, C. H.

Proceedings of the XXVII Iberian Latin-American Congress on Computational Methods in Engineering – CILAMCE- 2006

September 3 to 6, 2006 – Belém, Pará - Brazil.

D.5 Efeito de Parâmetros do Método *Multigrid CS e FAS* sobre o Tempo de CPU para a Equação de Laplace Bidimensional

Pinto, M. A. V., Marchi, C. H.

Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering – ENCIT
2006

Braz. Soc. of Mechanical Sciences and Engineering (ABCM), Curitiba, Brazil, Dec. 5–8,
2006.